

Ein Framework als Grundlage der Ausgestaltung von Quality-Gate-Referenzprozessen für die Software-Entwicklung

Von der

Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des Grades eines

Doktors der Naturwissenschaften
Dr. rer. nat.

genehmigte Dissertation

von

Dipl.-Math. Thomas Flohr
geboren am 2. Januar 1977 in Burgwedel

2008

Referent: Prof. Dr. Kurt Schneider,
Gottfried Wilhelm Leibniz Universität Hannover
Korreferent: Prof. Dr. techn. Wolfgang NejdI,
Gottfried Wilhelm Leibniz Universität Hannover
Tag der Promotion: 3. November 2008

Danke

Ich bedanke mich an dieser Stelle ganz besonders bei meinem Doktorvater Prof. Dr. Kurt Schneider für die hervorragende Betreuung und die zahlreichen kritischen, aber stets konstruktiven Rückmeldungen bezüglich meiner wissenschaftlichen Arbeit.

Auch bei meinen Kollegen Daniel, Eric, Kai, Leif, Sebastian und Ulla möchte ich mich für die sehr gute Zusammenarbeit bedanken.

Weiterhin möchte ich mich bei allen bedanken, die diese Arbeit sprachlich korrektur gelesen haben. Insbesondere bedanke ich mich diesbezüglich bei Patrick.

Ganz besonders möchte ich bei meiner Ehefrau Yurong und meinem kleinen Sohn Tim für die emotionale Unterstützung und die viele aufgebrachte Geduld danken.

Kurzzusammenfassung

Noch immer wird eine große Anzahl von Softwareprojekten nur durch eine erhebliche Überschreitung von Zeit und Kosten oder gar nicht erfolgreich abgeschlossen, da die qualitative Situation für das Management nicht einsehbar ist. Maßnahmen durch das Managements erfolgen daher häufig zu.

Eine Lösung des Problems besteht darin, die qualitative Kontrolle von Projektergebnissen und die Möglichkeit für steuernde Eingriffe zu bestimmten signifikanten Zeitpunkten (so genannten Quality Gates) verbindlich für alle Projekte festzuschreiben. Beim Erreichen eines Quality Gates entscheidet das Management über die Fortsetzung des Projektes und notwendige Maßnahmen. Dazu werden geforderte Ergebnisse formal gegen vorher definierte Kriterien auf ihre Erfüllung hin geprüft. Quality Gates werden vielfach in anderen Domänen genutzt, zum Beispiel in der Automobil-Entwicklung und der Serienproduktion technischer Güter. Auch im Bereich der Software-Entwicklung werden sie zunehmend genutzt. Im Moment fehlt hierfür jedoch der theoretische Unterbau.

Damit ein Unternehmen Quality Gates erfolgreich einsetzen kann, werden verschiedene Strukturen, Aktivitäten, Rollen, Dokumente und Hilfsmittel benötigt, die in einem Quality-Gate-Referenzprozess zusammengefasst werden. Die genaue Ausgestaltung eines Quality-Gate-Referenzprozesses orientiert sich dabei an den Bedürfnissen des Unternehmens. Da die Ausgestaltung keine triviale Aufgabe ist, benötigt das Prozessmanagement eines Unternehmens ein Framework, das die Ausgestaltung in vertretbarer Zeit ermöglicht. Dieses Framework stellt das zentrale Ergebnis dieser Dissertation dar.

Leider wird das Quality-Gate-Konzept häufig mit anderen Konzepten gleichgesetzt. Ebenso existieren viele synonyme Begriffe. Daher ist eine einheitliche Terminologie und eine Charakterisierung und Abgrenzung des Quality Gates ein notwendiger Bestandteil des Frameworks. Um die notwendige Abgrenzung und Charakterisierung vornehmen zu können, wurden verschiedene Quality-Gate-Referenzprozesse aus der Literatur untersucht. Des Weiteren wurde eine empirische Erhebung im Rahmen dieser Arbeit durchgeführt. Sie liefert weitere Ergebnisse und Hinweise auf Problempunkte.

Aufbauend auf diesen Ergebnissen wurden die notwendigen Teilkonzepte von Quality-Gate-Referenzprozessen untersucht. Diese Teilkonzepte können auf insgesamt fünf Pakete aufgeteilt werden: Struktur-, Inhalts-, Review-, Steuerungs- und Anpassungskonzepte. Die Ausgestaltung der Teilkonzepte hängt maßgeblich davon ab, welche Strategie ein Prozessmanagement verfolgt: *Quality Gates als einheitliche Qualitäts-Leitlinie* oder *Quality Gates als flexible Qualitätsstrategie*.

Das Prozessmanagement eines Unternehmens kann in Abhängigkeit einer der folgenden Ausgangssituationen das Framework einsetzen:

- Das Unternehmen besitzt keinen Quality-Gate-Referenzprozess.
- Das Unternehmen besitzt bereits einen Quality-Gate-Referenzprozess mit Defiziten.
- Das Unternehmen will seinen Quality-Gate-Referenzprozess optimieren.

Resultierend können verschiedene Wege beschritten werden: entweder die Ausgestaltung von Grund auf *oder* die Analyse der Defizite durch Bewertungsprofile *oder* die Optimierung hinsichtlich Effizienz und Effektivität durch Erfahrungskreisläufe und den Einsatz von Softwarewerkzeugen.

Nutzen und Mächtigkeit des Frameworks wurden anhand von drei Fallstudien validiert. Zwei der Fallstudien zeigen die Anwendung des Frameworks in studentischen Software-Projekten. Die erste Fallstudie untersucht die Verbesserung eines Quality-Gate-Referenzprozesses durch Bewertungsprofile und durch den Einsatz von Softwarewerkzeugen und Erfahrungskreisläufen. Dabei konnte der zeitliche Aufwand für das Gate-Review auf rund 28% gesenkt werden. Zusätzlich wurde die Anpassung des Quality-Gate-Referenzprozesses auf verschiedene Projektsituationen verbessert und die Kriterien durch Erfahrungskreisläufe optimiert. Die zweite Fallstudie beschäftigt sich mit der Ausgestaltung eines Quality-Gate-Referenzprozesses von Grund auf. Durch den Einsatz konnten die Auswirkungen verschiedener Vorgehen zur Erstellung von Kriterien untersucht werden. Die letzte Fallstudie untersucht die Mächtigkeit des Anpassungskonzeptes des Frameworks. Dabei kann das im Rahmen dieser Arbeit entwickelte und auf Fuzzy-Logik basierende Anpassungskonzept problemlos zur Anpassung des Quality-Gate-Referenzprozesses des V-Modells XT genutzt werden.

Abstract

A considerable number of software projects still exceed time and budget or completely fail, because the qualitative situation of these projects is not visible to the management. Consequently, the management is not able to take actions in time.

The problem can be resolved by monitoring the quality of project results and by steering a project at certain major points (so-called Quality Gates). Then each project has to pass these Quality Gates. At each Quality Gate the management makes a decision whether a project may proceed or not. If necessary, the management can take actions at those Quality Gates. In order to determine the situation of a project, the project's results have to be checked against predefined criteria. Quality Gates are often used in certain domains, e.g. in car development or in serial production of industrial goods. In the domain of software development Quality Gates are used cumulatively. Unfortunately, a theoretical foundation for Quality Gates is currently missing in the domain of software development.

Special structures, activities, roles, documents and support tools are required in order that a company can successfully establish Quality Gates. These subconcepts can be combined to a quality gate reference process. A company has to design a quality gate reference process in order to apply it to the company's projects. The exact design of the quality gate reference process depends on a company's needs. Since designing an appropriate quality gate reference process is not an easy task, the process management of a company needs a framework supporting the process of designing. This framework is the central result of this thesis.

Unfortunately, the quality gate concept is often equated with other concepts. At the same time a remarkable number of synonym concepts exist. Thus, the framework has to provide a consistent terminology and has to define the scope of the quality gate concept. To achieve this aim, a study of the available literature was conducted. Furthermore, an empirical survey was conducted. The empirical survey provides more results as well as an overview of current problems.

The results were used to identify the mandatory subconcepts of quality gate reference processes. These subconcepts can be subdivided into five packages: structural, criteria, review, steering and tailoring concepts. When designing a quality gate reference process, a process management has to choose one of the following two strategies: *Quality Gates as a consistent quality guideline* or *Quality Gates as a flexible quality strategy*. The selection of the strategy strongly influences the design of each subconcept.

A company's process management can use the framework to design an appropriate quality gate reference process. The process of designing has to take in account the situation of the company:

- The company does not possess a quality gate reference process.
- The company already possesses a quality gate reference process with shortcomings.
- The company wants to optimize its quality gate reference process.

Each of these situations requires different actions to be taken: either a quality gate reference process has to be designed from scratch *or* the available quality gate reference process has to be judged through assessment profiles *or* the quality gate reference process can be optimized by applying experience cycles and using software tools.

Three case studies show the power of the framework. Two case studies show the application of the framework in students' software projects. The first case study analyzes the improvement of a quality gate reference process through assessment profiles and by introducing software tools. As a result the time effort of the gate review could be reduced to roughly 28%. Moreover, some criteria and the ability to tailor the quality gate reference process to different project situations were improved through the application of an experience cycle. The second case study deals with a quality gate reference process being designed from scratch. This case study analyzes the effects of different approaches to obtain suitable criteria. The last case study shows the power of the framework's tailoring concepts. The framework uses a tailoring concept which rests upon the fuzzy concept. This tailoring concept can easily be used to tailor the quality gates reference process within the V-Model XT of the German federal administration.

Schlagwörter: Quality Gates, Qualitätsmanagement, Softwareprozess-Verbesserung

Keywords: Quality Gates, Quality Management, Software Process Improvement

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation.....	1
1.2	Zielsetzung dieser Arbeit	4
1.3	Gliederung der Arbeit	6
2	Grundlagen der Software-Entwicklung	9
2.1	Software	9
2.2	Software-Entwicklung	9
2.2.1	Grundlagen der Modellbildung	9
2.2.2	Projekt.....	10
2.2.3	Prozess und Prozessmodell.....	12
2.2.4	Software-Entwicklungsprozess.....	12
2.2.5	Referenzprozess und Anpassung	13
2.3	Die besonderen Schwierigkeiten der Software-Entwicklung	15
2.4	Die Modellierungssprache SPEM.....	16
3	Grundlagen des Managements der Software-Entwicklung	19
3.1	Projektmanagement.....	19
3.1.1	Projektorganisation und Rollen	20
3.1.2	Verhältnis von Prozess und Projekt.....	22
3.1.3	Planungsaktivitäten.....	22
3.1.4	Die Entwicklung als Teil des Software-Entstehungsprozesses	24
3.1.5	Projektfortschrittskontrolle und Steuerung.....	25
3.1.6	Projektbesprechungen.....	26
3.1.7	Multiprojektmanagement.....	27
3.2	Risikomanagement.....	28
3.2.1	Risikomanagement nach Phasen.....	28
3.2.2	Risikoermittlung	29
3.2.3	Gegenmaßnahmen und Verfolgung.....	31
3.3	Qualitätsmanagement.....	31
3.3.1	Qualitätssichernde Maßnahmen.....	32
3.3.2	Qualitätsorganisation als Schattenhierarchie	32
3.3.3	Zum Begriff des Messens	33
3.3.4	Technische Reviews	41
4	Untersuchung von Quality-Gate-Referenzprozessen	45
4.1	Methodische Herangehensweise.....	45
4.2	Das Quality-Gate-Konzept in der Literatur	46
4.2.1	Das Quality-Gate-Konzept als Forschungsgegenstand im Software Engineering	46
4.2.2	Das Quality-Gate-Konzept als Forschungsgegenstand in anderen Domänen	47
4.2.3	Zusammenfassung der Literatur-Untersuchung	49
4.3	Empirische Erhebung.....	50
4.3.1	Zielsetzungen der empirischen Erhebung.....	51
4.3.2	Durchführung und Inhalte der empirischen Erhebung	51
4.3.3	Auswertung.....	53
4.3.4	Zusammenfassung und Bewertung der Ergebnisse	59
4.3.5	Ausgangspunkte für die weiteren Untersuchungen	60
4.4	Quality-Gate-Referenzprozess bei interdisziplinären Projekten	61
4.4.1	Hintergrund und Überblick.....	61
4.4.2	Ausgestaltung des Quality-Gate-Referenzprozesses nach Pfeifer.....	61
4.5	Quality-Gate-Referenzprozess des V-Modells XT	63
4.5.1	Hintergrund und Überblick.....	64
4.5.2	Ausgestaltung des Quality-Gate-Referenzprozesses im V-Modell XT	64
4.5.3	Anpassung des Quality-Gate-Referenzprozesses im V-Modell XT	66

4.6	Quality-Gate-Referenzprozess des Stage-Gate-Prozesses	69
4.6.1	Hintergrund und Überblick.....	70
4.6.2	Verwendung von Gates im Stage-Gate-Prozess	73
4.6.3	Ausgestaltung effektiver und effizienter Gates nach Cooper	74
4.7	Bezug zwischen Prozessreifegradmodellen und Quality-Gate-Referenzprozessen	75
4.7.1	Bezug zwischen CMMI und Quality-Gate-Referenzprozessen	76
4.7.2	Bezug zwischen SPICE und Quality-Gate-Referenzprozessen	78
4.7.3	Fazit der Untersuchung von Prozessreifegradmodellen	79
4.8	Abschließende Betrachtung	79
4.8.1	Charakteristika von Quality Gates.....	80
4.8.2	Einordnung als Meilenstein-Konzept	80
4.8.3	Einordnung als Review-Konzept.....	81
4.8.4	Weitere identifizierte Problempunkte.....	82
5	Abstrahiertes Domänenmodell für Quality-Gate-Referenzprozesse	83
5.1	Elemente des abstrahierten Domänenmodells	84
5.1.1	Strukturelle Konzepte	84
5.1.2	Inhaltliche Konzepte.....	85
5.1.3	Reviewkonzepte.....	86
5.1.4	Steuerungskonzepte	87
5.1.5	Anpassungskonzepte	87
5.2	Paketübergreifende Beziehungen.....	88
5.3	Überblick über das weitere Vorgehen.....	89
6	Ausgestaltung der strukturellen Konzepte.....	91
6.1	Formalisierung des Gate-Netzwerk-Konzeptes	91
6.2	Allgemeine Richtlinien zur Ausgestaltung von Gate-Netzwerken	95
6.3	Gate-Netzwerke bei wasserfallartigem Vorgehen	96
6.4	Gate-Netzwerke bei iterativem Vorgehen.....	97
6.5	Gate-Netzwerke bei agilem Vorgehen.....	98
6.6	Gate-Netzwerke bei inkrementellem Vorgehen.....	99
6.7	Gate-Netzwerke bei parallel verlaufenden Handlungssträngen	100
6.8	Gate-Netzwerke bei Projekten mit mehreren Teilprojekten	101
6.9	Operationen auf Basis-Gate-Netzwerken.....	101
7	Ausgestaltung der inhaltlichen Konzepte	105
7.1	Kriterienerstellung	105
7.1.1	Individualität der Kriterien	107
7.1.2	Zeitpunkt der Kriterienerstellung	108
7.1.3	Systematik des Vorgehens bei der Kriterienerstellung.....	109
7.2	Ersteller	110
7.3	Kriterien	112
7.4	Ergebnisse	117
7.5	Beispiel für eine Ausgestaltung der inhaltlichen Konzepte	117
8	Ausgestaltung der Reviewkonzepte.....	123
8.1	Gate-Review	123
8.1.1	Risikomanagement im Gate-Review	127
8.1.2	Überlegungen zum Aufwand eines Gate-Reviews	128
8.2	Gate-Verantwortlicher	130
8.3	Gutachter.....	130
8.4	Projektvertreter	130
8.5	Protokollant.....	131
8.6	Protokoll.....	131
9	Ausgestaltung der Steuerungskonzepte	133

9.1	Mögliche Entscheidungen.....	133
9.2	Gatekeeper	134
9.3	Entscheidungsunterstützung	135
10	Konzepte zur Anpassung von Quality-Gate-Referenzprozessen.....	141
10.1	Gate-Management.....	141
10.2	Prozess-Tailorer	142
10.3	Tailoring-Verfahren	142
10.3.1	Grundlagen der Fuzzy-Logik	144
10.3.2	Das Fuzzy-Inferenz-System	146
10.3.3	Beispiel für die Unterstützung des Tailoring mit einem Fuzzy-Inferenz-System	151
10.3.4	Unterstützung des Tailorings durch Softwarewerkzeuge.....	153
10.3.5	Einschränkungen und Herausforderungen eines Fuzzy-Inferenz-Systems	155
10.3.6	Umgang mit den Herausforderungen und Einschränkungen.....	156
10.4	Auswählbare Elemente	158
10.5	Projektmodell.....	161
10.6	Beziehungen zwischen Projektmerkmalen und anpassbaren Quality-Gate-Teilkonzepten.....	162
10.7	Wechselwirkung zur Anpassung des Entwicklungs-Referenzprozesses	165
11	Verbesserung von Quality-Gate-Referenzprozessen.....	167
11.1	Verbesserung durch Erfahrungskreisläufe.....	167
11.2	Überlegungen zur Verbesserung der Effektivität und Effizienz von Quality Gates	168
11.3	Analyse der Informationsflüsse	169
11.4	Probleme durch nichtverfestigte Erfahrungen und fehlendes Gate-Management	170
11.5	Softwarewerkzeuge zur Unterstützung von Quality-Gate-Prozessen	172
11.5.1	Unterstützung der Einplanung von Quality Gates.....	173
11.5.2	Darstellung des Projektfortschritts	173
11.5.3	Unterstützung der Kriterienerstellung und Operationalisierung	175
11.5.4	Unterstützung der Durchführung des Gate-Reviews.....	177
11.5.5	Unterstützung des Erfahrungskreislaufs.....	181
11.5.6	Zielerreichung durch den Einsatz von NetQGate.....	182
12	Möglichkeiten zur Verwendung des Frameworks.....	185
12.1	Unternehmen ohne Quality-Gate-Referenzprozess.....	185
12.2	Unternehmen mit unzureichend ausgestalteten Quality-Gate-Referenzprozess	187
12.3	Unternehmen mit Quality-Gate-Referenzprozess mit Verbesserungspotential	191
13	Fallstudien.....	193
13.1	Quality-Gate-Referenzprozess in studentischen Softwareprojekten.....	193
13.1.1	Ausgestaltung der strukturellen Konzepte	195
13.1.2	Ausgestaltung der inhaltlichen Konzepte	199
13.1.3	Ausgestaltung der Reviewkonzepte	200
13.1.4	Ausgestaltung der Steuerungskonzepte.....	202
13.1.5	Ausgestaltung der Anpassungskonzepte	203
13.1.6	Erfahrungen mit NetQGate als Teil des Erfahrungskreislaufes	205
13.1.7	Untersuchung der Auswirkungen des Quality-Gate-Referenzprozesses.....	206
13.1.8	Abschließende Betrachtung.....	207
13.2	Einsatz von Quality Gates in der Projektarbeit PWS.....	208
13.2.1	Ausgestaltung der strukturellen Konzepte	209
13.2.2	Ausgestaltung der inhaltlichen Konzepte.....	209
13.2.3	Ausgestaltung der Reviewkonzepte	212
13.2.4	Ausgestaltung der Steuerungskonzepte.....	213
13.2.5	Untersuchung der Auswirkungen des Quality-Gate-Referenzprozesses.....	214
13.2.6	Abschließende Betrachtung.....	214
13.3	Abbildung des V-Modell-XT-Tailoring-Verfahrens in Fuzzy-Logik.....	215
14	Zusammenfassung und Ausblick.....	217

14.1	Zusammenfassung.....	217
14.2	Behandlung der Problempunkte durch das Framework	220
14.3	Ausblick	222
	Literaturverzeichnis	225
	Anhang A – Fragebogen und Auswertung	235
	Anhang B – Mathematische Definition der Zugehörigkeits-funktionen	245
	Anhang C – Umsetzung des Tailoring-Verfahrens für die studentischen Softwareprojekte	247
	Anhang D – Erstellte Kriterien für die Projektarbeit PWS	251
	Anhang E – Zugehörigkeitsfunktion für das Tailoring-Verfahren des V-Modells XT.....	253
	Werdegang	255

Mathematische Konventionen

N	Menge der natürlichen Zahlen
N^0	Menge der natürlichen Zahlen mit 0
R	Menge der reellen Zahlen
B	Menge der Booleschen Werte: $B = \{true, false\}$
$[a; b]$	Abgeschlossenes Intervall: $[a; b] = \{x \in R \mid a \leq x \leq b\}$
$(a; b)$	Offenes Intervall: $(a; b) = \{x \in R \mid a < x < b\}$
$ x $ mit $x \in R$	Betrag einer reellen Zahl x : $ x = \begin{cases} x, & \text{falls } x \geq 0 \\ -x, & \text{falls } x < 0 \end{cases}$
$\prod_{i=1}^n A_i = A_1 \times \dots \times A_n$ oder $A^n = \prod_{i=1}^n A = A \times \dots \times A$	Kartesisches Produkt: $\prod_{i=1}^n A_i = \{(a_1, \dots, a_n) \mid a_i \in A_i, i = 1 \dots n\}$ oder $A^n = \{(a_1, \dots, a_n) \mid a_i \in A, i = 1 \dots n\}$
$P(A)$	Potenzmenge $P(A) = \{M \mid M \subseteq A\}$
$ A $ mit A ist eine Menge	Kardinalität der Menge A bzw. Anzahl der Elemente in A
$\min(x, y)$ mit $x, y \in R$	$\min(x, y) = \begin{cases} x, & \text{falls } x \leq y \\ y, & \text{sonst} \end{cases}$
$\max(x, y)$ mit $x, y \in R$	$\max(x, y) = \begin{cases} x, & \text{falls } x \geq y \\ y, & \text{sonst} \end{cases}$

1 Einleitung

1.1 Motivation

Die Standish Group stellt im viel zitierten Chaos-Report 1994 [CHAOSReport '94] fest, dass lediglich rund 16% der Projekte, die sich mit Informationstechnik (IT) beschäftigen, uneingeschränkt erfolgreich abgeschlossen werden. Rund 31% der IT-Projekte scheitern hingegen. Die verbleibenden rund 53% werden zwar abgeschlossen, jedoch werden dabei Zeit und Kosten teilweise erheblich überschritten. Die Kosten für gescheiterte IT-Projekte beliefen sich allein 1994 auf ungefähr 81 Milliarden US-Dollar.

2006 sieht die Lage zwar entspannter aus [CHAOSReport '06], allerdings scheitern immer noch rund 19% der IT-Projekte, während hingegen immerhin rund 35% zum Erfolg geführt werden. Weiterhin leidet mit 46% der größte Teil der IT-Projekte (obwohl abgeschlossen) unter Kosten- oder Terminüberschreitungen.

Seit 1999 benennt die Standish Group ununterbrochen die drei folgenden kritischen Erfolgsfaktoren für IT-Projekte:

1. Die Beteiligung von Endanwendern,
2. die Unterstützung durch das höhere Management,
3. und die Formulierung von klaren Geschäftszielen.

Die Forderung nach umfassender Unterstützung durch das höhere Management sieht unter anderem vor, die Qualität und den Projektfortschritt von IT-Projekten rechtzeitig systematisch anhand von Kriterien zu kontrollieren, um steuernd eingreifen zu können. Durch die Vorgabe von Kriterien, wirkt das Management in zweierlei Hinsicht steuernd auf den Software-Entwicklungsprozess ein:

1. Die Kriterien sind Vorgaben des Managements, die durch die Entwickler einzuhalten sind. Die Entwickler werden also alles Notwendige tun, um diese Kriterien zu erfüllen.
2. Das Management kann bei Abweichung vom definierten Soll steuernd eingreifen. Das Soll wird dabei durch die Kriterien festgelegt.

Im Rahmen der Steuerung können Maßnahmen ergriffen werden, die Fehlentwicklungen innerhalb eines Projektes entgegenwirken können. Maßnahmen, die ohne Einwirkung des Managements nicht ergriffen worden wären. Der Idealfall für die Projektteilnehmer stellt die uneingeschränkte Fortsetzung des Projektes dar. Im Extremfall ist aber auch der Abbruch des Projektes möglich.

Rechtzeitigkeit bedeutet dabei, Maßnahmen ergreifen zu können, bevor ein größerer (finanzieller) Schaden entstehen kann. Dies setzt aber gleichzeitig voraus, dass geeignete Ergebnisse vorhanden sind, damit eine Kontrolle sinnvoll und möglich ist. Das Ende einer Aktivität innerhalb eines Projektes ist hierfür ein geeigneter Zeitpunkt, da hier Ergebnisse vorliegen, die kontrolliert werden können. Aufgrund der Vielzahl von Aktivitäten innerhalb eines Projektes, würde eine zu häufige Kontrolle durch das höhere Management sich jedoch bremsend auf das Projekt auswirken. Außerdem würde es bei vielen parallel verlaufenden Projekten (die wiederum alle kontrolliert werden müssten) zu einer stärkeren Belastung des Managements kommen.

Ein Mittelweg besteht darin, die Kontrolle und Steuerung an bestimmten signifikanten Meilensteinen bindend für jedes Projekt zu etablieren. Hierfür bieten sich Quality Gates als Möglichkeit zur Prüfung des Erfüllungsgrades der einzelnen Qualitätsziele an. Ein Quality Gate ist dabei ein spezieller Meilenstein in einem Projekt, an dem auf Grundlage einer formalen Prüfung von qualitätsbezogenen Kriterien eine Entscheidung über die Fortsetzung oder den Abbruch eines Projektes getroffen wird. Quality Gates sind bildlich gesprochen Schranken zwischen den verschiedenen Prozessschritten eines Projek-

tes: ein Projekt kann bei Erreichen des Quality Gates nur dann weitergeführt werden, wenn alle Kriterien oder zumindest eine genügend große Anzahl der Kriterien erfüllt sind. Dabei wird sichergestellt, dass alle Ergebnisse des Projektes zum Zeitpunkt des Quality Gates gut genug sind, um mit ihnen weiterarbeiten zu können. Durch die Kriterien eines Quality Gates können die Ergebnisse einerseits und die qualitativen Anforderungen an die Ergebnisse andererseits festgelegt werden. Hierdurch können sie dazu benutzt werden, um die Schnittstellen zwischen einzelnen Projektphasen festzulegen. Um Quality Gates etablieren zu können, sind bestimmte Strukturen, Aktivitäten, Rollen, Dokumente und Hilfsmittel notwendig, die in einem Quality-Gate-Referenzprozess zusammengefasst werden. Die genaue Ausgestaltung des Quality-Gate-Referenzprozesses orientiert sich dabei an den Bedürfnissen des Unternehmens, das Quality Gates etablieren möchte.

Quality Gates haben ihren Ursprung in der Automobil-Entwicklung und in der Serienproduktion technischer Güter, halten jedoch in den letzten Jahren zunehmend Einzug in Systementwicklungsprojekte und in letzter Zeit auch in reine Software-Entwicklungsprojekte. Im Bereich der Software-Entwicklung sind Quality Gates dennoch weitgehend unerforscht [Wallin und Larsson et al. '02].

Quality Gates in der Serienproduktion setzen auf statistisch ermittelte Werte, die für die Kontrollaktivitäten in zukünftigen Projekten als Soll genutzt werden können. In der Software-Entwicklung ist eine derartige Ausgangslage nicht gegeben, da Software-Entwicklungsprojekte in einem hohen Maß individuell sind. Folglich kann ein in der Serienproduktion praktizierter Quality-Gate-Referenzprozess nur bedingt auf die Software-Entwicklung übertragen werden. Vielmehr muss ein geeigneter Quality-Gate-Referenzprozess anders ausgestaltet werden, um den speziellen Problemen der Software-Entwicklung gerecht zu werden. Eine Orientierung an den Quality-Gate-Referenzprozessen aus anderen Domänen ist dennoch sinnvoll, da sie über Jahre hinweg entwickelt und optimiert wurden.

Bei der Verwendung von Quality Gates können zwei grundlegende Strategien unterschieden werden. Ein Unternehmen kann sich, je nach Zielsetzung, bei der Ausgestaltung eines Quality-Gate-Referenzprozesses für eine dieser beiden Strategien entscheiden:

- **Quality Gates als einheitliche Qualitäts-Leitlinie:** Hierbei muss jedes Projekt die gleichen Quality Gates durchschreiten und wird an den gleichen Kriterien gemessen. Die Anpassung eines Quality-Gate-Referenzprozesses, der dieser Strategie folgt, ist (wenn überhaupt) überaus eingeschränkt zulässig. Ziel ist es, in jedem Projekt mindestens das gleiche Qualitätsniveau zu erreichen – eine qualitative Leitlinie für jedes Projekt wird somit etabliert. Quality Gates können damit als einheitliches Fortschrittsmaß verwendet werden [Schneider '07b]. Der Fortschrittsvergleich zwischen Projekten wird auf diese Weise möglich, da geprüft werden kann, welche Projekte ein bestimmtes Quality Gate bereits passiert haben und welche nicht. Das Management kann leicht erkennen, wenn ein Projekt einem anderen Projekt gegenüber (qualitativ) im Verzug ist und entsprechend handeln. Quality Gates können somit leicht als Instrument des Multi-Projektmanagements eingesetzt werden.
- **Quality Gates als flexible Qualitätsstrategie:** Hierbei sind Quality Gates weniger ein Fortschrittsmaß und damit auch weniger ein Instrument des Multi-Projektmanagements. Vielmehr können im stärkeren Maße Anzahl, Anordnung und Auswahl der Quality Gates oder Kriterien an ein Projekt angepasst werden [Pfeifer et al. '03]. Quality Gates und Kriterien können damit genauer auf die qualitativen Anforderungen eines Projektes zugeschnitten werden – ein Vergleich zwischen den Projekten wird jedoch gleichzeitig erschwert. Folglich kann die Qualität der Ergebnisse eines Projektes sehr viel genauer gesteuert werden – die Qualität der Ergebnisse steigt also. In der Regel werden jedoch ähnliche Projekte vergleichbare Quality Gates haben und an ähnlichen Kriterien gemessen werden. In gewisser Weise kommen bei dieser Strategie Quality Gates dem Konzept des projektbezogenen Meilensteins recht nahe, grenzen sich jedoch in ihrer Häufigkeit und ihrer Möglichkeit zur Entscheidung über die Fortsetzung des Projektes von ihnen ab.

Eine Beschäftigung mit dem Themenkomplex Quality Gate durch Recherche im Internet und in der Literatur (Dissertationen, Standardwerke und Konferenzbände) liefert ein weites Begriffsfeld. Zum einen wird an vielen Stellen eine synonyme Bezeichnung verwendet, zum anderen wird das Konzept häufig mit verschiedenen anderen Konzepten gleichgesetzt. Ein Beispiel für das letztere Problem ist

die Gleichsetzung des Review-Konzeptes oder des Meilenstein-Konzeptes mit dem Konzept des Quality Gates. Diese Gleichsetzung ist nicht unberechtigt, sie bedarf jedoch einer näheren Betrachtung.

Abbildung 1 zeigt das Quality-Gate-Konzept im Verhältnis zu anderen Meilenstein-Konzepten. Konzepte im gleichen Kreis sind hinsichtlich der Relation, die im entsprechenden Kommentar steht, gleich. Demnach ist ein Quality Gate ein spezieller Entscheidungspunkt, dessen Prüfkriterien qualitätsorientiert sind. Entscheidungspunkte sind hinsichtlich ihrer Prüfkriterien unfokussiert, zeichnen sich jedoch dadurch aus, dass bei ihrem Erreichen eine Entscheidung über die Projektfortsetzung getroffen wird. Ein Entscheidungspunkt ist ein spezieller Hauptmeilenstein und kommt daher selten und nur an besonderen Stellen in einem Projekt vor. Deutlich wird ferner, dass projektbezogene Meilensteine und Hauptmeilensteine verschiedene Konzepte sind. Insbesondere ist ein Quality Gate kein projektbezogener Meilenstein, wengleich eine gewisse Anpassung an die Projektsituation möglich ist.

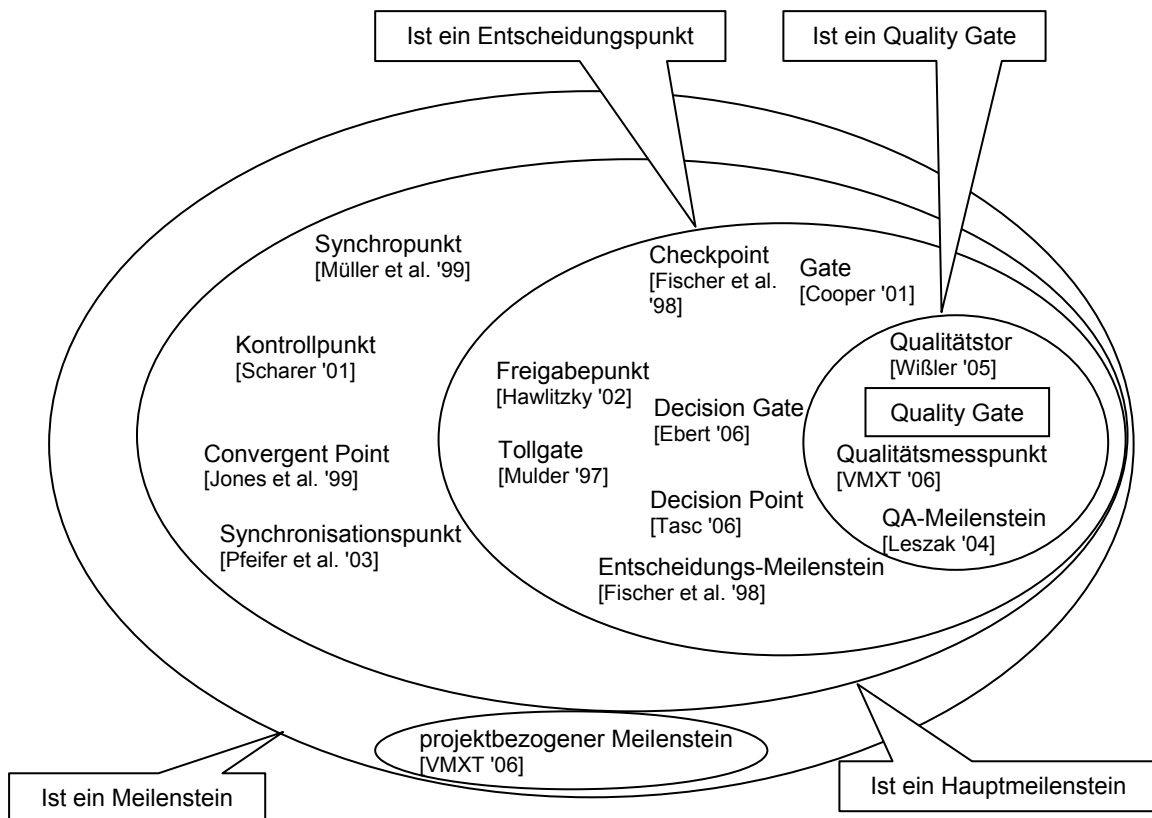


Abbildung 1: Verhältnis des Quality-Gate-Konzeptes zum Meilenstein-Konzept

Abbildung 2 zeigt analog zu Abbildung 1 das Verhältnis eines Quality Gates zum Review. Ein Quality Gate (oder genauer das zugehörige Gate-Review) kann als spezielles Management-Review aufgefasst werden, das auf die Prüfung qualitätsbezogener Kriterien fokussiert ist. Ein Management-Review ist wiederum ein spezielles Review und kann klar vom technischen Review abgetrennt werden. Insbesondere ist also ein Gate-Review kein technisches Review.

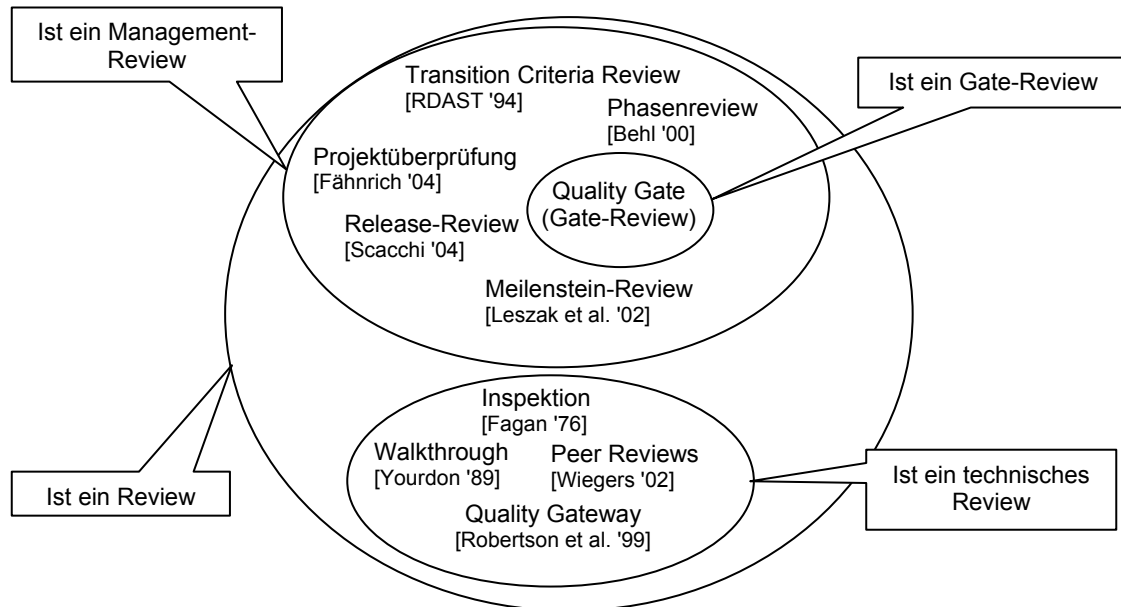


Abbildung 2: Verhältnis des Quality-Gate-Konzeptes zum Review-Konzept

Auf die Abgrenzung zum projektbezogenen Meilenstein und zum technischen Review wird näher in Abschnitt 4.8 eingegangen.

Die Ausgestaltung eines Quality-Gate-Referenzprozesses durch ein Software-Unternehmen gestaltet sich ohne Anleitung als Herausforderung. Eine derartige Anleitung existiert jedoch gegenwärtig nicht. Das Ergebnis dieser Arbeit ist eine Anleitung, auf dessen Basis das Prozessmanagement eines Software-Unternehmens einen Quality-Gate-Referenzprozess ausgestalten kann. Eine solche Anleitung gibt Aufschluss darüber,

- welche Aktivitäten im Rahmen von Quality Gates durchgeführt werden müssen,
- welche Dokumente benötigt werden,
- welche Rollen etabliert werden müssen,
- welche Hilfsmittel genutzt werden sollten,
- an welchen Stellen Quality Gates idealerweise im Entwicklungsprozess platziert werden und
- welches Spektrum, welche Richtlinien und Empfehlungen bei diesen Punkten jeweils bestehen.

1.2 Zielsetzung dieser Arbeit

Zielsetzung dieser Dissertation ist es, den theoretischen Unterbau von Quality-Gate-Referenzprozessen zu untersuchen und zu verfestigen. Die Ergebnisse können als Anleitung verwendet werden, die es dem Prozessmanagement eines Software-Unternehmens (im Folgenden auch kurz Unternehmen) ermöglicht, einen Quality-Gate-Referenzprozess auszugestalten. Diese Anleitung kann in Anlehnung an Software-Frameworks ebenfalls als Framework verstanden werden – jedoch selbstverständlich nicht, um auf Basis des Frameworks eine Software-Anwendung zu entwickeln, sondern um einen Quality-Gate-Referenzprozess abzuleiten. Das Framework bildet das zentrale Konzept und Ergebnis dieser Arbeit. Das Framework im Kontext dieser Dissertation ist kein fertiger und damit anwendbarer Quality-Gate-Referenzprozess. Vielmehr kann das Prozessmanagement eines Unternehmens auf Grundlage des Frameworks mit vertretbarem Aufwand einen Quality-Gate-Referenzprozess ausgestalten, der auf die spezifischen Anforderungen des jeweiligen Unternehmens hinsichtlich

- der Qualitätssicherung,
- des Ressourcenaufwands

- und der notwendigen Flexibilität im Kontext verschiedener Projektsituationen

zugeschnitten ist. Dabei werden Strukturen, Richtlinien, Empfehlungen und eine einheitliche Terminologie, jedoch auch Freiheiten der Ausgestaltung vorgegeben. Letztere ermöglichen den optimalen Zuschnitt auf die Bedürfnisse eines Unternehmens.

Der Nutzen des Frameworks für ein Unternehmen hängt von seiner Ausgangssituation ab. Diese Ausgangssituationen lassen sich wie folgt zusammenfassen:

1. Praxisanwender oder Forscher möchten Wissen und Erfahrungen zu Quality Gates austauschen. Allerdings fehlt ihnen ein Domänenmodell, das eine einheitliche Terminologie bereitstellt und somit den (wissenschaftlichen) Austausch erleichtert. Zusätzlich erleichtert eine Charakterisierung des Quality-Gate-Konzeptes eine Abgrenzung von scheinbar verwandten Konzepten. Das Framework beinhaltet daher eine einheitliche Terminologie und eine Charakterisierung.
2. Ein Unternehmen nutzt bereits einen Referenzprozess für die Software-Entwicklung, jedoch gibt es kein definiertes Vorgehen, das es dem Management erlaubt, bei qualitativen Problemen auf ein Projekt kontrollierend und steuernd einwirken zu können. Infolgedessen scheitern einige Projekte oder werden nur durch eine erhebliche Überschreitung von Zeit und Kosten zum Ziel geführt. Die Einführung eines Quality-Gate-Referenzprozesses auf Basis des Frameworks ist sinnvoll.
3. Ein Unternehmen setzt bereits einen Quality-Gate-Referenzprozess ein. Dennoch kommt es noch verhäuft zu Fehlschlägen. Mit Hilfe des Frameworks kann der verwendete Quality-Gate-Referenzprozess zunächst bewertet werden. Basierend auf identifizierten Schwächen kann der Quality-Gate-Referenzprozess auf Basis des Frameworks verbessert werden.
4. Ein Unternehmen hat bereits einen Quality-Gate-Referenzprozess etabliert, der nahezu vollständig und optimal ausgestaltet ist. In diesem Fall kann die Effizienz oder Effektivität möglicherweise noch verbessert werden. Beispielsweise können Softwarewerkzeuge eingesetzt werden, die die notwendige Sitzung eines Quality Gates von Zeit und Raum entkoppeln oder die Einplanung von Quality Gates erleichtern. Unter Umständen ist die Einführung eines Erfahrungskreislaufes sinnvoll, um den Quality-Gate-Referenzprozess eines Unternehmens auf längere Sicht hin noch weiter verbessern zu können.

Notwendige Bestandteile des Frameworks sind demnach:

- Eine *Charakterisierung* von Quality Gates und *Abgrenzung* zu scheinbar verwandten Konzepten.
- Ein *Domänenmodell*, das die Teilkonzepte von Quality-Gate-Referenzprozessen enthält und damit eine einheitliche Terminologie bereitstellt.
- *Richtlinien, Empfehlungen und Spektrum zur bzw. der Ausgestaltung* für jedes Teilkonzept, um auf Basis des Frameworks einen neuen Quality-Gate-Referenzprozess erstellen zu können oder aber einen vorhandenen Quality-Gate-Referenzprozess optimieren zu können.
- *Ansatzpunkte zur Verbesserung* der Effizienz und Effektivität eines vorhandenen Quality-Gate-Referenzprozesses.

Abbildung 3 fasst die Bestandteile des Frameworks und ihre Abhängigkeiten voneinander zusammen. Für die wissenschaftliche Untersuchung sind die Abhängigkeiten als Vorleistungs-Abhängigkeiten zu interpretieren. Für ein Unternehmen stellen sie die ideale Leseabfolge durch das Framework dar.

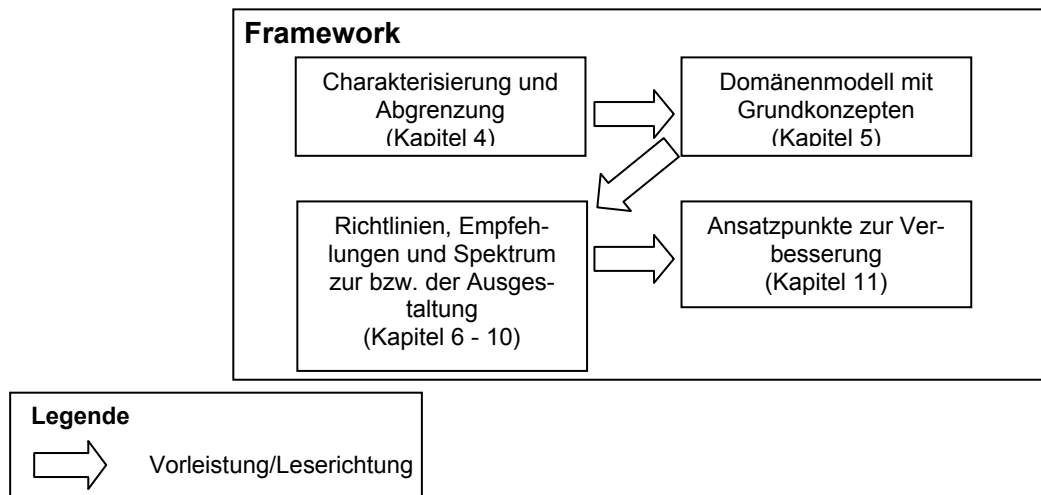


Abbildung 3: Bestandteile des Frameworks

Im Rahmen dieser Arbeit wird das Framework stufenweise vorgestellt. Dazu werden zunächst vorhandene Quality-Gate-Referenzprozesse betrachtet. Diese Betrachtung umfasst ebenfalls die Betrachtung von Quality-Gate-Referenzprozessen in anderen Domänen als der Software-Entwicklung. Da Quality-Gate-Referenzprozesse im Bereich der Software-Entwicklung nur wenig erforscht sind, aber dennoch in der Praxis vielfach eingesetzt werden, wurde als Teilleistung dieser Arbeit eine empirische Erhebung unter Software-Unternehmen mit dem Ziel des Erkenntnisgewinns durchgeführt.

In erster Näherung kann damit das Quality-Gate-Konzept charakterisiert und abgegrenzt werden. Bestehende Problempunkte konnten ermittelt werden, auf die das Framework gezielt durch Richtlinien und Empfehlungen eingehen kann. Durch die Befolgung der Richtlinien und Empfehlungen können die Probleme größtenteils abgemildert werden.

Weiterhin kann ein abstrahierendes Domänenmodell erstellt werden, das die Teilkonzepte von Quality-Gate-Referenzprozessen enthält. Basierend hierauf können Richtlinien, Empfehlungen und Spektrum zur bzw. der Ausgestaltung der Teilkonzepte formuliert werden. Eine Orientierung an den Quality-Gate-Referenzprozessen anderer Domänen ist hierbei nützlich, jedoch nicht ausreichend, da sonst die besonderen Schwierigkeiten der Software-Entwicklung nicht ausreichend berücksichtigt werden würden. Die Erarbeitung von Möglichkeiten zur Verbesserung eines Quality-Gate-Referenzprozesses stellt den letzten Beitrag dieser Arbeit dar und komplettiert das Framework.

1.3 Gliederung der Arbeit

Die vorliegende Arbeit gliedert sich wie folgt:

Die Kapitel 2 und 3 bilden die Grundlagenkapitel dieser Arbeit. Kapitel 2 beschäftigt sich mit allgemeinen Grundlagen, insbesondere mit Entwicklungsprozessen, Referenzprozessen und den Schwierigkeiten der Software-Entwicklung. In Kapitel 3 werden verschiedene Grundlagen des Managements der Software-Entwicklung erläutert, die zum weiteren Verständnis der Folgekapitel benötigt werden. Namentlich sind dies das Projekt-, Risiko- und Qualitätsmanagement.

Kapitel 4 bildet den Ausgangspunkt der Untersuchung von Quality-Gate-Referenzprozessen. Dazu werden sowohl Quality-Gate-Referenzprozesse aus der Literatur, als auch in verschiedenen Unternehmen praktizierte Quality-Gate-Referenzprozesse (letztere anhand einer im Rahmen dieser Arbeit durchgeführten empirischen Erhebung) näher betrachtet. Basierend darauf werden die wichtigsten Quality-Gate-Referenzprozesse weiter untersucht. Im weiteren Verlauf des Kapitels werden Quality Gates ebenfalls mit etablierten Prozessreifegradmodellen in Beziehung gesetzt. Das Kapitel endet mit einer Charakterisierung des Quality-Gate-Konzeptes, sowie mit einer Abgrenzung des Quality-Gate-Konzeptes vom Meilenstein- und Review-Konzept. Ein wichtiger zentraler Punkt des Kapitels ist die Identifikation von Problempunkten von Quality-Gate-Referenzprozessen – das Framework muss auf diese Problempunkte durch Richtlinien und Empfehlungen eingehen.

Kapitel 5 fasst die Teilkonzepte, die mit Quality Gates in Beziehung stehen, in einem abstrahierenden Domänenmodell zusammen. Kapitel 5 zeigt damit kompakt und übersichtlich, welche Teilkonzepte ein Unternehmen umsetzen muss, um einen Quality-Gate-Referenzprozess zu etablieren. Die Teilkonzepte lassen sich dabei in verschiedene Pakete aufteilen. Jedes der Pakete wird hinsichtlich der Ausgestaltung seiner Teilkonzepte in den nachfolgenden Kapiteln 6 (strukturelle Konzepte), 7 (inhaltliche Konzepte), 8 (Reviewkonzepte), 9 (Steuerungskonzepte) und 10 (Anpassungskonzepte) betrachtet. Die genauen Beziehungen dieser Kapitel zueinander sind in Abbildung 4 dargestellt.

Unternehmen, die bereits einen Quality-Gate-Referenzprozess etabliert haben, finden in Kapitel 11 mögliche Verbesserungsansätze, durch die sich ihr Quality-Gate-Referenzprozess effektiver und effizienter ausgestalten lässt. Insbesondere werden dabei die Etablierung von Erfahrungskreisläufen und der Einsatz von Softwarewerkzeugen betrachtet.

Kapitel 12 zeigt, wie das entwickelte Framework von Unternehmen eingesetzt werden kann. Dabei hängt die Art der Verwendung von der Ausgangslage des Unternehmens ab (vgl. Seite 5, Punkte 2 bis 4).

Um den Nutzen und die Mächtigkeit des Frameworks aufzuzeigen, werden in Kapitel 13 verschiedene Fallstudien betrachtet. Hierbei werden ausschließlich Projekte im studentisch-universitären Umfeld betrachtet. Sie weisen nicht alle Charakteristika von „realen“ Industrieprojekten auf, jedoch sind die Parallelen ausreichend um auf die „Realität“ übertragen werden zu können.

Die Arbeit schließt mit einer Zusammenfassung in Kapitel 14 ab. Hierbei wird auch darauf eingegangen, wie die identifizierten Problempunkte durch das Framework behandelt werden können. Kapitel 14 gibt des Weiteren einen Ausblick auf mögliche zukünftige Forschungsarbeiten.

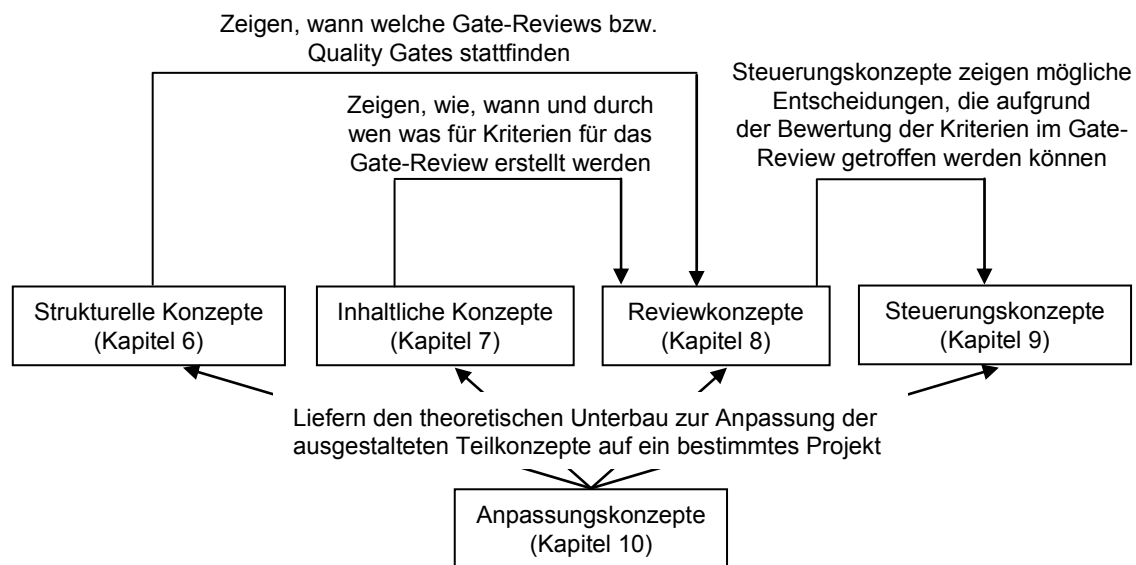


Abbildung 4: Beziehungen der Kapitel 6 bis 10 untereinander

2 Grundlagen der Software-Entwicklung

Die Einleitung verwendete in naiver Weise schon einige Begriffe, die in diesem Abschnitt noch etwas näher gefasst werden. Diese Begriffe werden in dieser Arbeit immer wieder verwendet werden, so dass eine Definition notwendig ist.

2.1 Software

Im allgemeinen Sprachgebrauch bezieht sich der Begriff *Software* fast ausschließlich auf ein ausführbares Programm. Software kann nach IEEE Standard (Std. 610.12-1990) [IEEEESG '90] jedoch deutlich weiter gefasst werden:

„Computer programs, procedures and possibly associated documentation and data pertaining to the operation of a computer system“

Software umfasst demnach auch die Dokumentation, Konfigurationsdateien und Daten, die zum Betrieb eines Computersystems nötig sind.

Software lässt sich unterscheiden nach *Individualsoftware* und *Standardsoftware*:

- *Individualsoftware* wird für einen bestimmten Kunden und damit für einen besonderen Anwendungsfall entwickelt.
- *Standardsoftware* wird für eine sehr große Anzahl von anonymen Anwendern erstellt und kann dann für einen einzelnen Anwendungsfall angepasst werden. Diesen Anpassungsvorgang bezeichnet man als *Customizing*. Standardsoftware ist daher sehr viel generischer auszurichten, so dass eine Anpassung für verschiedene Anwendungsfälle möglich ist.

2.2 Software-Entwicklung

Dieser Abschnitt wirft zunächst einen Blick auf die Grundlagen der Modellbildung. Modelle lassen sich in verschiedensten Bereichen finden. Die Hauptmotivation für die Modellbildung ist dabei stets, Untersuchungen am Modell durchzuführen, die am Original nicht möglich sind, weil sie unter Umständen zu teuer sind oder weil das Original nicht zugänglich ist. In dieser Arbeit sind Untersuchungen an Software-Prozessmodellen von herausragender Bedeutung.

2.2.1 Grundlagen der Modellbildung

Die Grundlagen der Modellbildung beschreibt Stachowiak in [Stachowiak '73]. Gemäß Stachowiak ist ein Modell ein abstrahierendes Abbild eines Originals (deskriptives Modell) oder ein Vorbild für ein Original (präskriptives Modell). Demnach werden bestimmte, für den Betrachter relevante Merkmale des Originals in das Modell abgebildet. Nicht relevante Merkmale werden hingegen nicht abgebildet (präterierte Merkmale). Dementsprechend bilden die Menge der präterierten Merkmale und die Menge der abgebildeten Merkmale die Gesamtmenge aller Merkmale des Originals. Andererseits verfügt das Modell jedoch über Merkmale, welche nicht Bestandteil des Originals sind. Diese Merkmale heißen abundante Merkmale. Letztere Merkmale erleichtern zumeist die Handhabbarkeit des Modells. Der Prozess der Erstellung eines Modells wird Modellbildung genannt.

Die oben genannten Beziehungen zwischen den Merkmalen des Originals und den Merkmalen des Modells werden in Abbildung 5 veranschaulicht. So ist zum Beispiel der Bauplan eines Gebäudes auf Papier ein präskriptives Modell des daraus zu erstellenden Gebäudes. In diesem Fall ist ein präteriertes Merkmal der Abbildung die Größe des Gebäudes. Die Verhältnisse zwischen den Größen einzelner Elemente des Gebäudes sind hingegen ein abgebildetes Merkmal. Ein abundantes Merkmal des Bauplans ist sein Material, nämlich Papier. Die Wahl, den Bauplan auf Papier zu drucken, ermöglicht es jedoch, den Plan leicht transportieren zu können (womit also seine Handhabbarkeit erhöht wird).

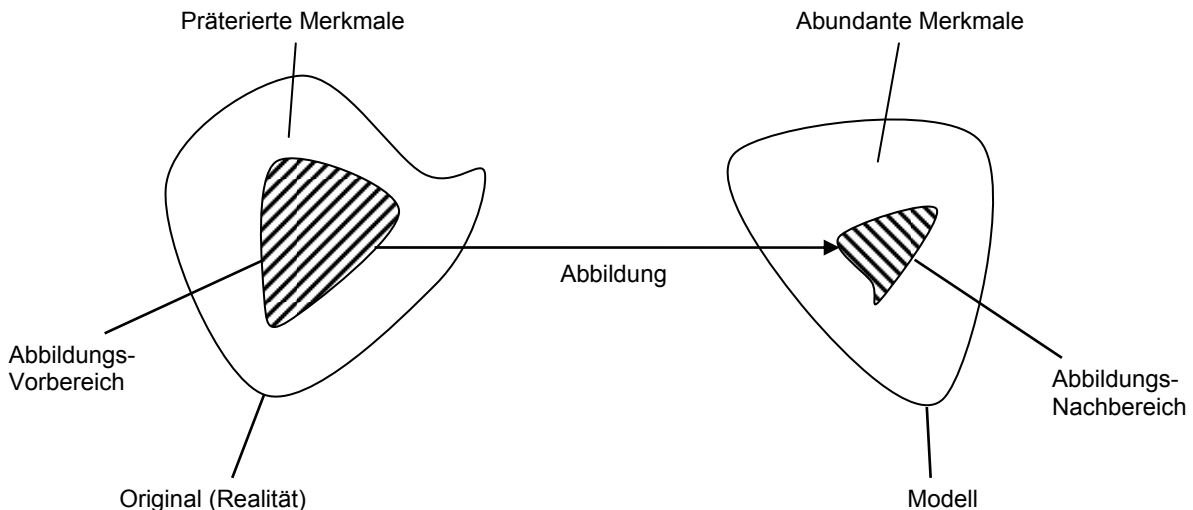


Abbildung 5: Abbildungs-Beziehung zwischen Original und Modell

Die Modellbildung erfolgt stets aus pragmatischer Sicht. Das heißt, die Abbildung vom Original erfolgt immer so, dass sie für den Nutzer des Modells nützlich ist. Ein Modell wird erst durch eine Zuordnung zu einem Original zu einem Modell. Diese Zuordnung kann über die folgenden fünf Fragen geschehen:

- Was ist das Original?
- Für wen ist das Modell?
- Zu welchem Zweck wurde modelliert, wozu wird das Modell verwendet?
- Wann wird das Modell verwendet?
- Welches sind die relevanten Merkmale im Modell und welchen Merkmalen entsprechen sie im Original?

Die vorhergehenden Erkenntnisse zur Modellbildung lassen sich in drei Modell-Eigenschaften zusammenfassen:

- **Die Abbildungseigenschaft:** Modell und Original werden durch ihre Merkmale beschrieben. Merkmale des Originals werden auf das Modell abgebildet, da ein Modell ein Abbild oder ein Vorbild für ein Original ist.
- **Die Verkürzungseigenschaft:** Modelle besitzen nur einen Teil der Merkmale des Originals. Insbesondere sind die Merkmale von Interesse, die für den Nutzer des Modells nützlich sind.
- **Die Pragmatismuseigenschaft:** Modelle werden stets zu einem bestimmten Zweck erstellt. Dieser Zweck kann durch die obigen fünf Fragen definiert werden.

2.2.2 Projekt

Ein *Projekt* ist ein einmaliges, komplexes und zielgerichtetes Vorhaben mit definiertem Anfangs- und Endtermin. Dabei ist die Einmaligkeit von besonderer Bedeutung. Sie trägt dazu bei, dass sich ein Projekt vom operativen Tagesgeschäft absetzt. Für ein Projekt sind stets limitierte Ressourcen vorgesehen, daher steht ein Projekt in Konkurrenz zu anderen Projekten. Ressourcen sind dabei Zeit, Personal, Räume, Werkzeuge, Materialien und daraus resultierend Geldmittel. Das Ziel eines Projektes ist es, ein neuartiges Problem zu lösen, wobei je nach Problemstellung unterschiedliche Methoden angewendet werden müssen. Daher besitzt ein Projekt immer ein gewisses Risiko, das heißt ein Erfolg ist nicht von vornherein garantiert. Allerdings ist das Risiko hinsichtlich eines Erfolges nicht so hoch, wie bei explorativen Forschungsvorhaben. Abbildung 6 illustriert das Verhältnis zwischen operativem Tagesgeschäft, einem Projekt und einem Forschungsvorhaben.

Ein Projekt involviert häufig Mitarbeiter verschiedener Fachabteilungen eines Unternehmens. Hughes und Cotterell [Hughes et al. '02] heben ferner die Größe und Komplexität eines Projektes, gegenüber dem operativen Geschäft hervor.

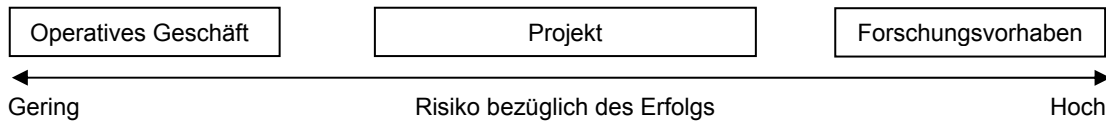


Abbildung 6: Einordnung eines Projektes hinsichtlich seines Risikos (nach [Hughes et al. '02])

Ein *Softwareprojekt* ist ein Projekt mit dem Ziel Software zu entwickeln. Dabei kann neue Software entwickelt werden oder eine bereits erstellte Software weiterentwickelt werden. Ein Softwareprojekt folgt einem mehr oder weniger definierten Software-Entwicklungsprozess. Im weiteren Vorlauf der Arbeit werden die Begriffe Projekt und Softwareprojekt synonym verwendet. An wenigen Stellen ist jedoch der allgemeine Projektbegriff gemeint. In diesen Fällen wird gesondert darauf hingewiesen.

Projekte können nach Zuser [Zuser et al. '01] unter anderem nach folgenden Projektmerkmalen klassifiziert werden:

- Die *Größe* des Projekts, gibt den Aufwand in Personenjahren und die Anzahl der beteiligten Mitarbeiter an.
- Die *Dauer* des Projekts in Kalenderwochen.
- Die *Zielsetzung* bestimmt die Erfolgskriterien eines Projektes.
- Die *Domäne* benennt die fachliche Ausrichtung des Projekts.
- Die *Technologien* definieren, welche Technologien im Projekt verwendet werden und beeinflussen den Realisierungsaufwand.
- *Ausgangsprodukte* und *Zielprodukte* bestimmen den Aufwand und den Projektverlauf.

Diese Projektmerkmale dienen dem Management dazu, kompakt Projekte miteinander vergleichen zu können oder einen passenden Entwicklungsprozess abzuleiten. Diese Idee der Klassifikation lässt sich wie folgt weiter verallgemeinern:

Jedes Projektmerkmal (auch kurz Merkmal) kann verschiedene Projektmerkmalswerte (auch kurz Merkmalswerte) annehmen – sie ergeben zusammen ein Projektmodell. Merkmale können hierbei wie Maße behandelt werden (vgl. Abschnitt 3.3.3), ihr Messobjekt ist immer ein Projekt. Ihre Merkmalswerte sind daher Elemente verschieden mächtiger Skalen (vgl. Tabelle 7). Eine Projektsituation beschreibt kompakt ein bestimmtes Projekt. Eine Projektsituation ist eine feste Belegung der einzelnen Merkmale mit Merkmalswerten. Die folgende Definition formalisiert Projektmodell und Projektsituation:

Definition: Seien m_1, \dots, m_n Merkmale und M_1, \dots, M_n die zugehörigen nichtleeren Mengen von Merkmalswerten. Ein *Projektmodell* P ist dann beschrieben durch das kartesische Produkt der Mengen von Merkmalswerten:

$$P = M_1 \times \dots \times M_n.$$

Eine *Projektsituation* p ist ein Element des Projektmodells:

$$p \in P.$$

Definition 1: Projektmodell und Projektsituation

Ein etabliertes Projektmodell bietet dem Management also die Möglichkeit an, kompakt und einheitlich Projekte beschreiben zu können, um sie mit anderen Projekten hinsichtlich ihrer Ausgangslage vergleichen zu können. Projektmodell und Projektsituation sind jeweils Modelle. Das Original zum Projektmodell sind daher alle vorhandenen und zukünftig durchzuführenden Projekte eines Unternehmens. Das Original zur Projektsituation ist ein konkretes Projekt.

Die Auswahl der Projektmerkmale und ihrer Merkmalswerte eines Projektmodells liegt dabei in der Hand des Managements. Im Sinne der Modelltheorie soll ein Projektmodell nur die relevanten Merkmale erfassen (Verkürzungseigenschaft), so dass das Management einen Nutzen daraus zieht (Pragmatismuseigenschaft). Welche Merkmale und Merkmalswerte dies also jeweils sind, hängt daher von der Situation des jeweiligen Unternehmens ab. Es gibt jedoch eine Reihe typischer Merkmale und Merkmalswerte. Diese werden in Abschnitt 10.5 näher erläutert.

2.2.3 Prozess und Prozessmodell

Ein Projekt folgt immer einen mehr oder weniger definierten Prozess. Nach Ittner [Ittner '05] ist ein Prozess wie folgt definiert:

Definition: Ein *Prozess* ist eine natürlich auftretende oder geplante Abfolge von Operationen und Ereignissen der realen Welt, die in den meisten Fällen durch den Einsatz von Zeit, Raum, Fachwissen oder weiteren Ressourcen ein Ergebnis erzeugt.

Definition 2: Prozess

Prozesse können durch Prozessmodelle beschrieben werden. Wendet man die Grundlagen der Modelltheorie auf Prozessmodelle an, so bedeutet dies, dass ein Prozess-Modell stets nur einen Teil des wirklichen Prozesses (des Original-Prozesses) darstellt. Die Modellierung erfolgt stets für einen bestimmten Betrachter (welcher einen bestimmten Nutzen daraus zieht) und ist auch nur für einen bestimmten Zeitpunkt gedacht. So kann ein Prozessmodell beispielsweise die an jeder Phase beteiligten Rollen zeigen, ein anderes Modell zeigt, welche Dokumente wo und wann erstellt und benötigt werden.

Ein Modell eines Prozesses ist entweder beschreibend oder vorschreibend und ist daher Abbild oder Vorbild eines Original-Prozesses. Ein beschreibendes Modell definiert einen Prozess, der bereits stattgefunden hat. Ziel ist es, den Prozess zu analysieren und zu verstehen und letztlich mit den daraus gewonnenen Erkenntnissen zu verbessern. Ein vorschreibendes Modell definiert, wie ein Prozess ausgeführt werden soll, also zu sein hat. Solche Modelle sind daher Handlungsanleitungen, um den Entwicklungsprozess zu organisieren und zu strukturieren.

2.2.4 Software-Entwicklungsprozess

Ein Software-Entwicklungsprozess (im Folgenden auch kurz Entwicklungsprozess) ist ein Prozess, der Software als Ergebnis liefert. Das Modell eines Software-Entwicklungsprozesses kann folgendermaßen definiert werden:

Definition: Das Modell eines *Software-Entwicklungsprozesses* ist eine Beschreibung, die definiert „wer was wann und wie tut“, um Software qualitativ hochwertig zu entwickeln [Kruchten '99].

Definition 3: Software-Entwicklungsprozess

Im Folgenden wird zwischen dem Modell eines Software-Entwicklungsprozesses und dem eigentlichen Software-Entwicklungsprozess (dem Original) nicht mehr unterschieden werden, da alle Untersuchungen in dieser Arbeit an Modellen von Software-Entwicklungsprozessen stattfinden. Wird also ein Software-Entwicklungsprozess untersucht, so wird eigentlich das Modell des jeweiligen Software-Entwicklungsprozesses untersucht.

Die Elemente eines Entwicklungsprozesses lassen sich nach Huff [Huff '96] unterteilen in Prozessschritte, Artefakte und Ressourcen.

Prozessschritte sind die wesentlichen Elemente eines Prozesses. Ein Prozessschritt selbst ist nicht atomar, sondern kann in kleinere Prozessschritte zerlegt werden. Einige dieser Prozessschritte auf der Zerlegungsebene können wiederum zerlegt werden. Dieser Zerlegungsvorgang lässt sich fortsetzen bis man auf eine Ebene gelangt, auf der alle Prozessschritte nur noch atomar sind. Welche Prozessschritte letztlich atomar sind, entscheidet der Ersteller des Entwicklungsprozesses. Es ist wichtig die richtige Granularität des Modells zu treffen, um den größtmöglichen Nutzen für den Verwender des Modells zu schaffen. Im Rahmen der Software-Entwicklung ist ein Aufbau wie in Abbildung 7 üblich.

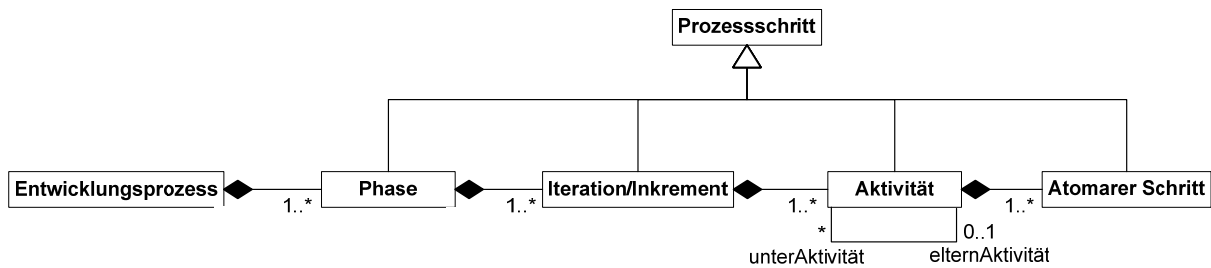


Abbildung 7: Hierarchischer Aufbau eines Entwicklungsprozesses (nach [SPEMSpec '05])

Die folgende Tabelle schlüsselt die Charakteristika der verschiedenen Prozessschritte auf.

Prozessschritt	Charakteristika
Phase	Phasen sind sequenziell angeordnet; Ergebnisse vorgehender Phasen sind bindend für nachfolgende Phasen; viele Personen/Rollen sind an einer Phase beteiligt; viele Ergebnisse werden geliefert und viele Artefakte werden dazu benötigt; üblicherweise existieren nur wenige Phasen (~3-10); jede Phase wird mit einem Hauptmeilenstein abgeschlossen.
Iteration/Inkrement	Werden beim iterativen oder inkrementellen Vorgehen mehrfach ausgeführt; viele Personen/Rollen sind beteiligt; werden durch kleinere Meilensteine abgeschlossen.
Aktivität	Nur eine Rolle ausführend beteiligt; nur wenige konkrete Ergebnisse; kommt sehr häufig vor; eine Aktivität kann in verschiedene Unteraktivitäten unterteilt werden.
Atomarer Schritt	Kann aus Sicht des Prozessmanagements nicht weiter zerlegt werden; nur eine Person beteiligt; Ergebnisse sind feingranular.

Tabelle 1: Charakteristika einzelner Prozessschritte

Artefakte sind physisch vorhandene Eingaben (also Dokumente) für Prozessschritte. Modelle von Artefakten beziehen sich auf die Struktur des jeweiligen Artefaktes, zum Beispiel auf das Datenschema. Ein Prozessschritt definiert Operationen auf diesen Artefakten, also was mit den Artefakten passiert. Artefakte können ebenso zerlegt werden wie Prozessschritte. Dementsprechend ergibt sich ebenso die Frage nach der Granularität wie bei Aktivitäten. Software ist ein Beispiel für ein zusammengesetztes Artefakt, da sie aus dem ausführbarem Programm, dem Quellcode, den Konfigurationsdateien und Daten besteht, welche wiederum Artefakte darstellen. Artefakte werden üblicherweise mit Metadaten versehen, welche das Artefakt näher beschreiben. Versionsnummer und Erstellungsdatum eines Artefakts sind typische Metadaten.

Ressourcen werden benötigt, um einen Prozessschritt ausführen zu können. Eine Ressource ist entweder eine Rolle oder ein Werkzeug. Im weiteren Sinne können Ressourcen auch Zeit, Räume, Materialien und Geldmittel sein. Eine Rolle kann zum Beispiel der Qualitätsbeauftragte eines Entwicklungsteams sein. Rollen werden mit Personen besetzt. Es ist möglich, mehrere Rollen einer Person zuzuweisen oder auch eine Rolle mit mehreren Personen zu besetzen. Zusätzlich ist es möglich, dass diese Zuweisung sich mit Fortschreiten des Prozesses ändert.

Hinsichtlich des Aufbaus lassen sich grob sequenzielle (wasserfallartige) und iterative Entwicklungsprozesse unterscheiden. In sequenziellen Entwicklungsprozessen wird eine Sequenz von Aktivitäten nur einmal durchlaufen, während in iterativen dies beliebig häufig passieren kann. Das Wasserfallmodell von Royce [Royce '98] ist das klassische Beispiel für eine sequenzielle Anordnung von Aktivitäten. Ein typischer Vertreter für einen iterativen Prozess ist Boehms Spiralmodell [Boehm '88].

2.2.5 Referenzprozess und Anpassung

Ein Software-Entwicklungsprozess wird aus einem Referenzprozess über den Vorgang der Anpassung abgeleitet. Ziel ist es dabei, den geeigneten Entwicklungsprozess für eine konkrete Projektsituation zu

erhalten, da unterschiedliche Projektsituationen verschiedenartige Software-Entwicklungsprozesse benötigen. Ein Referenzprozess lässt sich folgendermaßen definieren:

Definition: Ein *Referenzprozess* (oder auch Vorgehensmodell) r ist eine generische Beschreibung eines Entwicklungs- oder Entstehungsprozesses. Über den Vorgang der Anpassung (englisch Tailoring) lässt sich aus r ein Entwicklungs- oder Entstehungsprozess ableiten, der für eine konkrete Projektsituation p geeignet ist.

Definition 4: Referenzprozess

Beispiele für Referenzprozesse sind der *Rational Unified Process (RUP)* [Kruchten '99] und das V-Modell XT [VMXT '06]. In manchen Fällen deckt der Referenzprozess noch die Vorphase und die Einführungsphase mit ab. In diesem Fall kann von einem Referenzprozess für die Software-Entstehung gesprochen werden. In Abschnitt 3.1.4 wird hierauf näher eingegangen.

In Anlehnung an Ittner [Ittner '05] lässt sich der Vorgang der Anpassung weiter formalisieren. Die Anpassung arbeitet auf den auswählbaren Elementen des Referenzprozesses. Ein auswählbares Element ist folgendermaßen definiert:

Definition: Ein auswählbares Element a eines Referenzprozesses r ist ein Prozess-Element, das im Rahmen der Anpassung entweder gewählt oder ausgelassen werden kann. Vereinfacht ausgedrückt, kann a in einem aus r abgeleiteten Entwicklungsprozess e vorhanden sein oder nicht. Auswählbare Elemente können Prozessschritte, Artefakte und Ressourcen gemäß [Huff '96] sein.

Definition 5: Auswählbares Element

Der Vorgang der Anpassung lässt sich in Anlehnung an Ittner nun weiter formalisieren:

Definition: Sei $A = \{a_1, \dots, a_n\}$ die Menge der im Referenzprozess r definierten auswählbaren Elemente. Sei $C = \{c_1, \dots, c_k\}$ die Menge der nicht auswählbaren Elemente. Eine Anpassungsregel f_{a_i} entscheidet auf Grundlage einer Projektsituation $p \in P$, ob ein auswählbares Element a_i im Entwicklungsprozess vorhanden sein sollte oder nicht:

$$f_{a_i} : P \rightarrow B$$

$$f_{a_i} : p \mapsto \begin{cases} true, & \text{falls } a_i \text{ zu } p \text{ passt} \\ false, & \text{falls } a_i \text{ nicht zu } p \text{ passt.} \end{cases}$$

Betrachtet man die Anpassungsregeln für alle auswählbaren Elemente $a_i \in A$ für eine gegebene Projektsituation p , so erhält man zusammen mit den nicht auswählbaren Elementen einen Entwicklungsprozess e . Dies kann als Funktion $e(p)$ aufgefasst werden, die einen geeigneten Entwicklungsprozess für eine gegebene Projektsituation zurückliefert:

$$e : P \rightarrow C \cup A$$

$$e : p \mapsto C \cup \left\{ a_i \in A \mid f_{a_i}(p) = true \right\}.$$

Definition 6: Anpassung

Der Vorgang der Anpassung ist keineswegs trivial und lässt sich daher meistens nicht einfach durch Regeln ausdrücken. Vielmehr sind dafür Erfahrungen notwendig, die nur zum Teil durch Regeln formalisiert werden können. Darüber hinaus sind diejenigen Elemente des Referenzprozesses zu identifizieren, die überhaupt auswählbar sind. Oftmals wird man auf Heuristiken oder die Intuition zurückgreifen müssen, um eine Aussage darüber zu treffen, ob ein auswählbares Element in einem Entwicklungsprozess zu verwenden ist oder nicht. Dieser Vorgang ist nicht wiederholbar und mit Unsicherheit behaftet – zwei gleiche Projektsituationen könnten daher zu ungleichen Entwicklungsprozessen und damit zu verschiedener Qualität der Ergebnisse des Projektes führen. Andererseits ist die Anpassung ohne formalisierte Regeln flexibler, insbesondere wenn eine Projektsituation eventuell nicht alle rele-

vanten Merkmale eines konkreten Projekts einfängt. In diesem Fall ist es möglich, dass zwei Projekte trotz gleicher Projektsituation unterschiedlichen Entwicklungsprozessen folgen müssen.

2.3 Die besonderen Schwierigkeiten der Software-Entwicklung

Die Entwicklung von Software unterscheidet sich im großen Maße von der Entwicklung anderer (durch Ingenieure gefertigter) technischer Güter [Humphrey '92, Schneider '07a]:

- **Entwicklung statt Produktion:** Software wird nur entwickelt, danach findet keine aufwändige Produktion statt. Diese Entwicklung ist intellektuell anspruchsvoll und individuell, da Entwickler häufig neue Lösungen für stets neue Aufgabenstellungen finden müssen. Aus diesem Grund kann nicht immer auf Standardlösungen zurückgegriffen werden, wie dies bei anderen industriell gefertigten Produkten häufig der Fall ist.
- **Unstetigkeit beim Testen:** Anders als andere Produkte, weist Software beim Testen ein unstetiges Verhalten auf. Ohne Zweifel kann beispielsweise ein Personenaufzug, welcher vier und sechs Personen transportieren kann, auch fünf Personen sicher transportieren. Hingegen muss Software auf alle Eingaben hin geprüft werden, weil zum Beispiel fünf eine Primzahl ist und daher eine besondere Regel im Programm greifen könnte. Dies führt im schlimmsten Fall zu einer kombinatorischen Explosion beim Testen, da alle möglichen Eingaben getestet werden müssen. Besonders hier sind also systematische Testverfahren gefragt, die einerseits effektiv, andererseits auch ressourcenschonend (also effizient) sind.
- **Software ist überall:** Software findet in vielen unterschiedlichen Anwendungsbereichen Verwendung: in Hochleistungsrechnern, in eingebetteten Systemen, im Anwendungsbereich oder in verteilten Web-Anwendungen. Die Umgebung der Software ist also stets komplex und es sind häufig die Einschränkungen und Unzulänglichkeiten externer Schnittstellen zu berücksichtigen. So kann ein externer Farbsensor bei unterschiedlichen Lichtverhältnissen verschiedene Farbwerte für gleiche Farben liefern. Diese Einschränkung muss dann häufig durch Software aufgehoben werden. Teilweise sind auch Funktionen durch Software zu realisieren, die früher durch Hardware realisiert wurden.
- **Flexibilität von Software:** Eine weitere Schwierigkeit bei der Entwicklung von Software ist ihre Flexibilität. Software gehorcht (im Gegensatz zu anderen Baustoffen) keinen physikalischen Gesetzen [Brooks '95]. Software besitzt insbesondere keine inhärente Struktur. Daher kann Software auf verschiedene Weisen konstruiert werden und leider ist nicht offensichtlich, ob die eine oder andere Bauweise besser oder schlechter zur Lösung eines gegebenen Problems geeignet ist. Insbesondere ist daher eine offensichtliche Lösung für eine Problemstellung nicht immer gegeben.

Humphrey [Humphrey '92] sieht folgende weitere Unterschiede bei der Entwicklung von Hardware und Software:

- **Komplexität:** Software ist weitaus komplexer als Hardware.
- **Änderbarkeit:** Änderungen der Software erscheinen einfach im Gegensatz zu Änderungen an der Hardware, da es keinen Produktionsprozess gibt.
- **Integrationsrolle:** Software dient häufig als Vermittler zwischen verschiedenen Systemen. Ändern sich die beteiligten Systeme, so muss dies meist durch Software ausgeglichen werden.
- **Sichtbarkeit:** Software ist für den Nutzer eines Systems meistens sichtbarer als die (zumeist verborgene) Hardware. Daher muss stets ein besonderes Augenmerk auf die Bedienbarkeit (engl. Usability) gelegt werden.
- **Fehlende Erfahrung:** Als noch relativ junge Disziplin verfügt die Softwaretechnik noch nicht über stabile und effektive Entwicklungsprozesse mit entsprechender Qualitätssicherung, so wie sie bei der Entwicklung von Hardware und in der Serienproduktion verwendet werden.

2.4 Die Modellierungssprache SPEM

Zur Modellierung der Software-Entwicklungsprozesse dieser Arbeit wird das *Software Process Engineering Metamodel* (kurz SPEM) [SPESpec '05] verwendet. In vielen Bereichen der Wissenschaft und Praxis werden unterschiedliche (teils nicht standardisierte) Prozess-Notationen angewendet, weshalb eine gemeinsame grundlegende Prozess-Notation für diese Arbeit wichtig ist. Auf diese Weise wird die Vergleichbarkeit gesichert. Im Rahmen dieser Arbeit mussten daher die verschiedenen Prozess-Notationen mittels SPEM übersetzt werden.

SPEM ist ein so genanntes UML-Profil, also eine semantische Erweiterung des Sprachumfangs der *Unified Modeling Language* [UMLSpec '01] (*UML*). Ziel eines UML-Profiles ist es, die UML selbst für eine spezifische Domain anzupassen. In diesem Fall also für die Domain der Software-Entwicklungsprozesse, wobei viele Konzepte der UML einfach vererbt werden. Der Vorteil ist, dass sich Prozessmanager, die bereits mit der UML vertraut sind, sich auch in SPEM schnell zurechtfinden und daher leicht Prozesse modellieren können.

SPEM bietet verschiedene Diagrammtypen an, von denen in dieser Arbeit lediglich das Aktivitäts- und das Anwendungsfalldiagramm genutzt werden. Mit einem Aktivitätsdiagrammen kann die Beziehung zwischen Prozessschritten mit ihren ein- und ausgehenden Artefakten und jeweils verantwortlichen Rollen modelliert werden. Durch ein Anwendungsfalldiagramm kann die Beziehung zwischen Rollen und Prozessschritten modelliert werden.

SPEM verfügt über verschiedene eigene Stereotypen, von denen die relevanten Stereotypen in Tabelle 2 zusammengefasst werden. Darüber hinaus wird SPEM in dieser Arbeit um den Stereotyp Quality Gate ergänzt.




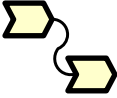




Stereotyp	Notationselement	Beschreibung
Dokument		Ein Dokument ist ein schriftlich niedergeschriebenes Artefakt.
Rolle		Eine Rolle ist verantwortlich für die Erstellung eines Artefaktes innerhalb einer Aktivität. Eine Rolle kann dabei ausführend (<i>perform</i>) oder assistierend (<i>assist</i>) mitwirken.
Prozessrolle		Eine Prozessrolle fasst verschiedene Rollen zusammen, die an bestimmten „größeren“ Prozessschritten (Phasen und Iterationen/Inkrementen) beteiligt sind.
Prozessschritt		Oberklasse von Phase, Iteration/Inkrement, Aktivität und atomarer Schritt.
Phase		Phasen werden stets sequenziell angeordnet. Eine Phase kann aus mehreren Iterationen/Inkrementen zusammengesetzt sein.
Iteration/Inkrement		Eine Iteration/ein Inkrement ist eine Folge von Aktivitäten, die mindestens einmal, in der Regel jedoch mehrfach durchlaufen wird.
Aktivität		Eine Aktivität wird von einer Rolle ausgeführt, jedoch können beliebig viele Rollen an ihr assistierend mitwirken. Eine Aktivität setzt sich aus beliebig vielen atomaren Schritten zusammen.
Quality Gate (zusätzlicher Stereotyp)		Ein Quality Gate wird wegen des verbundenen Gate-Reviews als spezielle Aktivität aufgefasst.

Tabelle 2: Stereotypen in SPEM

3 Grundlagen des Managements der Software-Entwicklung

Der Begriff Management hat zwei Bedeutungen:

1. Management als Menge von führenden Aktivitäten.
2. Das Management als institutionalisierte Einheit innerhalb eines Unternehmens, welche für diese Aktivitäten verantwortlich ist.

Die Entwicklung von Software umfasst nicht nur die rein technische Realisierung, sondern auch im größeren Umfang Management-Aufgaben. Es existieren folgende phasenübergreifende Management-Aufgaben, die eine wesentliche Rolle im Zusammenhang mit Quality Gate spielen:

- **Projektmanagement:** Projektmanagement beschäftigt sich u.a. damit, welche Rollen entscheidungsbefugt sind und daher den Fortsetzung eines Projektes beeinflussen können. Zur Entscheidungsfindung sind verschiedene Projektbesprechungen notwendig. Ein Gate-Review lässt sich ebenfalls als eine besondere Projektbesprechung einordnen. Quality Gates können ferner als Instrument des Multiprojektmanagements eingesetzt werden (vgl. Abschnitt 1.1, grundlegende Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie*).
- **Risikomanagement:** Quality-Gate-Kriterien können so gebildet oder gewichtet werden, dass mit ihnen bekannten Risiken entgegengewirkt werden kann.
- **Qualitätsmanagement:** Quality Gates sind ein Mittel der projektübergreifenden Qualitätssicherung. Die Erstellung von Quality-Gate-Kriterien erfordert ein systematisches zielorientiertes Vorgehen (z.B. über Qualitätsmodelle oder über die GQM-Methode). Ebenso erfordert die Prüfung der Kriterien ein systematisches Gate-Review.

In den folgenden Abschnitten werden die drei genannten Management-Aufgaben in dem Umfang beschrieben, der für das weitere Verständnis dieser Doktorarbeit notwendig ist.

3.1 Projektmanagement

Das Projektmanagement innerhalb der Software-Entwicklung beschäftigt sich mit allen Tätigkeiten, welche die Planung, Kontrolle und Steuerung von Projekten betreffen [Zuser et al. '01].

Nach Zuser [Zuser et al. '01] umfasst die Projektplanung drei Aspekte:

- **Technische Planung:** Dies umfasst die Auswahl eines geeigneten Entwicklungsprozesses. Ferner sind (Zwischen-)Produkte zu definieren, die im Verlauf des Projektes zu erstellen sind. Die Planung unterliegt zunächst keinen wirtschaftlichen Rahmenbedingungen, sondern versucht die technischen Herausforderungen in den Griff zu bekommen und damit das Risiko so gering wie möglich zu gestalten.
- **Qualitätsplanung:** Die Einhaltung von Qualitätszielen setzt die Einplanung geeigneter Maßnahmen wie Inspektionen [Fagan '76] oder Tests voraus.
- **Wirtschaftliche Planung:** Die wirtschaftliche Planung umfasst die Einplanung von Ressourcen, vorrangig von Personal sowie die dafür anfallenden finanziellen Mittel.

Bei der Planung eines Projektes muss das Projektmanagement den Spagat zwischen den vier Größen Kosten, Termin, Qualität und Umfang bewältigen [Houdek '06]. Dieses Spannungsfeld zwischen diesen vier Größen wird in Abbildung 8 illustriert. Dabei stellt die weiße quadratische Fläche das Projekt dar, dessen „Fläche“ stets konstant bleibt. Verändert man also eine Größe positiv, so beeinflusst man mindestens eine der drei anderen Größen negativ. Müssen beispielsweise Kosten eingespart werden, so kann dies durch eine Verminderung der Qualitätssicherungs-Maßnahmen geschehen (Abbildung 9).

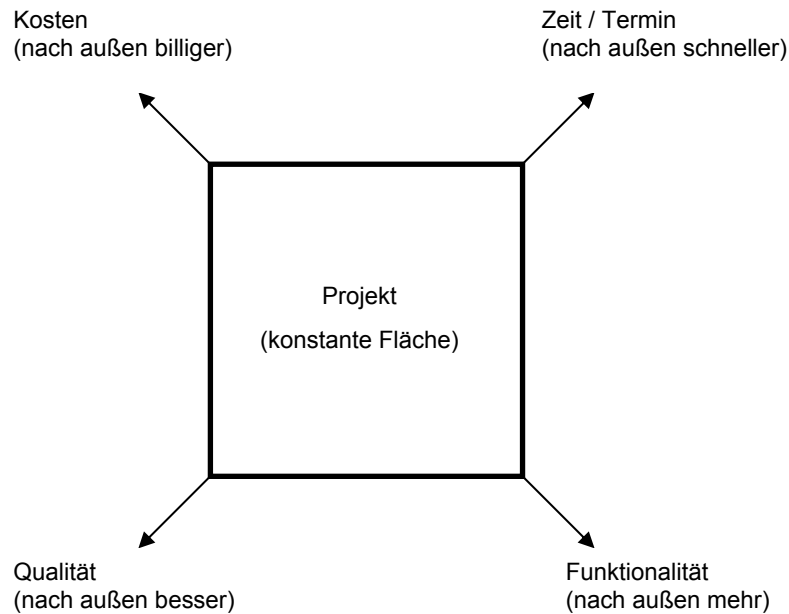


Abbildung 8: Das Projekt im Spannungsfeld verschiedener Größen (nach [Houdek '06])

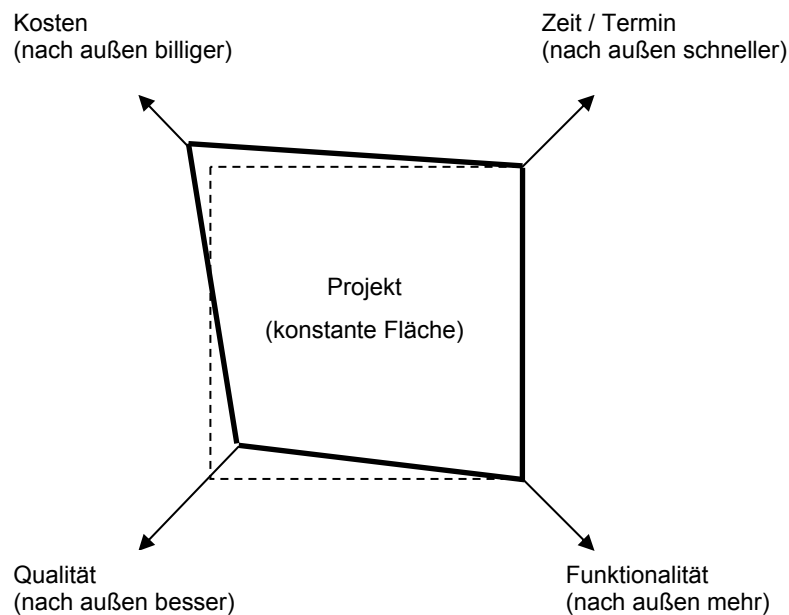


Abbildung 9: Das Projekt im Spannungsfeld verschiedener Größen (Verringerung der Kosten) (nach [Houdek '06])

3.1.1 Projektorganisation und Rollen

Ein Projekt ist auf bestimmte Weise in ein Unternehmen eingebettet. Abbildung 10 illustriert diesen Sachverhalt [Houdek '06]. Mitunter kann die in Abbildung 10 illustrierte Projektorganisation auch tiefer oder flacher ausfallen. Von besonderer Bedeutung ist die Rolle des Projektleiters, welcher die Verbindung zwischen Projektmitarbeitern und dem Projektmanagement bildet. Das Projektmanagement ist dabei am Ist-Zustand des Projektes interessiert und lässt Vorgaben und Ziele in das Projekt einfließen. Das Projektmanagement kommuniziert den Ist-Zustand in geeigneter Form der Geschäftsleitung und empfängt wiederum Vorgaben und Ziele von dieser.

Ein Kunde kann intern oder extern sein, dabei gehört ein interner Kunde dem Projekt selbst an, während ein externer Kunde außerhalb des Projektes steht. Interne Kunden sind beispielsweise Entwickler,

die mit bereits erstellten Produkten anderer Entwickler innerhalb des gleichen Projektes weiterarbeiten. Externe Kunden können klassische Kunden im Sinne eines Auftraggebers außerhalb des Unternehmens sein. Ebenfalls kann der externe Kunde ein Auftraggeber innerhalb des gleichen Unternehmens sein. Kunden definieren Anforderungen und fordern (Zwischen-)Produkte, die diese Anforderungen erfüllen müssen.

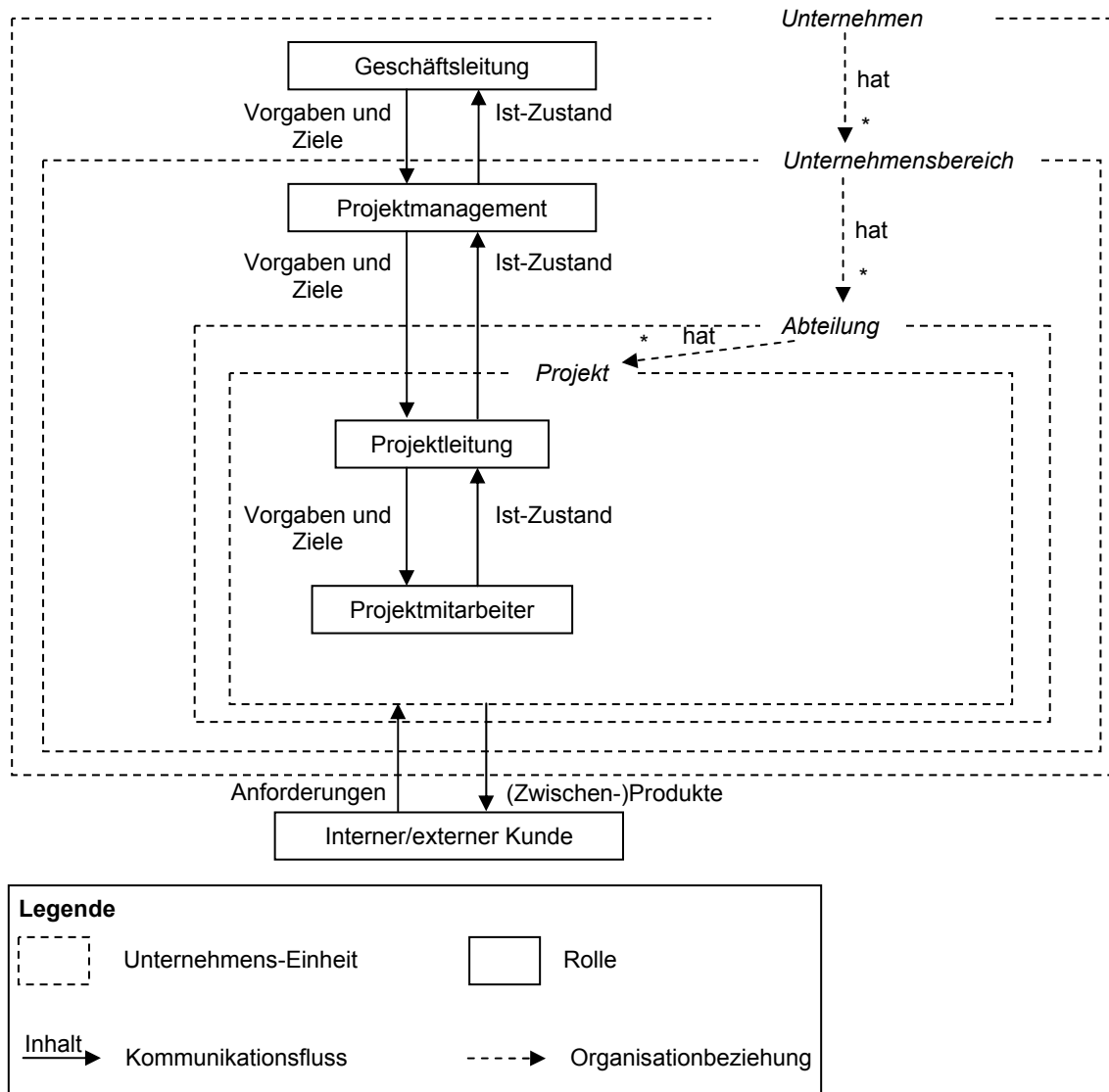


Abbildung 10: Umfeld eines Projekts (nach [Houdek '06])

Innerhalb der Projektorganisation können im Wesentlichen drei zentrale Rollen unterschieden werden:

- **Projektmanager:** Der Projektmanager ist verantwortlich für das Management der Rahmenbedingungen mehrerer Projekte (eines Unternehmensbereichs). Er bildet die Schnittstelle zwischen den Projektleitern dieser Projekte und der Geschäftsleitung. Die Rolle ist mit einer Person mit besonderen Führungsqualitäten zu besetzen.
- **Projektleiter:** Der Projektleiter ist für die operative Leitung eines Projektes verantwortlich. Er plant, koordiniert, überwacht und steuert den Ablauf seines Projektes und des zugehörigen Projektteams. Projektergebnisse sind von ihm zu beobachten und gegebenenfalls Nachbesserungen einzufordern. In seiner Verantwortlichkeit liegen ebenfalls die technische Umsetzung und die Realisierung der Ergebnisse. Er liefert regelmäßig und falls notwendig auch außerplanmäßig Berichte über den Status des Projektes an das Projektmanagement. Demnach besitzt er Kenntnisse in der Projektabwicklung, der Betriebswirtschaft und des Faches.

- Lenkungsausschuss:** Der Lenkungsausschuss ist ein Gremium, das weitreichende Entscheidungen über den Fortgang eines Projekt fällen darf. Üblicherweise gehören dem Lenkungsausschuss die betroffenen Projektleiter, -manager und je nach Relevanz des Projektes auch Mitglieder der Geschäftsleitung an. Gegebenfalls kann auch ein externer Kunde dem Lenkungsausschuss angehören.

3.1.2 Verhältnis von Prozess und Projekt

Abbildung 11 zeigt auf abstrakte Weise den Zusammenhang zwischen Prozess und Projekt. Das Management eines Unternehmens beauftragt das Prozessmanagement damit, den Referenzprozess zu erstellen oder den bereits vorhandenen Referenzprozess zu pflegen. Das Prozessmanagement kann als Teil des Qualitätsmanagements angesehen werden und hat die Aufgabe, den Referenzprozess eines Unternehmens zu gestalten und zu pflegen.

Ein Projektteam leitet aus dem Referenzprozess über den Vorgang der Anpassung einen geeigneten Prozess für eine gegebene Projektsituation ab. Dieser Prozess dient dann als Grundlage des Projektplans, der den Prozessschritten des Entwicklungsprozesses Termine zuordnet, die Rollen personell besetzt und Dokumente konkretisiert und als Ergebnisse von Meilensteinen festlegt. Im weiteren Verlauf des Projektes obliegt die Aufgabe der Pflege des Projektplans dem Projektleiter des jeweiligen Projektes.

Erfahrungen, die während oder nach der Projektdurchführung gesammelt werden, fließen zurück und können im Rahmen der Prozessverbesserung in den Referenzprozess eingepflegt werden.

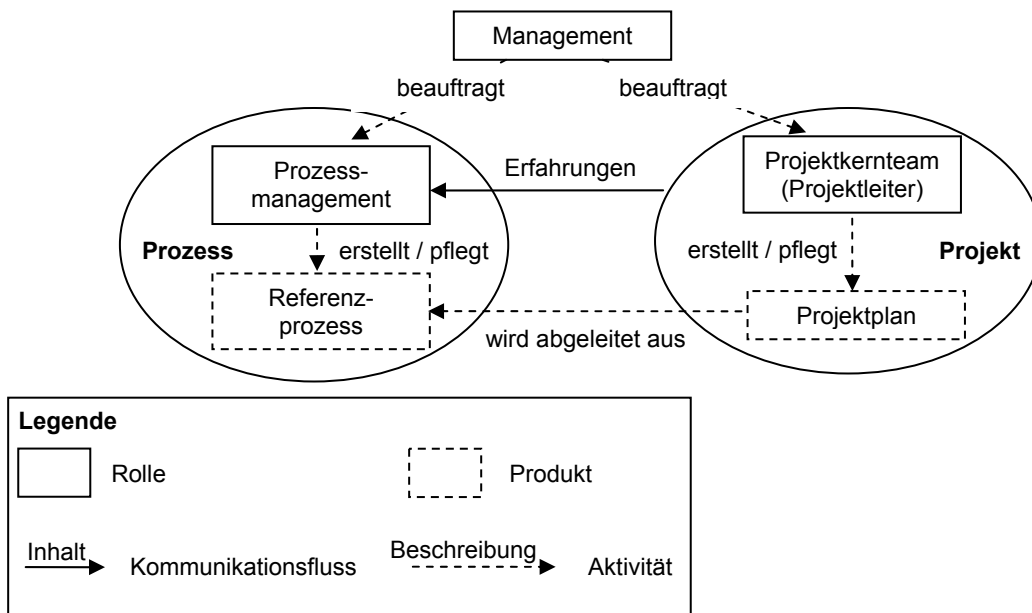


Abbildung 11: Zusammenhang zwischen Prozess und Projekt (nach [Hindel et al. '06])

3.1.3 Planungsaktivitäten

Ein Planungsprozess für ein Projekt untergliedert sich typischerweise grob in die drei Phasen Projektstart, Planungsphase und Durchführungsphase (Abbildung 12). Diese drei Phasen unterscheiden sich sowohl in der Granularität der Planungsergebnisse, als auch in der Intensität der Planungsaktivitäten. Innerhalb der Phase Projektstart wird mit mittlerer Intensität ein Grobplan erstellt, während in der Planungsphase mit hoher Intensität ein relativ detaillierter Projektplan erstellt wird. Während der Durchführungsphase, in der die eigentliche Entwicklung stattfindet, erfolgt lediglich eine detaillierte Planung der nächsten Vorgehensschritte („Planung im Kleinen“, z.B. Planung der nächsten Iteration).

Die Hauptziele der Phase Projektstart liegen in der Festlegung von Projektzielen sowie in der Prüfung der technischen und wirtschaftlichen Machbarkeit und der Prüfung der Vermarktungsaussichten innerhalb einer Vorstudie. Bei positiver Prüfung wird im Rahmen der Projekteinrichtung ein Projektteam aufgesetzt, welches unter anderem den Entwicklungsprozess für das jeweilige Projekt festlegt,

indem ein Referenzprozess auf Basis der Projektsituation ausgewählt und angepasst wird. Das Projektkernteam setzt sich in der Regel aus dem betroffenen Projektmanager, dem Qualitätsmanager, dem zukünftigen Projektleiter und dem zukünftigen Qualitätsbeauftragten des Projektes zusammen.

Dieser Entwicklungsprozess kann im späteren Projektverlauf noch in gewissem Rahmen angepasst werden. Die Entscheidung über den ausgewählten und angepassten Prozess wird neben anderen Kern-daten im Projekthandbuch festgehalten. Das Projektkernteam bestimmt ein Planungsteam, das die weiteren Planungsaktivitäten durchführen soll. Das Planungsteam setzt sich dabei idealerweise aus dem Projektleiter und dem Qualitätsbeauftragten des jeweiligen Projektes, sowie aus weiteren Rollen (mit speziellen Fachkenntnissen) zusammen. Der Qualitätsbeauftragte hat dafür Sorge zu tragen, dass qualitätssichernde Maßnahmen bei der Planung berücksichtigt werden.

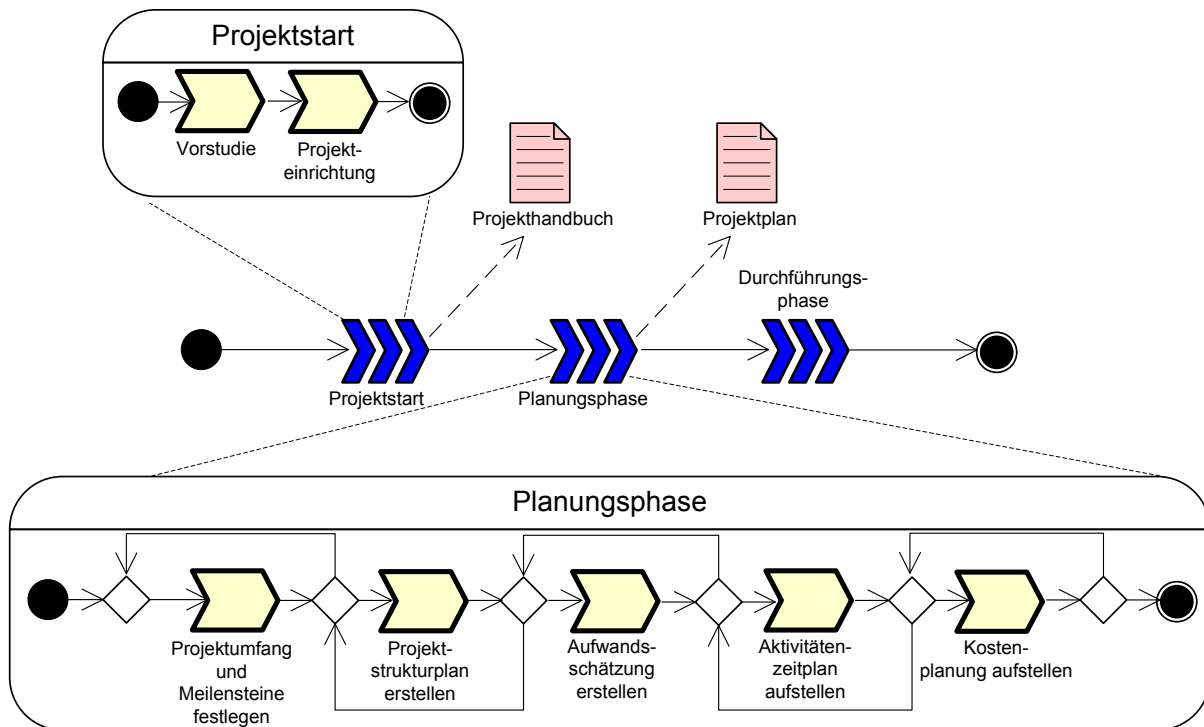


Abbildung 12: Planungsaktivitäten (nach [Hindel et al. '06])

Die Projektplanung folgt dem Projektstart und beinhaltet fünf verschiedene Aktivitäten (Abbildung 12), die nicht notwendigerweise sequenziell durchlaufen, sondern vielmehr eine Rückkopplung erfordern. Die Aktivitäten der Planungsphase sind beendet, wenn der Projektplan erarbeitet wurde, der als Planungs-Baseline für die Durchführungsphase dient. Die Planungs-Baseline liefert das Soll für die innerhalb Projektfortschrittskontrolle. Im Folgenden sollen die fünf Aktivitäten der Planungsphase nach Hindel et. al [Hindel et al. '06] kurz aufgeschlüsselt werden:

- Die Aktivität *Projektumfang und Meilensteine festlegen* umfasst die Festlegung des Projektumfangs in Form einer Projektdefinition, die Ziele, Rahmenbedingungen, Liefergegenstände und einen groben Zeitplan enthält. Weiterhin werden wesentliche Ergebnisse des Projektes als Liefergegenstände von Meilensteinen definiert. Innerhalb einer formalen Prüfung wird das Ist mit Planungs-Soll im *Meilenstein-Review* verglichen und daraus das weitere Vorgehen abgeleitet.
- Innerhalb der Aktivität *Projektstrukturplan erstellen* wird eine hierarchische Struktur aller Liefergegenstände des Projektes erarbeitet. Die obersten Ebenen des Projektstrukturplans bezeichnen die wesentlichen Liefergegenstände oder aber auch die Phasen, während untere Ebenen diese Liefergegenstände verfeinern. Die unterste Ebene enthält die Arbeitspakete des Projektes. Die identifizierten Arbeitspakete werden als Ergebnisse von Meilensteinen definiert.
- Die Aktivität *Aufwandschätzung erstellen* befasst sich mit der systematischen Schätzung des Aufwands der einzelnen Arbeitspakete.

- Die Aktivität *Aktivitätenzeitplan festlegen* umfasst die Bestimmung der Abhängigkeiten der einzelnen Aktivitäten untereinander, die Zuordnung von Ressourcen und Terminen, sowie das Einbringen der bereits definierten Arbeitspakete in den Zeitplan.
- Innerhalb der Aktivität *Kostenplanung aufstellen* werden die Kosten für notwendige Ressourcen aufgestellt, wobei die wesentlichen Kosten meistens durch das benötigte Personal verursacht werden.

Wie bereits angedeutet, müssen unter Umständen auch Änderungen des Projektplans in der Durchführungsphase vorgenommen werden, da dort erst im Rahmen der Projektfortschrittskontrolle Soll-Ist-Vergleiche durchgeführt werden und damit Handlungsbedarf ersichtlich ist. Planungsänderungen lassen sich unterscheiden in [Hindel et al. '06]:

- *Detailplanungen*, die aufgrund eines iterativen oder inkrementellen Vorgehens in der Regel erst vor Start der nächsten Iteration möglich sind. Die grobe Planung liegt durch die Planung der Phasen und wesentlichen Meilensteine in diesem Fall bereits vor.
- *Kleine Planungsänderungen*, bei denen die Zustimmung des Projektleiters ausreicht und keine Zustimmung des Managements erfolgen muss. Dies sind insbesondere solche Änderungen, die zu keiner oder nur zu einer geringfügigen Terminverschiebung führen können.
- *Größere Planungsänderungen*, die die Funktionalität oder Qualität des Produktes signifikant negativ beeinflussen oder zu deutlichen Terminverschiebungen oder Erhöhung der Kosten führen.
- *Neuplanung*, die bei einer erheblichen Fehlplanung erforderlich ist.

3.1.4 Die Entwicklung als Teil des Software-Entstehungsprozesses

Die eigentliche Entwicklung ist nur ein Teil des Software-Entstehungsprozesses. Der eigentlichen Entwicklung ist in den allermeisten Fällen noch eine Machbarkeitsstudie (zum Beispiel in Form eines Vorprojektes) [Hughes et al. '02], Ideenphase oder Marktstudie [Cooper '01] oder Vertragsphase vorangestellt. Sie bilden zusammen die Vorphase. Ideenphase und Marktstudie sind insbesondere bei Eigenentwicklungen sinnvoll. Bei Projekten mit externen Kunden sind diese beiden Aktivitäten in aller Regel schon abgeschlossen, bevor ein Lastenheft an einen potentiellen Auftragnehmer herangetragen und in die Vertragsphase übergegangen wird.

Eine vorangestellte Machbarkeitsstudie prüft, ob ein Unternehmen die notwendigen Ressourcen besitzt, um ein Softwareprojekt erfolgreich abschließen zu können. Die Ideenphase dient der Sammlung von Ideen für neue Produkte, während eine Marktstudie diese Produktideen auf ihre Erfolgchancen hin evaluiert – also mit anderen Worten gesprochen fragt: „Lohnt sich das Projekt?“.

Im Anschluss an die Entwicklung wird die Software in eine Organisation oder den Markt eingeführt. Je nach Ansicht ist dies eine Entwicklungsaktivität oder nicht. Für einen gewissen Zeitraum ist es noch möglich, die im Betrieb aufgetretenen Fehler im Rahmen der Entwicklung zu beheben und neue Anforderungen und Änderungswünsche im Rahmen des Change Managements einfließen zu lassen, insbesondere dann, wenn dies vertraglich vereinbart wurde. Irgendwann gilt das Projekt jedoch als abgeschlossen. Die Beseitigung dann auftretender Fehler gehört ab hier zu den operativen Aktivitäten. Neue Anforderungen und Änderungswünsche können operativ oder über ein neues Projekt abgewickelt werden.

Abbildung 13 veranschaulicht die Einbettung der Entwicklung in den gesamten Software-Entstehungsprozess.



Abbildung 13: Der Software-Entstehungsprozess

3.1.5 Projektfortschrittskontrolle und Steuerung

Die Projektfortschrittskontrolle beschäftigt sich mit dem Vergleich zwischen dem Planungs-Solls (das durch die Planungs-Baseline vorgegeben wird) und dem Ist innerhalb eines Projektes. Von Interesse sind dabei vorrangig Zeit, Kosten, Qualität, Funktionalität und Risiken. Abweichungen von Soll und Ist signalisieren dann Handlungsbedarf in Form von Planänderungen. Die Fortschrittskontrolle findet auf zwei Ebenen statt:

- **Auf Aktivitätsebene:** Hierbei berichten die Entwickler der nächst höheren Hierarchieebene über den Fortschritt. Bei einer flachen Hierarchie kann die nächst höhere Ebene bereits durch den Projektleiter gebildet werden. Innerhalb der Aktivitätsebene werden auch die Rohdaten erhoben, die für die verdichteten Daten für die Projektebene nötig sind.
- **Auf Projektebene:** Hierbei „verlassen“ verdichtete Daten das Projekt und werden dem Management, dem Kunden oder in letzter Instanz dem Lenkungsausschuss oder der Geschäftsleitung sichtbar gemacht.

Die Schnittstelle zwischen beiden Ebenen bildet der Projektleiter des jeweiligen Projekts, der dem unteren Management zugeordnet wird. Abbildung 14 fasst die beschriebenen Beziehungen zwischen den Ebenen und Rollen zusammen. Je nach Höhe der Abweichung oder strategischer Wichtigkeit des Projektes muss ein Projekt entweder auf Aktivitätsebene oder Projektebene gesteuert werden. Hohe Abweichungen (bei geringer strategischer Wichtigkeit) oder hohe strategische Wichtigkeit (bei geringerer Abweichung) erfordern die Eskalation zum oberen Management oder in letzter Instanz zum Lenkungsausschuss oder zur Geschäftsleitung.

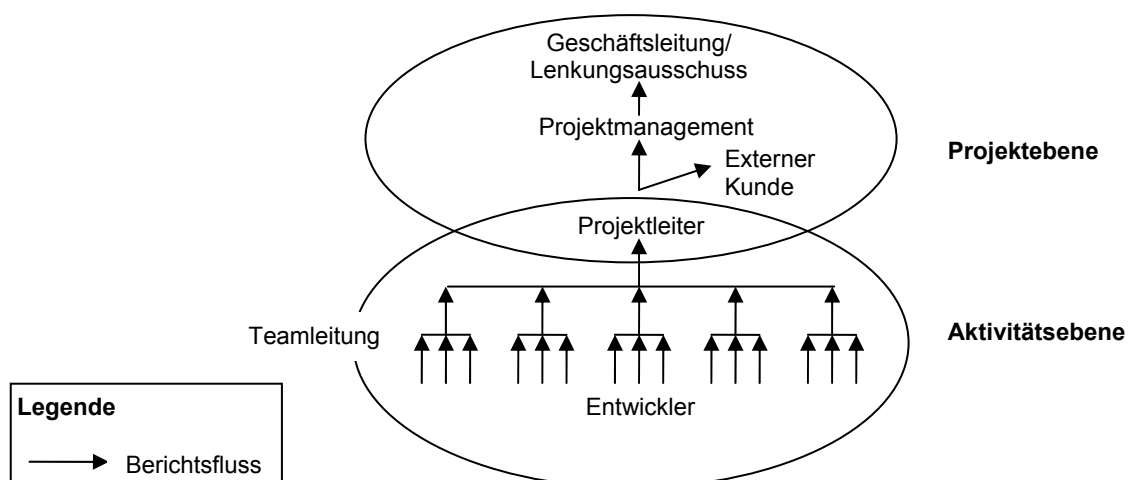


Abbildung 14: Berichtswesen zwischen Management und Projekt (nach [Hindel et al. '06])

Die Aufgaben des Projektmanagements lassen sich als Regelkreis bestehend aus den Aktivitäten Planung, Kontrolle und Steuerung auffassen (vgl. Abbildung 15). Innerhalb der Planung wird ein Soll festgelegt und im Idealfall in einem Planungsdokument fixiert. Die Kontrolle beschäftigt sich mit dem Vergleich zwischen Soll und Ist (das aus den Ergebnissen bestimmt wird) und der Risikoanalyse. Bei Abweichungen zwischen Soll und Ist hat das Projektmanagement die Möglichkeit steuernd auf die Durchführungsphase einzuwirken. Abweichungen, Risiken und steuernde Maßnahmen werden in einem Protokoll festgehalten und wirken sich auf die Planung aus.

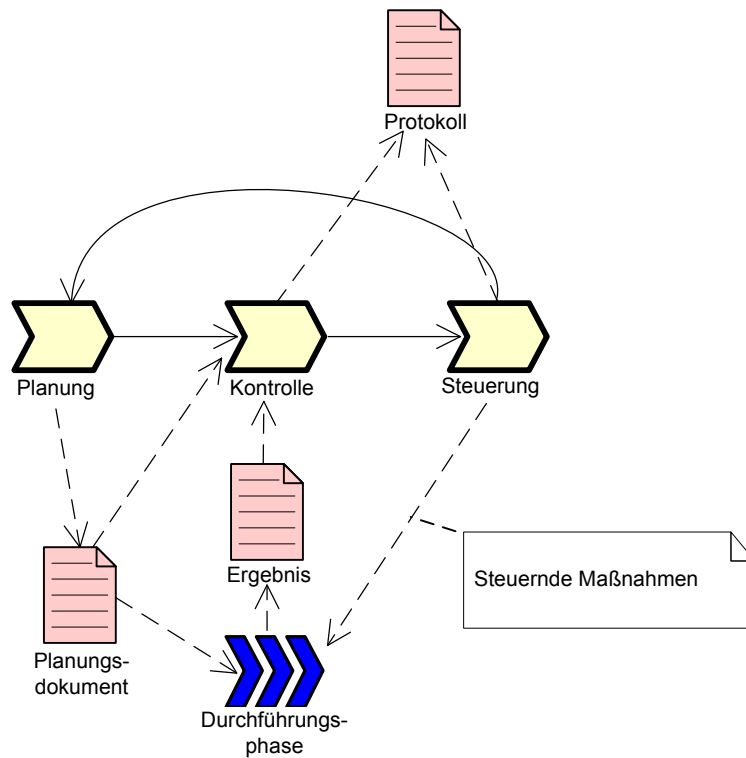


Abbildung 15: Projektmanagement-Regelkreis (nach [VMXT '06])

3.1.6 Projektbesprechungen

Auf jeder Hierarchieebene sind Projektbesprechungen durchzuführen. Innerhalb der Projektbesprechungen wird ein Projekt kontrolliert und über Maßnahmen entschieden. Es kann nach internen und externen Projektbesprechungen unterschieden werden. Interne Besprechungen finden auf der Aktivitätsebene statt und involvieren daher auch nur Rollen dieser Ebene. Externe Besprechungen finden auf der Projektebene statt. Tabelle 3 zeigt verschiedene Projektbesprechungen. Erkennbar ist, dass die Häufigkeit von Besprechungen mit höher werdender Hierarchieebene abnimmt.

Sichtbarkeit	Art der Besprechung	Zweck und Inhalt	Beteiligte Rollen	Typische Häufigkeit
Intern	Teambesprechung, technische Projektbesprechung	(Technische) Probleme und Maßnahmen auf Aktivitätsebene, Fortschritt im Kleinen kontrollieren	Projekt- bzw. Teamleiter, Entwickler	Wöchentlich
	Interne Fortschrittsbesprechung	Fortschrittsvergleich Soll und Ist, Maßnahmen, ausgewählte wichtige Probleme, Risiken, Change Requests	Projekt- bzw. Teamleiter, Entwickler	Monatlich
Extern	Formale Fortschrittsbesprechung	Fortschrittsvergleich Soll und Ist, Besprechung mehrerer Projekte parallel Werden nur kurz behandelt: ausgewählte wichtige Probleme, Maßnahmen, Risiken	Projektleiter und Projektmanager	Regelmäßig, üblicherweise Abstand > Monat
	Meilenstein-Review	Fortschrittsvergleich Soll und Ist, Prüfung formaler Voraussetzungen für die Freigabe der Ressourcen für den nächsten Prozessschritt, Prüfung der wichtigsten Risiken	Projektleiter und Projektmanager, eventuell Kundenvertreter	Beim Erreichen eines Meilensteins
	Projektsteuerkreis bzw. Lenkungsausschuss	Rechenschaftsbericht, ausgewählte Probleme, strategische Fragen, Koordination verschiedener Interessen, wichtige Projektfortschrittsentscheidungen	Projektleiter, Projektmanager, Entscheidungsträger (mit weitreichende Befugnissen), Kundenvertreter	Bei Eskalation

Tabelle 3: Ebenen von Projektbesprechungen (nach [Hindel et al. '06])

3.1.7 Multiprojektmanagement

Das Multiprojektmanagement befasst sich mit der gleichzeitigen Planung, übergreifenden Kontrolle und Steuerung mehrerer (möglicherweise von einander abhängiger) Projekte [Wikipedia '07]. Das Multiprojektmanagement lässt sich in zwei Teilbereiche unterscheiden:

- **Programmmanagement:** Dabei handelt es sich um das Management von verschiedenen Projekten, die inhaltlich zusammenhängen. Die Hauptaufgaben liegen in der Koordinierung der Projekte und in der Verteilung von Ressourcen unter ihnen.
- **Projektportfoliomanagement:** Ziel des Projektportfoliomanagements ist es, Projekte hinsichtlich ihrer Attraktivität zu bewerten. Dabei sind Ressourcen den attraktiveren Projekten zuzuteilen und die unattraktiveren Projekte abzubrechen oder zurückzustellen. Der wesentliche Unterschied zu den Projekten innerhalb des Programmmanagements besteht darin, dass die Projekte in Konkurrenz zu einander stehen.

Das Programmmanagement ist eine zeitlich befristete Aufgabe und endet mit dem Abschluss aller Projekte des Programms. Projektportfoliomanagement ist hingegen eine fortlaufende Aufgabe, deren Zielsetzung die Auswahl der attraktivsten Projekte ist.

Ein effizientes und effektives Multiprojektmanagement erfordert eine übersichtliche und kompakte Darstellung von Kennzahlwerten aller relevanten Projekte, so dass die Kontrolle aller dieser Projekte erleichtert wird. Kennzahlwerte werden dabei häufig in plakativen Indikatoren dargestellt. Gängig sind hierbei Indikatoren, die Ampeln, Druckmessern oder Thermometern ähneln und die Signalfarben rot, gelb und grün bei bestimmten Kennzahlwerten anzeigen. Derartige Indikatoren werden in einem so genannten Dashboard zusammengefasst, das zentral und kompakt die Situation aller relevanten Projekte zeigt. Abbildung 16 zeigt exemplarisch ein Dashboard. In der linken Spalte ist ein Ampel-Indikator für jedes Projekt aus der dritten Spalte zu sehen. Weiterhin kann durch verschiedene Piktogramme auf verschiedene Probleme in einem Projekt hingewiesen werden (erste Spalte). Auf diese Weise kann beispielsweise auf die Überschreitung eines Termins oder der Kosten hingewiesen werden. Dashboards sind vor allem bei Befolgung der grundlegenden Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* nützlich, wenn sie so konzipiert sind, dass sie den Projektfortschritt verschiedener Projekte hinsichtlich der passierten Quality Gates zeigen (vgl. auch 11.5.2).


Indicators	Proj #.	Project Name	Project Type	Project State	At Risk	Project Managers
		▼Implementation	Implementation			
   	19	Implementation Project	Implementation	Active	No	Project Insight, John Project
		▼Marketing	Marketing			
	21	Company Brochure	Marketing	Active	No	Project Insight, Sally Subco
		▼Professional Services	Professional Services			
	23	New Product Development Project	Professional Services	Active	No	John Project, Project Insight
	22	Development	Professional Services	Active	No	John Project, Project Insight
		Report Total:				

Abbildung 16: Beispiel für ein Dashboard (aus [Projektinsight '08])

3.2 Risikomanagement

Ein Risiko ist ein Problem, das möglicherweise in der Zukunft eintreten kann und dessen Auswirkungen die Projektergebnisse wahrscheinlich negativ beeinflussen können. Risiken bedrohen die Eckpfeiler des Projektes: Kosten, Zeit, Qualität und Funktionalität.

Nach Hindel et al. [Hindel et al. '06] ist das Risikomanagement ein kontinuierlicher und projektbegleitender Prozess, der als Teilaufgabe des Projektmanagements angesehen werden kann und Risiken aktiv entgegenwirkt. Das Risikomanagement beschäftigt sich mit den folgenden Aktivitäten:

- Die Risikoermittlung, die sich aus der
 - Identifikation von Risiken
 - und der Analyse, Bewertung und Priorisierung von Risiken zusammensetzt.
- Die Konzipierung und Einleitung von Gegenmaßnahmen.
- Die Verfolgung von Risiken im Projektverlauf.

Im weiteren Sinne kann auch die Identifikation, Bewertung und Verfolgung von Chancen als Aufgabe des Risikomanagements angesehen werden. Chancen ergeben sich durch eine günstigere Entwicklung des Projektes, beispielsweise wenn Ergebnisse früher als geplant vorliegen.

3.2.1 Risikomanagement nach Phasen

Bereits in der Phase Projektstart kann eine erste Risikoermittlung stattfinden. Falls die Risiken hier zu hoch erscheinen, kann das Projekt noch – ohne Gefahr eines zu hohen Verlustes – abgebrochen werden. Weitere Risikoermittlungen finden nach dem Projektstart und der eigentlichen Planungsphase statt, da zu diesen Zeitpunkten bereits genauere Informationen über das Projekt vorliegen (z.B. zu risikobehafteten Arbeitspaketen). Während der Durchführungsphase wird das Risiko an den Meilensteinen ermittelt, da hier bewertbare Ergebnisse und damit weitere Informationen vorliegen. Des Weiteren

werden während der Durchführung identifizierte Risiken und eventuelle Gegenmaßnahmen verfolgt. Abbildung 17 zeigt die Risikoermittlung nach Phasen.

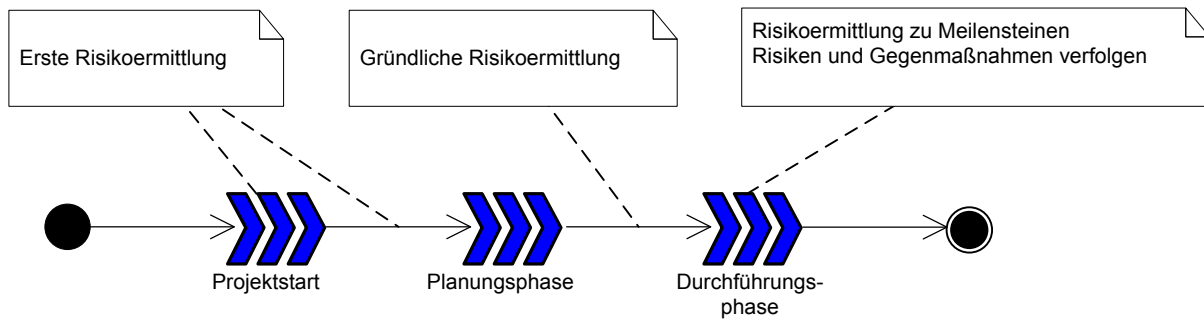


Abbildung 17: Risikoermittlung nach Phasen (nach [Hindel et al. '06])

3.2.2 Risikoermittlung

Die Risikoermittlung umfasst die Identifikation, Analyse, Bewertung und Priorisierung von Risiken. Die Identifikation sollte unter Einbeziehung von Projektbeteiligten stattfinden, dazu bietet sich beispielsweise ein Workshop an. Meistens ist bereits ein Risikokatalog vorhanden, der Aufschluss über mögliche Risiken gibt. Der Katalog orientiert sich dabei an abstrakten Risiken oder ist bereits organisationspezifisch befüllt.

Innerhalb der Analyse werden identifizierte Risiken des Katalogs für das aktuelle Projekt interpretiert. Anschließend wird eine Bewertung des Risikos vorgenommen. Hierfür sind die Wahrscheinlichkeit des Eintretens p und die Schadenshöhe d jedes Risikos entscheidend. Das Produkt der beiden Größen ergibt die Risikokennzahl r :

$$r = p \cdot d.$$

Da Wahrscheinlichkeit und Schadenshöhe häufig subjektiv und umgangssprachlich umschrieben sind, ist der Einsatz von Charakterisierungstabellen ratsam (vgl. Tabelle 4 und Tabelle 5).

Wahrscheinlichkeitsintervall	Interpretation
$0 \leq p \leq 0,25$ (gering)	Eintreten unwahrscheinlich.
$0,25 < p < 0,5$ (möglich)	Problem wird eher nicht eintreten.
$0,5 \leq p \leq 0,75$ (hoch)	Problem wird wahrscheinlich eintreten.
$0,75 < p \leq 1$ (sehr hoch)	Problem wird höchstwahrscheinlich eintreten.

Tabelle 4: Charakterisierung der Eintretenswahrscheinlichkeit (nach [Hindel et al. '06])

Die Schadenshöhe kann in den bekannten Dimensionen Kosten, Termin, Qualität oder Funktionalität beschrieben werden. Meistens ist mehr als eine der Dimension betroffen. Für jede der Dimensionen kann eine Intervallskala gebildet werden, auf der sich zum Beispiel die Schadenshöhe mit Werten von 0 bis 10 bewerten lässt (vgl. Tabelle 5). Betrifft der Schaden mehrere Dimensionen, so werden die einzelnen Schadenshöhen addiert.

Schadenshöhe	Schaden			
	Kosten	Termin	Funktionalität	Qualität
0	Kein Schaden	Kein Schaden	Kein Schaden	Kein Schaden
1-2 (gering)	Überschreitung der Projektkosten $k \leq 5\%$	Verzögerung aufgetreten, Termin kann gehalten werden	Einschränkungen kaum bemerkbar.	Einschränkungen kaum bemerkbar.
3-5 (spürbar)	Überschreitung der Projektkosten $5 < k \leq 15\%$	Überschreitung der Projektdauer $d \leq 10\%$	Einschränkungen spürbar, wesentliche Funktionen jedoch vorhanden, Produkt nutzbar.	Einschränkungen spürbar, verschiedene Qualitätsaspekte wurden verletzt. Produkt jedoch nutzbar.
6-8 (hoch)	Überschreitung der Projektkosten $15 < k \leq 30\%$	Überschreitung der Projektdauer $d \leq 20\%$	Es fehlen wesentliche Funktionen, Software kann nur eingeschränkt genutzt werden.	Qualitätsaspekte stark verletzt. Software nur stark eingeschränkt nutzbar. (Prototypstadium)
9-10 (sehr hoch)	Überschreitung der Projektkosten $k \geq 30\%$	Überschreitung der Projektdauer $d > 20\%$	Software ist unbrauchbar.	Software ist unbrauchbar.

Tabelle 5: Beispiel für die Charakterisierung des Schadens (nach [Hindel et al. '06])

Auf Basis beider Charakterisierungstabellen lässt sich nun die Risikokennzahl für ein gegebenes Risiko bilden. Zum Beispiel lässt sich die Risikokennzahl für das Risiko, dass eine Anforderungsspezifikation nicht alle funktionalen Anforderungen beinhaltet, folgendermaßen bewerten:

Wahrscheinlichkeit	Kosten	Termin	Funktionalität	Qualität
0,3	6	4	7	0
Aus den obigen Werten ergibt sich die Risikokennzahl r :				
$r = p \cdot d = 0,3 \cdot (6 + 4 + 7 + 0) = 5,1$				

Tabelle 6: Beispiel für die Berechnung der Risikokennzahl

Die Risikokennzahl gibt an, wie wichtig ein Risiko genommen werden muss. Da es zu aufwändig wäre, für alle Risiken Gegenmaßnahmen zu treffen, müssen Risiken nach ihrer Risikokennzahl priorisiert werden. Eine weitere gängige Herangehensweise ist die Bildung eines Risikoportfolios, in das jedes Risiko nach Wahrscheinlichkeit des Eintretens und Schadenshöhe eingetragen wird. Von besonderem Interesse sind dabei Risiken mit höherer Eintretenswahrscheinlichkeit und höherem Schaden (vgl. Abbildung 18).

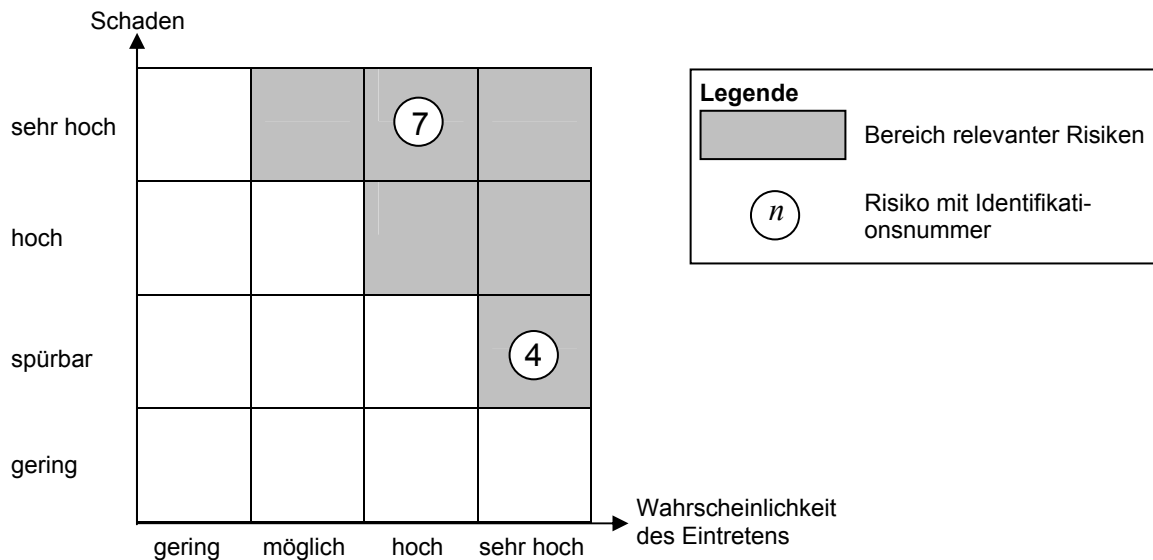


Abbildung 18: Beispiel eines Risikoportfolios

3.2.3 Gegenmaßnahmen und Verfolgung

Gegenmaßnahmen des Risikomanagements lassen sich nach Hindel et al. [Hindel et al. '06] unterscheiden:

- Nach präventiven Maßnahmen, die das Eintreten des Problems verhindern oder zumindest die Wahrscheinlichkeit des Eintretens reduzieren.
- Nach Notfallmaßnahmen, die im Falle des Eintretens greifen. Dazu gehört eine Verringerung des Schadens durch die Nutzung von Alternativlösungen.

Gegenmaßnahmen erfordern Ressourcen und vergrößern somit vor allem die Kosten im Spannungsfeld der Größen.

Der Erfolg von Gegenmaßnahmen ist genauso zu verfolgen, wie die weiteren Ergebnisse und Aktivitäten des Projektes. Dazu gehören vor allem aufgewendete Ressourcen und der damit erzielte Erfolg. Inhaltlich sind Risiken in internen Fortschrittsbesprechungen monatlich zu besprechen. Die formale Überwachung kann in der formalen Fortschrittsüberwachung oder zu Meilenstein-Reviews stattfinden.

3.3 Qualitätsmanagement

Zunächst soll auf den Begriff „Qualität“ eingegangen werden. Die DIN-Norm 55350 Teil 11 definiert Qualität als die

"Gesamtheit von Eigenschaften und Merkmalen eines Produktes oder einer Tätigkeit bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen."

Dieser Qualitätsbegriff spiegelt unter anderem die Sicht auf die Qualität aus Sicht des Kunden bzw. des Endanwenders wider. Problematisch ist die Ausdehnung des Qualitätsbegriffs auf die vorausgesetzten Erfordernisse, da diese nicht offensichtlich sind. Der Qualitätsbegriff unterliegt diesbezüglich einem Bedeutungswandel. Eine ältere Definition umfasst diese Erfordernisse nicht, sondern umschreibt Qualität als die Übereinstimmung der Software mit seiner Spezifikation [Crosby '79]. In dieser Arbeit wird jedoch die Definition nach DIN-Norm 55350-11 verwendet, da Software-Spezifikationen selten vollständig sind und überdies einige Anforderungen schwer zu greifen sind [Sommerville '07]. Selbst wenn eine Software also seine Spezifikation erfüllt, kann die Software nach DIN-Norm 55350-11 (aus Sicht des Kunden) qualitativ mangelhaft sein.

Qualität kann sich auf einen Prozess oder ein Produkt beziehen. Eine etablierte Annahme ist, dass ein „guter“ Prozess auch positiven Einfluss auf die Qualität der Software hat [Sommerville '07]. Die Qualität von Prozessen kann durch die Einstufung auf Reifeskalen stattfinden, die durch Prozess-Reifegradmodelle wie *SPICE* (Software Process Improvement and Capability Determination) [Hörmann et al. '06] oder *CMMI* (Capability Maturity Model Integration) [CMMI '06] vorgegeben wird.

Qualität ist ein Kostenfaktor in zweierlei Hinsicht:

- Das Hineinkonstruieren von Qualität verursacht Kosten.
- Die Fehlerbehebung und -analyse verursacht Kosten.

Als Faustregel kann festgestellt, dass das Hineinkonstruieren meistens weniger Kosten verursacht als die spätere Behebung von Unzulänglichkeiten. Das heißt, dass früh im Prozess begangene Fehler sich später hinsichtlich der Kosten um ein Vielfaches „rächen“ [Frühauf et al. '04].

Das Qualitätsmanagement ist für die Qualität von Prozess und Produkt verantwortlich und hat folgende Aufgaben [Hindell et al. '06]:

- Gestaltung von optimalen Prozessen und Verbesserung dieser. Folglich kann das Prozessmanagement, das ja diese Aufgaben wahrnimmt, als Teil des Qualitätsmanagements angesehen werden.
- Überwachung der Einhaltung von Prozessen und Prüfung der Produkt-Qualität durch die Qualitätssicherung.
- Einplanung von Ressourcen, die für die Umsetzung von qualitätssichernden Maßnahmen nötig sind.
- Die Schaffung eines organisatorischen Rahmens mit Qualitäts-Personal, Schulungs-Einheiten, Qualitätszirkel und einem Qualitätshandbuch.

3.3.1 Qualitätssichernde Maßnahmen

Maßnahmen zur Qualitätssicherung lassen sich unterteilen in konstruktive, analytische und organisatorische Maßnahmen.

Konstruktive Maßnahmen beinhalten den Einsatz bestimmter Methoden und Werkzeuge bei der Konstruktion, wie zum Beispiel den Einsatz von integrierten Entwicklungsumgebungen (IDEs) oder testgetriebener Entwicklung [Beck '00]. Sie umfassen damit alle Maßnahmen die im Vorfeld dafür sorgen, dass Fehler vermieden werden können.

Analytische Maßnahmen analysieren die (Teil-)Ergebnisse des Projektes. Übliche Methoden sind hier statische Analyseverfahren (vorrangig Reviews) und dynamische Analyse (vorrangig Testen). In Abschnitt 3.3.3.1 wird gesondert auf Reviews eingegangen, da sie eine starke Verwandtschaft zu den Gate-Reviews des Quality-Gate-Konzeptes aufweisen (vgl. auch Abbildung 2). Ein wesentlicher Bestandteil der analytischen Qualitätssicherung ist das Messen der Softwarequalität. Hinsichtlich der Planung, Kontrolle und Steuerung der Qualität kann ein identischer Regelkreis wie beim Projektmanagement gebildet werden (vgl. Abbildung 15).

Organisatorische Maßnahmen dienen dazu, einen ausreichenden Rahmen für die Qualitätssicherung zu schaffen. Dazu gehören die Etablierung von Schulungsmaßnahmen, die Etablierung und Pflege von Vorgehensmodellen und die Institutionalisierung des Qualitätsmanagement.

3.3.2 Qualitätsorganisation als Schattenhierarchie

Analog zur Projekthierarchie aus Abschnitt 3.1.1 kann die Qualitätsorganisation nach Glinz [Glinz '05] ähnlich hierarchisch organisiert werden. Es existiert in jeder Abteilung eines Unternehmensbereiches ein Qualitätsbeauftragter. Die Aufgabe des Qualitätsbeauftragten ist die Sicherung der Qualität für alle Projekte der entsprechenden Abteilung. Der Qualitätsmanager befindet sich auf der gleichen Ebene wie der Projektmanager. Seine Aufgabe liegt darin, das Qualitätsmanagement langfristig für die Pro-

jekte eines Unternehmensbereichs zu institutionalisieren. Auf höchster Ebene besitzt die Geschäftsleitung eine Stabsstelle Qualitätsleiter.

Die Qualitätsorganisation verfügt über ein eigenes Berichtswesen (mit Möglichkeiten der Eskalation) und über eigene Weisungsbefugnisse. Mit dieser Unabhängigkeit kann zwischen der Qualitätsorganisation einerseits und der Projektorganisation andererseits, das richtige Maß an Qualitätssicherung ausgehandelt werden. Dabei stehen beide Organisationen hinsichtlich ihrer Interessen in einem konträren Verhältnis. Die Projektorganisation möchte das Projekt möglichst schnell unter Verwendung möglichst weniger Ressourcen erfolgreich zum Ziel führen. Die Qualitätsorganisation möchte garantieren, dass die Software maximale Qualität aufweist. Beide Ziele verhalten sich konträr zueinander (vgl. Abbildung 8 und Abbildung 9), daher ist eine Aushandlung des richtigen Maßes notwendig.

3.3.3 Zum Begriff des Messens

Das Messen dient dem Verstehen, Steuern und Verbessern [DeMarco '82]. Zentraler Begriff des Messens ist das Maß. Zunächst sollen die Begriffe Maß, Messen, Messwert, Kennzahl, Kennzahlwert und Kriterium definiert werden. Ein Maß ist wie folgt definiert:

Definition: Ein *Maß* (sehr häufig auch *Metrik*) ist eine Abbildungsvorschrift, die Merkmalen von (Mess-)Objekten der realen Welt Zahlen oder Symbole zuordnet.

Genauer ausgedrückt ist ein Maß eine Funktion

$$m : O \rightarrow S,$$

wobei O eine Objektmenge und S eine beliebige Menge ist. S wird als Skala bezeichnet. Eine Skala kann hinsichtlich ihrer Mächtigkeit unterschieden werden (vgl. Tabelle 7). Die Mächtigkeit der Skala eines Maßes beeinflusst erheblich die Art der Aussagen, die sich auf Grundlage der Messwerte treffen lassen.

Das Maß m ordnet Merkmalen eines Messobjektes $o \in O$ einen Wert der Skala S zu. Im Sinne der Modelltheorie sind dies die für das Maß m (und damit für einen Nutzer) relevanten Merkmale des Objektes o .

Eine *Kennzahl* ist ein spezielles Maß, dessen Definitionsbereich entweder Projekte, Prozesse, Personen oder Produkte sind.

Definition 7: Maß und Kennzahl

Weiterhin lassen sich nun der Vorgang des Messens und der Messwert wie folgt definieren.

Definition: *Messen* ist der Vorgang, der ein Maß m aufgrund einer Messvorschrift v auf ein Objekt o der realen Welt anwendet. Die Messvorschrift v gibt an, wie, wann und durch wen das Maß auf das Objekt anzuwenden ist. Das Ergebnis der Messung $m(o)$ aufgrund einer Messvorschrift v wird als *Messwert*, *Istwert* oder im Falle einer Kennzahl als *Kennzahlwert* bezeichnet.

Definition 8: Messen und Messwert

Ein gutes Maß sollte nach Liggesmeyer [Liggesmeyer '02] die folgenden Anforderungen erfüllen:

- **Einfachheit:** Die Messwerte lassen sich leicht berechnen und interpretieren.
- **Eignung:** Das Maß ist ausreichend korreliert mit den zu messenden Merkmalen.
- **Stabilität:** Das Maß ist stabil gegenüber geringfügigen Manipulationen des Messobjektes.
- **Rechtzeitigkeit:** Die Messwerte sind schon vor dem Zeitpunkt der Nutzung verfügbar.
- **Analysierbarkeit:** Inwieweit lassen sich Messwerte miteinander in Relation setzen (vergleiche Tabelle 7) und inwieweit sind die Messwerte statistisch verwertbar?
- **Reproduzierbarkeit:** Das Maß ist unabhängig von der Messdurchführung, das heißt insbesondere unabhängig von der subjektiven Wahrnehmung.

Ein Kriterium setzt Sollwerte und einen Messwert in Beziehung. Ein Kriterium stellt eine messbare Forderung an ein Ergebnis des Entwicklungsprozesses dar. Auf diese Weise kann geprüft werden, ob ein bestimmtes Kriterium erfüllt ist oder nicht. Ein Kriterium ist zumeist umgangssprachlich gegeben, es ist jedoch formal wie folgt definiert:

Definition: Ein Kriterium ist ein vierstelliges Tupel $(m, \bar{o}, \bar{s}, p_m)$, bestehend aus einem Maß $m : O \rightarrow S$, einem konkreten Objekt \bar{o} , auf das das Maß angewendet werden soll, einer Menge konkreter Sollwerte $\bar{s} \in P(S)$ mit $|\bar{s}| \geq 1$ und einem Prädikat p_m . Das Prädikat vergleicht den Messwert mit dem Sollwert \bar{s} und liefert entweder wahr oder falsch zurück, je nachdem wie groß die Abweichung zwischen Soll- und Istwert ist.

Das Prädikat ist wie folgt definiert

$$p_m : O \times P(S) \rightarrow B$$

$$p_m : (o, s) \mapsto p_m(o, s)$$

und hat die folgende Eigenschaft

$$\text{Kriterium ist erfüllt} \Leftrightarrow p_m(\bar{o}, \bar{s}) = \text{true}.$$

$p_m(\bar{o}, \bar{s})$ heißt Bewertung des Kriteriums.

Anmerkung:

Im weiteren Verlauf wird \bar{s} stets als Sollwert bezeichnet, egal ob es sich dabei um eine Menge mit einem Element oder mehreren Elementen handelt.

Definition 9: Kriterium

Im Folgenden wird ein Beispiel für ein Kriterium gegeben.

Beispiel: Es sei das folgende umgangssprachliche Kriterium gegeben:

„Alle Methoden des Programms „MyWebSchedule“ dürfen höchstens 20 Codezeilen haben, wenn Kommentar- und Leerzeilen nicht berücksichtigt werden.“

Die einzelnen Tupteleinträge des Kriteriums stellen sich dann folgendermaßen dar:

- m ist ein Maß, das von der Menge der Programme nach N^0 abbildet und das Maximum der Codezeilen aller Methoden ohne Kommentar- und Leerzeilen eines Programms berechnet.
- Das Objekt \bar{o} ist das Programm „MyWebSchedule“.
- Der Sollwert \bar{s} beträgt 20 Codezeilen.
- Das Prädikat ist folgendermaßen definiert:

$$p_m : (o, s) \mapsto \begin{cases} \text{true}, & \text{falls } m(o) \leq s \\ \text{false}, & \text{falls } m(o) > s. \end{cases}$$

Beispiel 1: Beispiel für ein Kriterium

Im weiteren Verlauf der Arbeit werden Quality-Gate-Kriterien nicht durch die eingeführte formale Tupeldarstellung beschrieben, sondern in ihrer leichter lesbaren umgangssprachlichen Darstellung belassen. Durch die obige Definition liegen jedoch die verpflichtenden Elemente eines Kriteriums fest. Unabhängig von der Art der Darstellung ist, muss ein Kriterium stets immer alle Elemente aufweisen.

Eine Skala als Wertebereich eines Maßes kann anhand ihrer Mächtigkeit unterschieden werden. Sie ist daher ein wichtiges Indiz dafür, welche Schlussfolgerungen aus den Messwerten gezogen und inwieweit verschiedene Werte der Skala miteinander verglichen werden können. Es können insgesamt fünf verschiedene Skalentypen nach ihrer Mächtigkeit und den möglichen Interpretationen unterschieden werden. Tabelle 7 zeigt die verschiedenen möglichen Skalentypen und ihre Eigenschaften.

Art der Merkmale	Skala	Eigenschaften	Erlaubte math. Operationen (zwischen Werten der Skala)	Beispiel
Qualitativ	Nominalskala	Ungeordnete Menge, Vergleiche nur zwischen einzelnen Messwerten auf gleich oder ungleich möglich.	=, ≠	Gewählte Programmiersprache
	Ordinalskala	Geordnete Menge, Vergleiche auf kleiner oder größer möglich, keine Aussage über Differenz zweier Messwerte.	=, ≠, <, >	Risikobewertung auf Skala gering < möglich < hoch < sehr hoch
Quantitativ	Intervallskala	Differenzen zwischen Messwerten können berechnet werden und haben Bedeutung.	=, ≠, <, >, +, -	Zyklomatische Komplexität nach McCabe [McCabe '76]
	Rationalskala	Messwerte können zueinander in Relation gesetzt werden, Teilen und multiplizieren sind möglich.	=, ≠, <, >, +, -, *, /	Projektkosten
	Absolutskala	Das Maß steht für sich selbst und kann nicht anders ausgedrückt werden. Anwendung: Häufigkeit, Zählen, Wahrscheinlichkeiten	=, ≠, <, >, +, -, *, /	Lines of Code

Tabelle 7: Skalentypen von Maßen

Weiterhin gilt für die Skalentypen:

Sei $\Sigma_x = \{S \mid S \text{ ist eine Skala vom Typ } x\}$, dann gelten die folgenden echten Teilmengen-Inklusionen:

$$\Sigma_{\text{Nominalskala}} \supset \Sigma_{\text{Ordinalskala}} \supset \Sigma_{\text{Intervallskala}} \supset \Sigma_{\text{Rationalskala}} \supset \Sigma_{\text{Absolutskala}}$$

Das heißt zum Beispiel, dass jede Absolutskala gleichzeitig eine Nominalskala ist und damit ebenfalls deren Eigenschaften hat.

3.3.3.1 Einordnung von Maßen

Im Bereich der Software-Entwicklung werden Messungen am Prozess, Produkt und Projekt durchgeführt. Maße lassen sich mindestens einer dieser drei Kategorien zuordnen [Liggesmeyer '02]. Eine derartige Einordnung liefert Aufschluss darüber, *welcher* Art die Objekte sind, die mit dem Maß vermessen werden können.

Produktmaße liefern Informationen über die Merkmale eines Produktes. Abweichungen des Ist-Wertes vom Sollwert deuten auf Probleme im Produkt hin. Prozessmaße beziehen sich auf Merkmale des Prozesses. Durch den Vergleich von Soll und Ist kann zum Beispiel geprüft werden, ob einzelne Prozessschritte ordnungsgemäß durchgeführt wurden oder nicht. Es ist möglich aus jedem Produktmaß, ein Prozessmaß zu erhalten, indem man den zeitlichen Verlauf der Messwerte des Produktmaßes betrachtet. Projektmaße spiegeln Merkmale des Projektes wider und dienen vor allem der Steuerung des Projektes.

Wie bereits angedeutet, ist es möglich, dass ein Maß mehreren der drei Kategorien zugeordnet werden kann. Abbildung 19 zeigt eine Einordnung der Maße hinsichtlich der drei Kategorien.

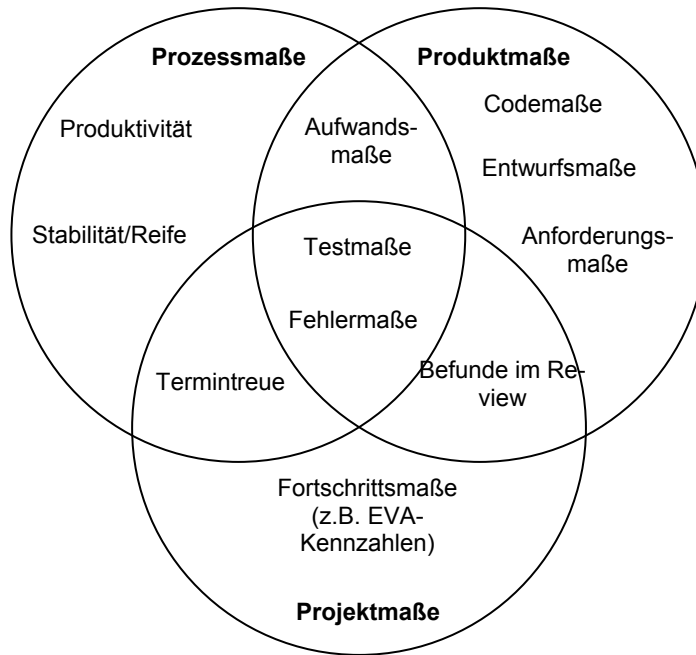


Abbildung 19: Einordnung von Maßen (nach [Liggesmeyer '02])

Ebenso ist es sinnvoll, Maße nach der Verwendung in den Phasen des Software-Entwicklungsprozesses einzuordnen. Auf diese Weise geht hervor, *wann* der Einsatz eines Maßes sinnvoll ist. Abbildung 20 zeigt eine grobe Einordnung der Maße nach Phasen.

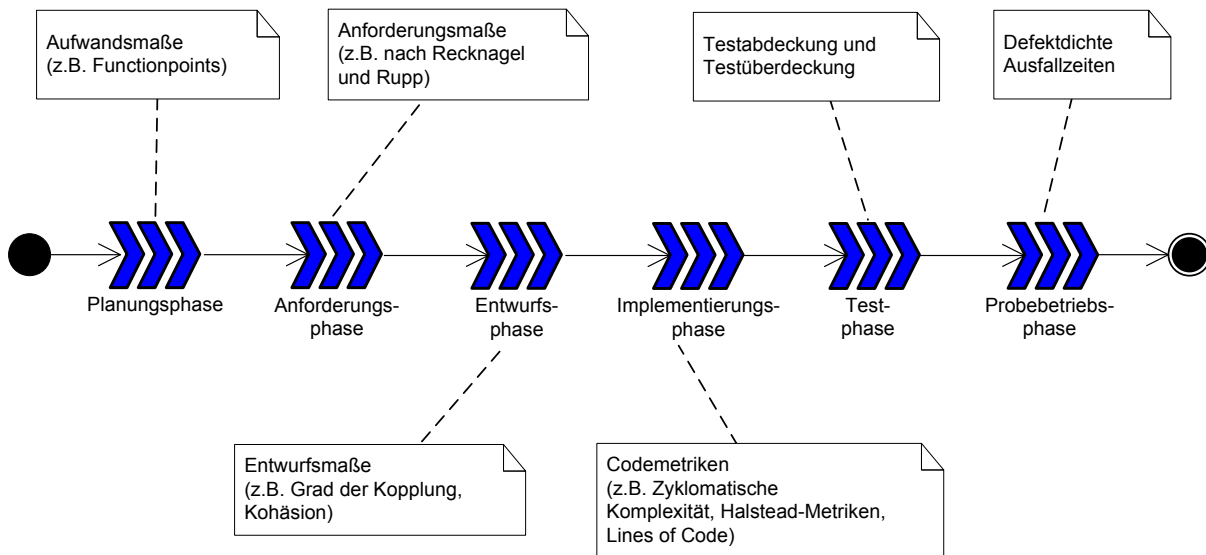


Abbildung 20: Maße eingeordnet nach Phasen

3.3.3.2 Zielgerichtetes Messen

Eine Ableitung von Maßen aus Zielen ist notwendig, um nützliche Maße identifizieren zu können. Prozessbeteiligte neigen häufig dazu, das zu messen, was leicht zu messen ist, aber eigentlich ungeeignet für die Bewertung von vorhandenen Zielen ist. Ungeeignete Maße führen zu einer ungeeigneten Darstellung des zu untersuchen Produktes, Prozesses oder Projektes und damit möglicherweise zu Fehlentscheidungen. Darüber hinaus werden für das Messen erhebliche Ressourcen benötigt. Insbesondere für die Feststellung der Qualität müssen Maße projektspezifisch bestimmt werden, da Qualitätsaspekte äußerst allgemein und oberflächlich definiert sind. Die relevanten Qualitätsaspekte müssen daher für jedes Projekt neu interpretiert werden.

Ein Ansatz, mit dem von oben nach unten aus Zielen passende Maße abgeleitet werden können, ist die *Goal-Question-Metric-Methode* (GQM) nach Basili et al. [Basili et al. '94b]. Ursprünglich auf die

Verbesserung von Entwicklungsprozessen ausgerichtet, lässt sich die GQM-Methode aber auch auf die Definition von Maßen für Produkte und Projekte ausdehnen und dort praktikabel anwenden. Ziel des Ansatzes ist es nicht, möglichst viele Maße abzuleiten, sondern die für den Kontext passenden Maße. Dabei muss nicht zwangsläufig auf bekannte Maße zurückgegriffen werden. Vielmehr können eigene Maße definiert werden, die häufig mehr Aussagekraft besitzen.

Es lassen sich nach [Schneider '07b] drei Messzwecke unterscheiden: verstehen (analysieren), kontrollieren und verbessern bestimmter Qualitätsaspekte. Verstehen bedeutet, einen Qualitätsaspekt messbar zu machen, um hieraus Schlussfolgerungen ziehen zu können. Kontrollieren hat das Ziel, nach der Einführung einer Änderung die Auswirkungen auf einen bestimmten Qualitätsaspekt messbar zu machen. Verbessern zielt auf die Verbesserung eines bestimmten Qualitätsaspekts. Um diese Verbesserung messbar zu machen, ist die GQM-Methode zweimal anzuwenden. Ein erstes Mal, um vorhandene Unzulänglichkeiten zu *verstehen* und entsprechende Maßnahmen abzuleiten; ein zweites Mal, um die Auswirkungen der Maßnahmen zu *kontrollieren*.

Ziele befinden sich auf höchster Ebene und sind entsprechend abstrakt. Durch die Verfeinerung dieser Ziele über Messziele, Fragestellungen und optionalen Indikatoren ergeben sich Maße. Basierend auf den abgeleiteten Maßen wird ein Messplan erstellt, der die notwendigen Messvorschriften für die Messung vorgibt. Auf Grundlage dieses Messplans werden Daten erhoben. Diese Daten werden aufbereitet, aggregiert und analysiert, um die Fragestellungen beantworten zu können. Über die Beantwortung der Fragen kann letztlich festgestellt werden, ob Messziele und damit ein Verbesserungsziel erreicht wurde oder nicht. Dabei erfolgt die Definition von Metriken von oben nach unten (top-down), die Interpretation der Messdaten von unten nach oben (bottom-up). Abbildung 21 illustriert diesen Ablauf.

Der Einsatz der GQM-Methode gestaltet sich mitunter aufwändig und mühsam, da die Ergebnisse nur in Workshops erarbeitet werden können. Eine Unterstützung und Zustimmung des Managements ist daher notwendig [Solingen et al. '99]. Bei vorhandener Zustimmung führt die Methode jedoch meistens zum Ziel. Zu beachten ist, dass die Erreichung von Verbesserungszielen teuer ist, da hierzu zweifach gemäß der GQM-Methode zu verfahren ist. Ein erstes Mal, um mögliche Problem zu identifizieren, die vor der Einleitung von Maßnahmen bestehen; ein zweites Mal, um die Auswirkungen der Maßnahmen zu identifizieren.

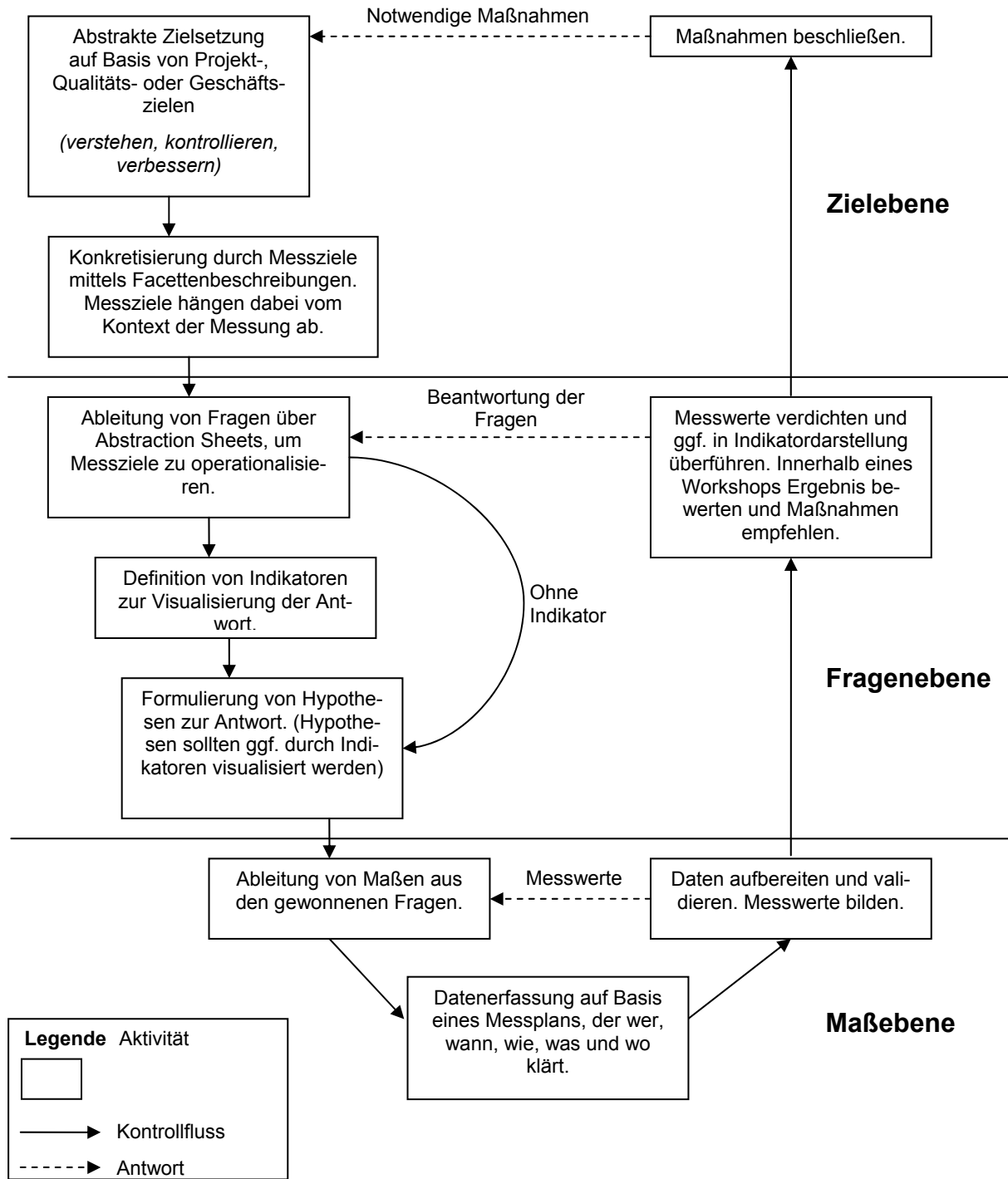


Abbildung 21: Die GQM-Methode (nach [Assmann et al. '02])

Eine weitere Möglichkeit Maße zielgerichtet auszuwählen, ist die Herleitung über Qualitätsmodelle. Mit ihnen lassen sich ebenfalls Messvorschriften gewinnen. Die Herleitung erfolgt schrittweise von oben nach unten ähnlich dem Vorgehen bei GQM-Methode. Der Unterschied zur GQM-Methode besteht im Wesentlichen darin, dass Qualitätsmodelle vor allem für die Herleitung von Produktmaßen genutzt werden können. Die folgende Abbildung zeigt den schematischen Aufbau eines Qualitätsmodells (links) und das hierzu notwendige Vorgehen (rechts).

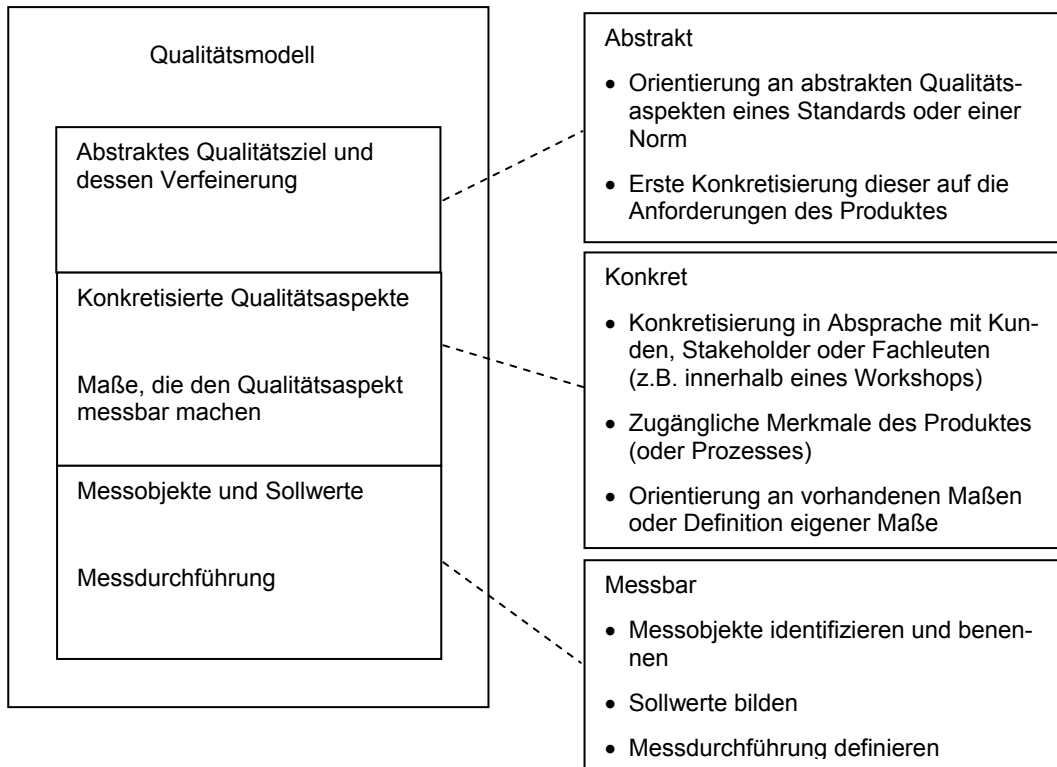


Abbildung 22: Schematische Darstellung eines Qualitätsmodells (nach [Schneider '07b])

Abbildung 22 zeigt beispielhaft ein erarbeitetes Qualitätsmodell für den abstrakten Qualitätsaspekt Zuverlässigkeit für die Software eines Navigationsgerätes.

Qualitätsziel	Zuverlässigkeit	
	Genauigkeit	Vollständigkeit
Qualitätsaspekt	Wie genau ist die Berechnung von Routen?	Sind alle Verbindungen gespeichert?
Maß	Durchschnittliche Abweichung der Routenlänge von der günstigsten Strecke in Prozent	Anzahl der hinterlegten Orte und Wege
Prüfgegenstände und Sollwerte	Routenlängen berechnen. Alle Abweichungen < 1%	Kartenmaterial mit Geodaten. Alle Orte über 100 Einwohner verzeichnet. Alle Straßen mit Bezeichnung hinterlegt. Verbindung jeweils berechenbar.
Prüfungsdurchführung	Routenalgorithmus mit Bibliothek mit automatisierten Tests separat von Oberfläche testen. Tausend beliebige Routenlängen mit den Routenlängen der Realität vergleichen.	Tausend beliebige Verbindungen prüfen. Mindestens 100 Orte mit 100-200 Einwohnern. Insbesondere auch kleinere Straßen testen.

Abbildung 23: Qualitätsmodell für den Qualitätsaspekt Zuverlässigkeit

Ausgangspunkt der GQM-Methode und für Qualitätsmodelle sind Standards und Normen wie *ISO/IEC 9126* oder *DIN 66272*. Sie liefern mögliche abstrakte Qualitätsaspekte. Dadurch, dass ein

Kunde bestimmte Qualitätsaspekte implizit oder explizit fordert (vgl. Qualitätsbegriff, Abschnitt 3.3), werden diese Qualitätsaspekte zu Qualitätszielen. Qualitätsaspekte lassen sich weiterhin aus dem Qualitätsbaum von Boehm et al. [Boehm et al. '78] ableiten (vgl. Abbildung 24). Mit Hilfe des Qualitätsbaums kann die oberste Ebene (also das Qualitätsziel) erarbeitet werden. Dabei sind vor allem die Ebenen *Intermediate-level Requirements* und *Low-level Requirements* von Interesse.

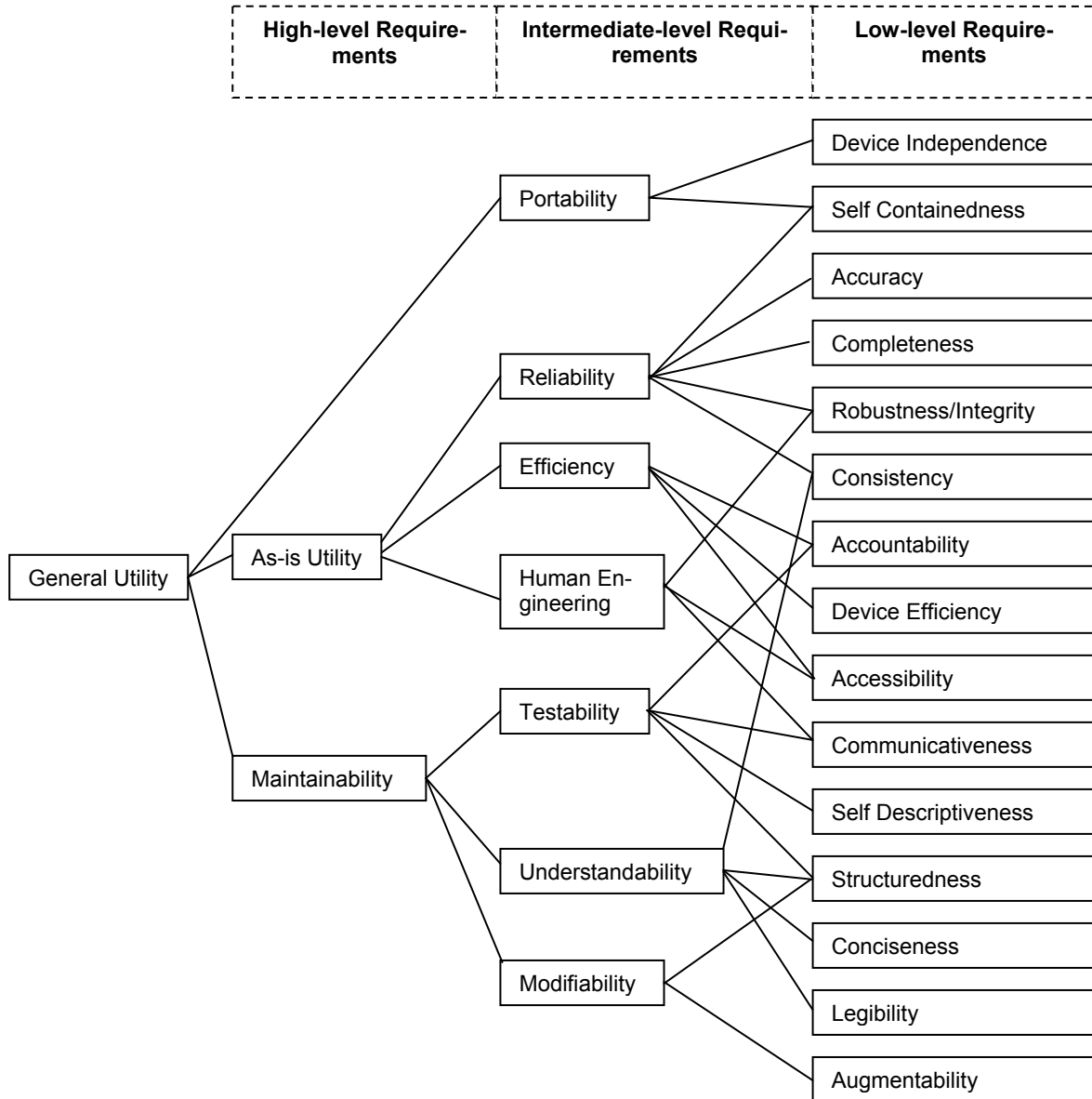


Abbildung 24: Qualitätsbaum nach Boehm [Boehm et al. '78]

Auch der IEEE-Standard *IEEE Std 610.12-1990* gibt verschiedene Qualitätsaspekte vor und definiert sie wie folgt (Tabelle 8).

Qualitätsaspekt	Beschreibung nach Standard IEEE Std 610.12-1990
Effizienz	Der Grad, zu welchem ein System die erwarteten Funktionen mit einem minimalen Ressourcenverbrauch erfüllt.
Flexibilität	Der Grad der Leichtigkeit, mit dem ein System oder eine Komponente verändert werden kann, um in einer anderen Anwendung oder Umgebung genutzt werden zu können, für die sie nicht ursprünglich entworfen wurden.
Integrität (Security)	Der Grad, zu welchem ein System oder eine Komponente nicht autorisierte Zugriffe oder Veränderungen auf bzw. von Daten oder Computerprogrammen verhindert.
Korrektheit	Der Grad, zu welchem die Software die Anforderungen der Anwender erfüllt, gleich ob sie spezifiziert wurden oder nicht.
Portabilität	Der Grad der Leichtigkeit, zu welchem ein System oder eine Komponente in eine andere Hard- oder Softwareumgebung transferiert werden kann.
Testbarkeit	- Der Grad, zu welchem ein System oder eine Komponente die Erstellung von Testfällen und die Testdurchführung ermöglicht. oder auch - Der Grad, zu welchem Anforderungen, die Erstellung von Testfällen und die Testdurchführung ermöglichen.
Wartbarkeit	Der Grad der Leichtigkeit, zu welchem ein Softwaresystem oder eine Komponente die Korrektur von Fehlern oder die Verbesserung oder Anpassung bestimmter Aspekte unterstützt
Wiederverwendbarkeit	Der Grad, zu welchem ein Softwaremodul oder ein anderes Prozessartefakt in einer anderen Software genutzt werden können.
Verwendbarkeit	Der Grad der Leichtigkeit, zu welchem ein Benutzer lernen kann, ein System oder eine Komponente zu bedienen, Eingaben zu tätigen und die Ausgaben zu verstehen.
Zuverlässigkeit (Safety)	Der Grad, zu welchem ein System oder eine Komponente seine Funktionen unter festgelegten Bedingungen erfüllen kann.

Tabelle 8: Typische Qualitätsaspekte für Software

3.3.4 Technische Reviews

Technische Reviews gehören zu den statischen Methoden innerhalb der analytischen Qualitätssicherung, da durch sie die Software nicht zur Ausführung gebracht wird. Daher eignen sie sich insbesondere auch schon für die frühen Phasen eines Entwicklungsprozesses, wenn dynamische Verfahren noch nicht möglich sind. Grob gesprochen werden innerhalb eines Reviews Dokumente von Gutachter gelesen und dabei gegen – mehr oder weniger verfestigte – technische Kriterien geprüft. Reviews sind gängige Praxis als Mittel der Qualitätssicherung bei der Entwicklung von Software, werden jedoch zumeist nur unsystematisch ausgeführt [Ciolkowski et al. '03].

Es lassen sich verschiedene Review-Varianten nach ihrer Intensität und Formalität unterscheiden. Tabelle 9 gibt eine Übersicht über die verschiedenen Review-Varianten.

Review-Variante	Beschreibung	Aufwand
Schreibtischttest (englisch Peer Review)	Informeller Prozess: einzelner (zumeist gleichgestellter) Gutachter erhält Dokument direkt vom Autor, liest es kritisch (ad hoc) und liefert es mit Anmerkungen zurück.	Niedrig
Walkthrough	Autor erläutert sein Dokument mehreren Gutachtern innerhalb einer gemeinsamen Sitzung. Gutachter suchen nach Mängeln. Mängel werden protokolliert.	Mittel
Formale Inspektion	Stark formalisierter Prüf-Prozess bestehend aus verschiedenen Aktivitäten (vgl. Abbildung 25).	Hoch
Formale Inspektion mit Diskussion über Verbesserungsvorschläge (sog. 3. Stunde)	Wie Inspektion, jedoch mit Diskussion über Verbesserungsvorschläge (diese gehört im eigentlichen Verständnis des Reviews bei den meisten Autoren nicht zum Review).	sehr hoch

Tabelle 9: Übersicht der verschiedenen Review-Varianten

Wie bereits angedeutet, lässt sich die formale Inspektion in verschiedene Aktivitäten unterteilen. Ebenso sind verschiedene Rollen und Dokumente beteiligt.

An den Aktivitäten sind verschiedene Rollen beteiligt: ein Moderator, ein Autor, mindestens ein (in der Regel jedoch mehrere) Gutachter und ein Schriftführer (auch Protokollant oder Aktuar). Der Moderator ist für die Koordinierung und Planung der Inspektion verantwortlich. Der Autor erstellt das zu prüfende Dokument (den sog. Prüfling). Die Gutachter inspizieren den Prüfling und verfügen idealerweise über notwendiges Fachwissen. Bei der Begutachtung können die Gutachter durch eine Checkliste oder Lesetechnik angeleitet werden, die die Suche nach möglichen Defekten erleichtert. Außerdem stehen ihnen Referenzdokumente zur Verfügung gegen die der Prüfling geprüft werden kann. Identifizierte Befunde notiert jeder Gutachter in einer Einzel-Befundliste. Der Schriftführer erstellt ein Protokoll (die sog. Gesamt-Befundliste) mit den wichtigen und unstrittigen Befunden.

Der generische Inspektionsprozess lässt sich oberflächlich unterteilen in Organisations-, Prüfungs- und Abschlussaktivität [Macdonald '98]. Ihre Abfolge erfolgt sequenziell. Die drei Aktivitäten unterteilen sich wiederum in verschiedene Unteraktivitäten. Abbildung 25 zeigt einen Überblick der Aktivitäten. Eine Variante der formalen Inspektion lässt es zu, die Sitzung auszulassen. Die anderen Aktivitäten sind jedoch verpflichtend, können jedoch in ihrer Intensität variieren.

Die Organisation umfasst eine formale Eingangsprüfung, die eigentliche Planung sowie eine Einführung. Die formale Prüfung stellt sicher, dass die Eintrittskriterien für den Start des Inspektionsprozesses erfüllt sind. Dazu gehört zum Beispiel die Prüfung, ob der Prüfling in geeigneter Form vorliegt, so dass er ohne Probleme von den Gutachter begutachtet werden kann vorliegt. Die Planung beschäftigt sich mit der Zuweisung von Personen zu den Rollen und der Vorbereitung und Verteilung aller im Review benötigten Dokumente. Darüber hinaus wird der zeitliche Rahmen festgelegt. Das Ziel der Einführung (häufig auch Kickoff genannt) ist es, die Gutachter mit dem Inspektionsprozess und dem Prüfling vertraut zu machen.

Die Prüfung umfasst die individuelle Vorbereitung der Gutachter und eine Sitzung. In der individuellen Vorbereitung bereiten sich die Gutachter einzeln vor, indem sie den Prüfling inspizieren und Befunde sammeln. Hierbei können verschiedene Lesetechniken zum Einsatz kommen, die sich grob klassifizieren lassen nach szenariobasierten und perspektivischen Lesetechniken [Basili et al. '96]. In der Praxis werden diese, vergleichsweise komplizierten Techniken selten eingesetzt. Stattdessen kommen unsystematische Lesetechniken oder Lesetechniken unter Verwendung von Checklisten in 90% aller Fälle zum Einsatz [Ciolkowski et al. '03].

Innerhalb der Sitzung wird über die so identifizierten Befunde diskutiert, ihre Kritikalität bewertet und eine Empfehlung ausgesprochen. Die Auffassungen über die Effektivität und Effizienz der individuellen Vorbereitung und der Sitzung gehen auseinander. Fagan [Fagan '76] und Gilb et al. [Gilb et al. '93]

gehen davon aus, dass in der Sitzung selbst noch entscheidende Fehler gefunden werden können und daher die Sitzung unerlässlich ist. Humphrey [Humphrey '89] geht hingegen davon aus, dass die meisten Befunde bereits während der Vorbereitung gefunden werden. Russel [Russel '91], Ballmann et. al [Ballman et al. '94] schätzen, dass die Durchführung einer Sitzung das Review um ungefähr 20% verteuert. Votta [Votta '93] geht weiterhin davon aus, dass die Durchführung einer Sitzung, die formale Inspektion um ungefähr ein Drittel verlängert, jedoch nur durchschnittlich 5% mehr Defekte identifiziert werden können. Befürworter der Sitzung argumentieren hingegen, dass ein erheblicher Teil der Befunde erst in der Sitzung selbst gefunden wird [Porter et al. '95]. Sie beziffern den Zugewinn in Einzelfällen auf bis zu 80%. Im Mittel beträgt er jedoch ein Drittel.

Der Abschluss umfasst die letzten Aktivitäten des Reviews. Dazu gehören die Überarbeitung (englisch Rework), die Nachverfolgung (englisch Follow-Up), eine formale Ausgangsprüfung und eine Leistungsprüfung. Die Überarbeitung sieht eine Korrektur des Prüflings durch den Autor vor, die in der Nachverfolgung durch den Moderator geprüft wird. Der Moderator entscheidet auf Basis der Korrektur über die Notwendigkeit einer weiteren Inspektion. Durch die formale Ausgangsprüfung wird geprüft, ob die Ausgangskriterien des Reviews eingehalten wurden. Als formales Ausgangskriterium kann beispielsweise verlangt werden, dass der Prüfling eine definierte Anzahl Befunde unterschreiten muss. Letztlich wird die Leistung der Inspektion gemessen, um Aussagen über Effektivität und Effizienz der Inspektion zu erhalten. Dazu gehört zum Beispiel die Frage nach der Anzahl der entdeckten kritischen Befunde in Relation zu den investierten Ressourcen.

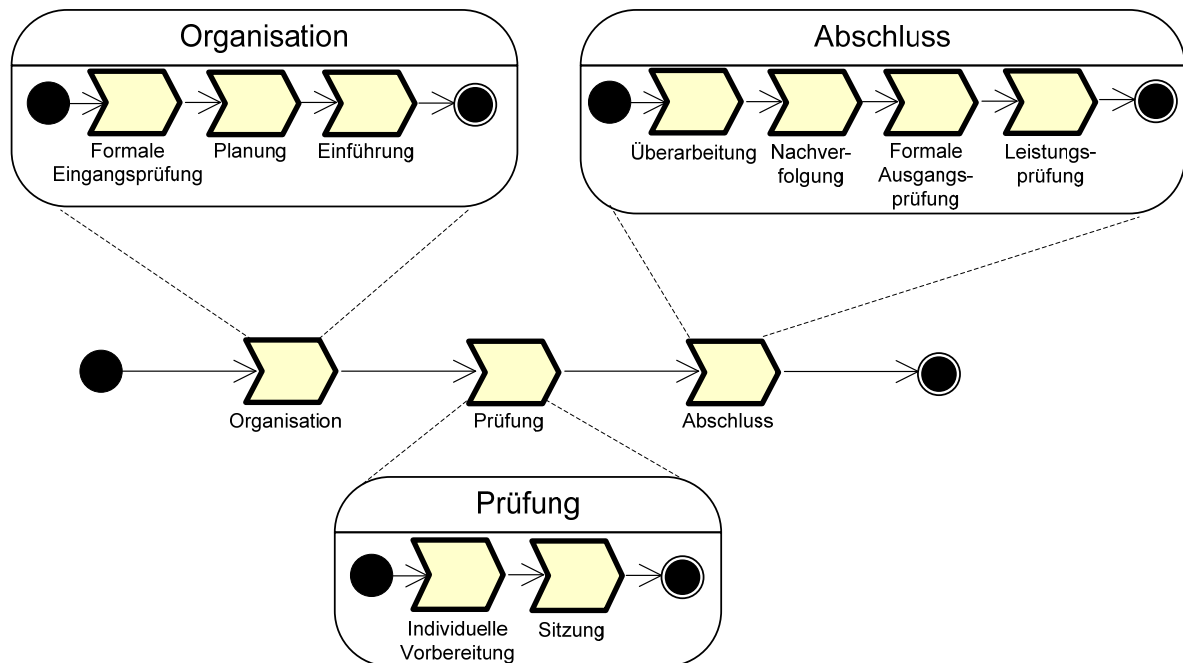


Abbildung 25: Generische formale Inspektion (nach [Macdonald '98])

Ein strukturiertes Walkthrough nach Yourdon [Yourdon '89] ist eine formale Inspektion ohne individuelle Vorbereitung der Gutachter. Ein (aufbereiteter) Prüfling wird hingegen vom Autor in der Sitzung vorgestellt und von den Gutachtern auch dort begutachtet. Die Gutachter müssen sich daher nicht speziell vorbereiten. Eine Einführung entfällt daher ebenfalls.

4 Untersuchung von Quality-Gate-Referenzprozessen

Im Rahmen dieses Kapitels wird das Quality-Gate-Konzept näher untersucht. Dazu werden sowohl Ansätze aus der Literatur, als auch in verschiedenen Software-Unternehmen praktizierte Ansätze näher betrachtet. Die wichtigsten so identifizierten Quality-Gate-Referenzprozesse werden näher untersucht. Im weiteren Verlauf dieses Kapitels werden Quality Gates ebenfalls mit etablierten Prozessreifegradmodellen in Beziehung gesetzt. Ziel dieses Kapitels ist eine abschließende Charakterisierung des Quality Gates sowie die Abgrenzung von den Konzepten des Meilensteins und des technischen Reviews. Darüber hinaus werden Problempunkte identifiziert, auf die das Framework eingehen muss.

4.1 Methodische Herangehensweise

Die Untersuchung beginnt mit den in der Literatur untersuchten Quality-Gate-Referenzprozessen. Dazu werden die Forschungsaktivitäten im Kontext des Software Engineering quantitativ und qualitativ betrachtet. Anschließend werden verschiedene in der Literatur vorhandene Konzepte einander gegenüber gestellt. Dabei handelt es sich vorrangig um Quality-Gate-Referenzprozesse, die in Entwicklungsprozessen in der Automobilindustrie und in Systementwicklungsprozessen Verwendung finden.

Die Untersuchung der vorhandenen Literatur zeigt, dass Quality-Gate-Referenzprozesse für Software-Entwicklungsprozesse noch kaum wissenschaftlich untersucht sind. Dies wird durch Wallin et al. [Wallin und Ekdahl et al. '02, Wallin und Larsson et al. '02] ebenfalls bestätigt. Zugleich ist festzustellen, dass Quality Gates dennoch in vielen Software-Unternehmen genutzt werden. Offenkundig klafft daher eine Lücke zwischen dem praktischen Einsatz einerseits und dem theoretischen Unterbau andererseits. Die Notwendigkeit zur näheren wissenschaftlichen Untersuchung wird ferner dadurch deutlich, dass noch eine Vielzahl von Problempunkten existiert.

Um Erkenntnisse über diese Quality-Gate-Referenzprozesse zu gewinnen, wurde als Teilleistung dieser Arbeit eine empirische Erhebung in Form einer Umfrage durchgeführt. Basierend auf den gewonnenen Erkenntnissen werden die wichtigsten (für die Praxis relevanten) Quality-Gate-Referenzprozesse betrachtet. Dies sind der Quality-Gate-Referenzprozess des V-Modells XT [VMXT '06], der Quality-Gate-Referenzprozess des Stage-Gate-Prozesses nach Cooper [Cooper '01] und der Quality-Gate-Referenzprozess nach Pfeifer [Pfeifer et al. '03].

Die Untersuchungsergebnisse liefern eine abschließende Charakterisierung, eine Abgrenzung des Quality-Gate-Konzeptes und eine Liste von Problempunkten als wesentliche Ergebnisse dieses Kapitels. Auf diese Weise kann ein Unternehmen bestimmen, ob es bereits Quality Gates einsetzt oder ein nur scheinbar ähnliches Konzept.

Ein weiterer Abschnitt dieses Kapitels widmet sich dem Verhältnis zwischen Quality-Gate-Referenzprozessen und den Prozessreifegradmodellen *CMMI* und *SPICE*. Diese Untersuchung ist relevant für Unternehmen, die bereits eine bestimmte Prozessreife erlangt haben oder planen, ihre Prozessreife zu erhöhen. Das Verhältnis ist hierbei von zweierlei Natur:

- Die Etablierung von Quality Gates hilft dabei, die Prozessreife zu erhöhen.
- Ein reifer Prozess erleichtert die Etablierung von Quality Gates.

Abbildung 26 fasst das Vorgehen der Untersuchung zusammen.

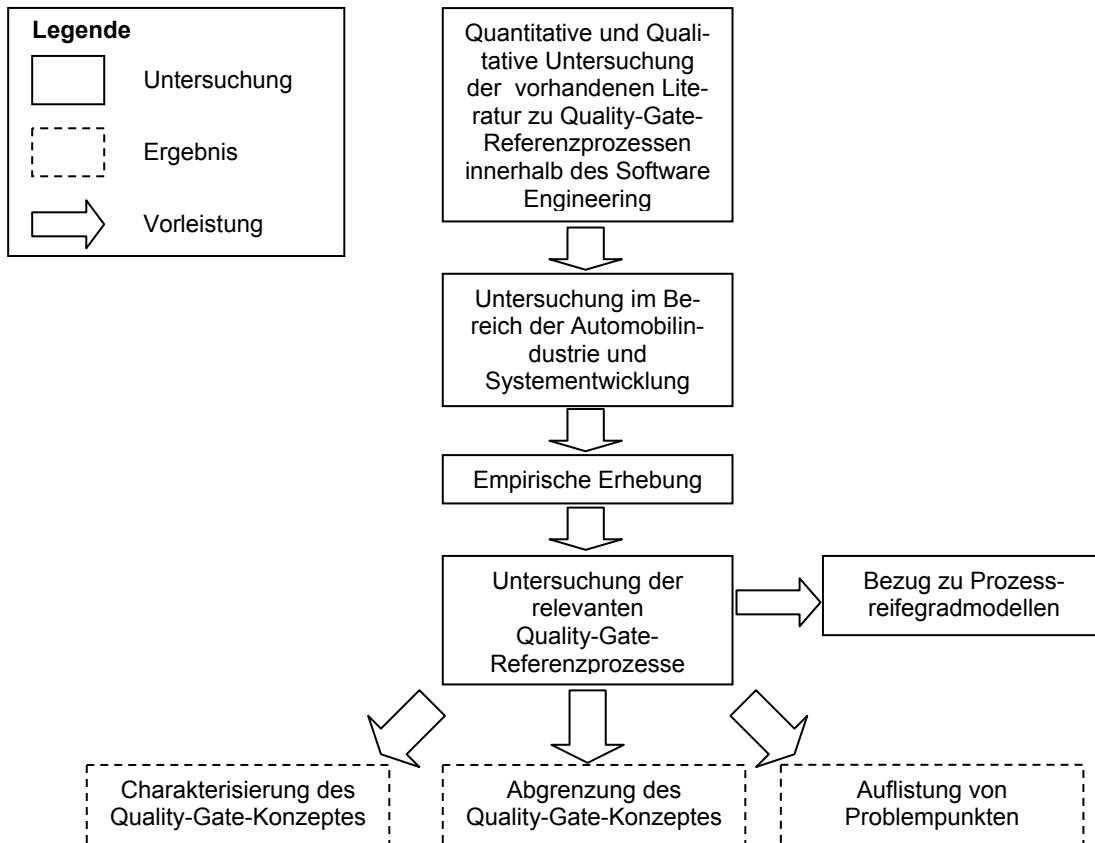


Abbildung 26: Vorgehen der Untersuchung vorhandener Quality-Gate-Referenzprozesse

4.2 Das Quality-Gate-Konzept in der Literatur

Dieser Abschnitt beschäftigt sich mit dem Quality-Gate-Konzept in der Literatur. Dazu gehört eine quantitative und qualitative Betrachtung des Konzeptes als Forschungsgegenstand im Software-Engineering, sowie die Betrachtung des Konzeptes in weiteren Domänen.

4.2.1 Das Quality-Gate-Konzept als Forschungsgegenstand im Software Engineering

Eine Untersuchung renommierter Fachzeitschriften und Konferenzbände, die sich mit dem Thema Software, Software Engineering oder Software-Qualität im Speziellen beschäftigen, zeigt, dass das Quality-Gate-Konzept nur sehr wenig untersucht wurde.

Als Teilleistung dieser Arbeit wurden verschiedene renommierte Fachzeitschriften und Konferenzbände ab dem Jahr 2000 für eine Untersuchung herangezogen, um hieraus ein Mengengerüst zu erstellen. Dabei wurden Titel und Zusammenfassungen der wissenschaftlichen Artikel jeweils auf Begriffe untersucht, die auf einen Hauptmeilenstein oder ein Management-Review hindeuteten (vgl. Abbildung 1 und Abbildung 2). Enthielt ein Artikel im Titel oder der Zusammenfassung einen dieser Begriffe, so wurde er näher auf seine Relevanz hin untersucht. Tabelle 10 fasst die Ergebnisse der Untersuchung zusammen.

Typ	Quelle	Jahrgänge	Relevante Artikel
Fachzeitschrift	IEEE Transactions on Software Engineering	1/2000 – 3/2007	–
	IEEE Software	1/2000 – 1/2007	[Karlström et al. '05] [Wallin und Ekdahl et al. '02]
	IEE Proceedings Software	1/2000 – 5/2006	–
	ACM Transactions on Software Engineering and Methodology	1/2000 – 1/2007	–
	ACM Ubiquity	1/2000 – 5/2007	–
	Communications of the ACM	1/2000 – 3/2007	–
	Springer Software Quality Journal	1/2000 – 2/2007	–
Konferenz oder Workshop	International Conference on Software Engineering (ICSE)	2000 – 2006	–
	Workshop on Software Quality (WoSQ)	2002 – 2006	–
	International Workshop on Software Quality Assurance (SOQUA)	2004 – 2006	–

Tabelle 10: Überblick über die vorhandene Literatur zu Quality Gates

Offenkundig lassen die Ergebnisse der Literaturrecherche darauf schließen, dass das Quality-Gate-Konzept im Software Engineering noch wenig untersucht wurde. Inhaltlich beschäftigt sich die vorhandene beiden Beiträge mit der Verwendung eines, auf den Stage-Gate-Prozess [Cooper '01] basierenden Quality-Gate-Referenzprozesses in verschiedenen Software-Entwicklungsprozessen [Karlström et al. '05, Wallin und Ekdahl et al. '02]. Dabei wird explizit auch auf die Nutzbarkeit bei iterativem (und auch agilem) Vorgehen eingegangen.

4.2.2 Das Quality-Gate-Konzept als Forschungsgegenstand in anderen Domänen

In diesem Abschnitt wird das Quality-Gate-Konzept auf Grundlage der vorhandenen Literatur näher untersucht und definiert. Dabei werden vorrangig Quality-Gate-Referenzprozesse betrachtet, die nicht nur auf Software-Entwicklungsprozesse im Speziellen anwendbar sind, sondern vielmehr auf die Entwicklungsprozesse anderer Domänen. Bei der Betrachtung der chronologischen Entwicklung zeigt sich, dass das Quality-Gate-Konzept seinen Ursprung in Entwicklungsprozessen in der Automobilindustrie hat und später Einzug in die Entwicklungsprozesse anderer Domänen (und damit auch in die Software-Entwicklung) gehalten hat. Dieser Abschnitt zeigt diese chronologische Entwicklung des Quality-Gate-Konzeptes auf. Dabei wird insbesondere auf das Quality-Gate-Konzept nach Hawlitzky [Hawlitzky '02] eingegangen, da es besonders universell und ausführlich beschrieben ist.

Fauth et al. [Fauth et al. '99] beschreiben ein Quality Gate als ergebnisorientiertes Konzept, das vor allem bei Entwicklungsprozessen (vor Anlauf der eigentlichen Serienproduktion) in der Automobilindustrie Anwendung findet. Quality Gates sind Beurteilungspunkte, an denen zwischen Kunden und Lieferanten vereinbarte Leistungen auf ihre Einhaltung und Vollständigkeit hin geprüft werden. Um den Erfüllungsgrad dieser Leistungen zu prüfen, sind eindeutige Abnahmekriterien mit Schwellwerten zu formulieren. Bei Abweichung von diesen Schwellwerten sind Maßnahmen, wie der Abbruch, die Fortsetzung oder korrigierende Maßnahmen möglich.

Oeltjenbruns [Oeltjenbruns '02] berichtet über Quality Gates in der Automobilindustrie, die den Entwicklungsprozess in Phasen unterteilen. In jeder Phase sind von Teilprojekten definierte Aufgaben zu erfüllen, die die Ausgangskriterien dieser Phasen widerspiegeln. Im Vorfeld eines Quality Gates wird

durch Vorprüfungen (sog. Previews) vorzeitig sichergestellt, dass die Ausgangskriterien eingehalten werden können. Letztlich werden zum Quality Gate die Ergebnisse rückblickend betrachtet und bei Abweichungen entsprechende Maßnahmen ergriffen.

Nach Schubert et al. [Schubert et al. '06] hat das Quality Gate seinen Ursprung in der Automobilindustrie. Mittlerweile wird es jedoch auch bei der Entwicklung von Software eingesetzt, insbesondere wenn Soft- und Hardware im Rahmen der Systementwicklung zu einem System zusammengefügt werden müssen. Allgemein betrachtet ist das Quality Gate ein ergebnisorientierter Checkpunkt innerhalb eines Projektes, an dem Qualitätskriterien erfüllt sein müssen, um zur nächsten Phase des Entwicklungsprozesses zu gelangen.

Für Scharer [Scharer '01] sind Quality Gates bereits in den Entwicklungsprozessen für Einzelteillieferanten, für Systemlieferanten, für technische Gebrauchsgüter und für den Maschinenbau hinterlegt. Das Quality Gate verschmilzt dabei das Konzept des Entscheidungspunktes mit Aspekten der Qualitätssicherung. Ein Quality-Gate-Referenzprozess kann hierbei so angepasst werden, dass eine Nutzung in verschiedenen Projektsituationen möglich ist. Quality Gates werden vor den Phasen angeordnet und spiegeln deren technische und betriebswirtschaftliche Eingangskriterien wider [Bernards '05]. Scharer erweitert innerhalb seiner Dissertation [Scharer '01] das Quality-Gate-Konzept um die Komponente des Risikomanagements, um so rechtzeitig wirtschaftliche und technologische Risiken aufdecken und entsprechende Maßnahmen einleiten zu können.

Wildemann [Wildemann '03] sieht ein Quality Gate als ergebnisorientierten Zeitpunkt, der durch produkt- oder prozessspezifische Inhalte und Leistungen definiert ist. Das Quality Gate hat dabei zurück- und vorausschauende Funktion und grenzt sich vom Meilenstein dadurch ab, dass Ergebnisse und das weitere Vorgehen dort formal freigegeben werden können. Überschreiten Istwerte vorher definierte Schwellwerte, so wird das Risiko des Projektes für die Entscheidungsträger transparent.

Nach Kamiske [Kamiske et al. '05] werden Quality Gates an kritischen Punkten innerhalb der Entwicklungsprozesse eines Unternehmens gesetzt. An jedem Quality Gate wird beurteilt, ob ein angestrebter Stand im Projekt erreicht wurde oder nicht. Beim Erreichen des Quality Gates wird vom Management über das „Öffnen“ des betreffenden Quality Gates entschieden. Die für die Entscheidung notwendigen qualitativen und quantitativen Kriterien sind idealerweise vor Start der Durchführungsphase festzulegen.

Hawlitzky [Hawlitzky '02] definiert ein Quality Gate als ergebnis- oder ereignisorientierten Messpunkt mit produkt- und prozessspezifischen Kriterien, der sich vom Konzept des Meilensteines abgrenzt, da die Einplanung bereits beim Projektstart stattfindet und weitreichende Entscheidungen zum weiteren Verlauf des Projektes getroffen werden können. Prüfkriterien umfassen dabei die Aktivitäten und Ergebnisse des betreffenden Prozessschrittes. Je nach Sicherheit und Reife des Entwicklungs- oder Fertigungsprozesses sind die Kriterien entweder konkret oder abstrakt. Reifere und sichere Prozesse lassen dabei konkretere Kriterien zu als unsichere Prozesse. Ein Quality Gate hat zurück- und vorausschauenden Charakter; ersteres um zu koordinieren, letzteres um die Voraussetzungen für die folgenden Prozessschritte zu prüfen und Chancen und Risiken rechtzeitig zu erkennen. Rück- und Vorrückschau erfordern geeignete Methoden wie beispielsweise spezielle Reviews. Ein Quality Gate dient als Freigabepunkt zwischen Prozessschritten, der es ermöglicht, Ressourcen freizugeben, im Prozess voranzuschreiten und Maßnahmen ergreifen zu können. Zum Spektrum der möglichen Maßnahmen gehören der Abbruch, das Zurückstellen des Projektes und die Einleitung von korrigierenden Maßnahmen. Hinsichtlich der Kunden-Lieferanten-Beziehung hilft ein Quality Gate dabei, Qualitätsanforderungen an das Produkt abzustimmen. Diese Qualitätsanforderungen spiegeln sich in den Kriterien wider. Eine Durchführung von Quality Gates in einem zeitlich äquidistanten Abstand (z.B. immer monatlich) ist ebenfalls sinnvoll, um den Fortschritt fortlaufend zu erfassen und Ergebnisse mit dem Kunden abzustimmen. Ebenso ist es möglich, Quality Gates an Prozessschritten, sprich an Ergebnissen auszurichten. Hawlitzky unterscheidet Quality Gates mit interner und externer Wirkung. Erstere prüfen Qualitätsanforderungen an Produkt und Prozess aus Unternehmenssicht (z.B. Einhaltung von Unternehmensstandards), letztere prüfen Qualitätsanforderungen aus Sicht des Anwenders oder Kunden (z.B. Bedienbarkeit). Abbildung 27 zeigt beispielhaft Quality Gates nach Hawlitzky in einem inkrementellen Software-Entwicklungsprozess. Auf Phasen-Inkrement-Ebene sind diejenigen Quality Gates mit externer Sichtbarkeit zu erkennen, während innerhalb der beiden Inkremente Quality Gates mit ausschließlich interner Wirkung definiert sind.

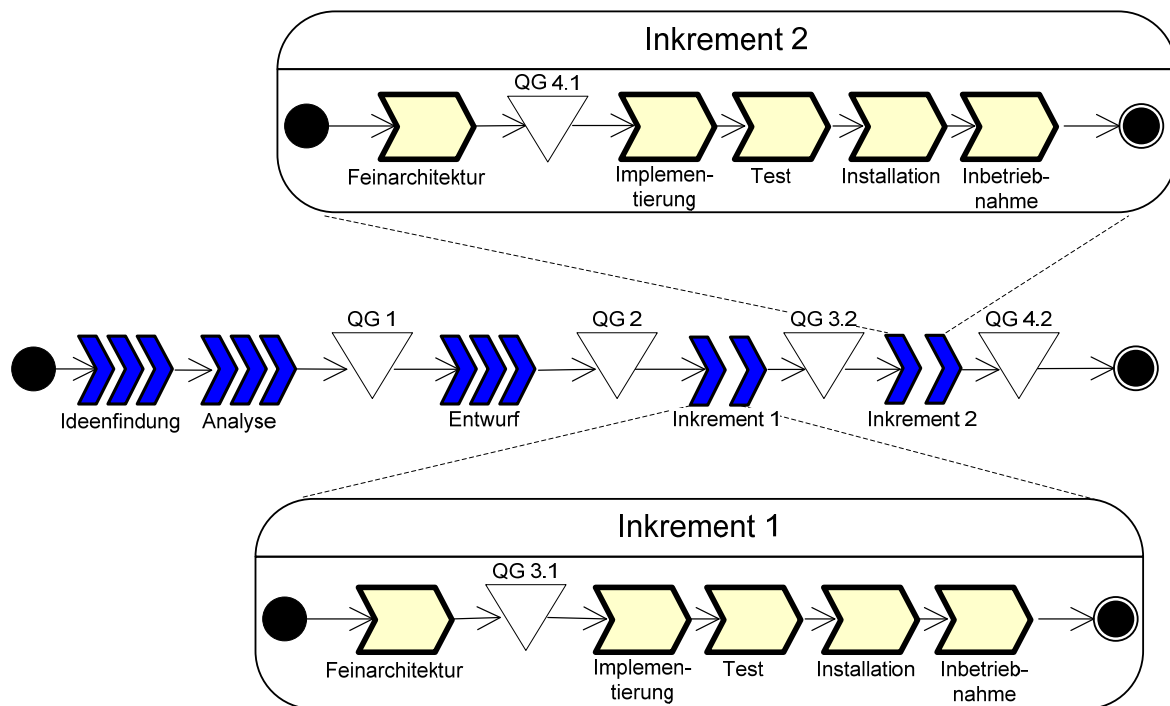


Abbildung 27: Quality Gates im Software-Entwicklungsprozess nach Hawlitzky [Hawlitzky '02]

Nach Ruffler und Leszak [Ruffler et al. '06] sind Quality Gates die herausragenden Zeitpunkte innerhalb eines Projekts, zu denen qualitätsbezogene Kriterien auf ihre Erfüllung hin geprüft werden. Quality Gates sind insbesondere an Übergabepunkten zwischen den verschiedenen technischen und fachlichen Disziplinen (z.B. Übergabe der Software zur Integration in die Hardware) zu finden. An Quality Gates können ferner parallele Entwicklungsstränge zusammengeführt werden. Innerhalb eines Gate-Reviews wird über die Erfüllung der Ausgangskriterien eines Prozessschrittes entschieden. Ein Quality Gate gilt bei Einhaltung aller qualitätsbezogenen Kriterien als erfüllt.

Charvat [Charvat '03] definiert Quality Gates im Wesentlichen als Reviews, in denen Akzeptanzkriterien geprüft werden. Ein Quality Gate schaltet die notwendigen Ressourcen formal frei, um von einer Phase eines Entwicklungsprozesses zur nächsten zu gelangen. Qualitätsbezogene Kriterien werden dabei im Vorfeld der Durchführungsphase vereinbart. Nach Charvat ist das Quality-Gate-Konzept so flexibel, dass es auch bei agilen oder in kleineren Projekten angewendet werden kann.

Darüber hinaus werden Quality Gates im V-Modell XT [VMXT '06], im Stage-Gate-Prozess nach Cooper [Cooper '01] und im Quality-Gate-Referenzprozess nach Pfeifer [Pfeifer et al. '03] definiert. Aufgrund der Relevanz dieser Ansätze für die weiteren Betrachtungen in dieser Arbeit, sind ihnen jeweils separate Abschnitte innerhalb dieses Kapitels gewidmet.

4.2.3 Zusammenfassung der Literatur-Untersuchung

Wie bereits in Abschnitt 4.2.1 festgestellt, ist das Quality-Gate-Konzept in der Software-Entwicklung nur wenig erforscht. Das Konzept ist vor allem in Entwicklungsprozessen der Automobilindustrie vorzufinden. Stellvertretend zeigt Abbildung 28 den Entwicklungsprozess des Verbands der Automobilindustrie (VDA) mit seinen Freigabepunkten. Ein weiteres Einsatzgebiet sind Systementwicklungsprojekte, also solche Projekte, bei denen die Software-Entwicklung nur einen Teil der Entwicklung ausmacht. Lediglich ein Ansatz ist speziell für die Software-Entwicklung geeignet, wengleich auch nur für wasserfallartiges Vorgehen.

Hinsichtlich der Einbettung in den Entwicklungsprozess wird ein Quality Gate überwiegend als ergebnisorientierter qualitätsfokussierter Entscheidungspunkt nach Phasen oder besonders kritischen Prozessschritten gesehen. Daher können Quality Gates bereits in einem Referenzprozess definiert werden.

Ferner kann ein Quality Gate nach Hawlitzky [Hawlitzky '02] auch kontinuierlich ereignisorientiert (z.B. immer monatlich) durchgeführt werden.

Kriterien werden idealerweise deutlich vor der Durchführungsphase festgelegt, auf jeden Fall aber spätestens vor dem Eintreten des jeweiligen Quality Gates. Je nach Blickrichtung wird ein Quality Gate als Eintritts- oder Austrittspunkt eines Prozessschrittes gesehen. Die Bezeichnung des Quality Gates gibt dabei zumeist einen Hinweis auf die Blickrichtung [Fauth et al. '99]. Rückwärtsgerichtete Quality Gates tragen den Namen des aktuellen Prozessschrittes, während vorwärtsgerichtete den Namen des nächsten Prozessschrittes tragen. Weiterhin wird der steuernde Charakter von Quality Gates hervorgehoben. Abweichungen von definierten Kriterien führen im Extremfall zum Abbruch des Projektes oder zumindest zu korrigierenden Maßnahmen.

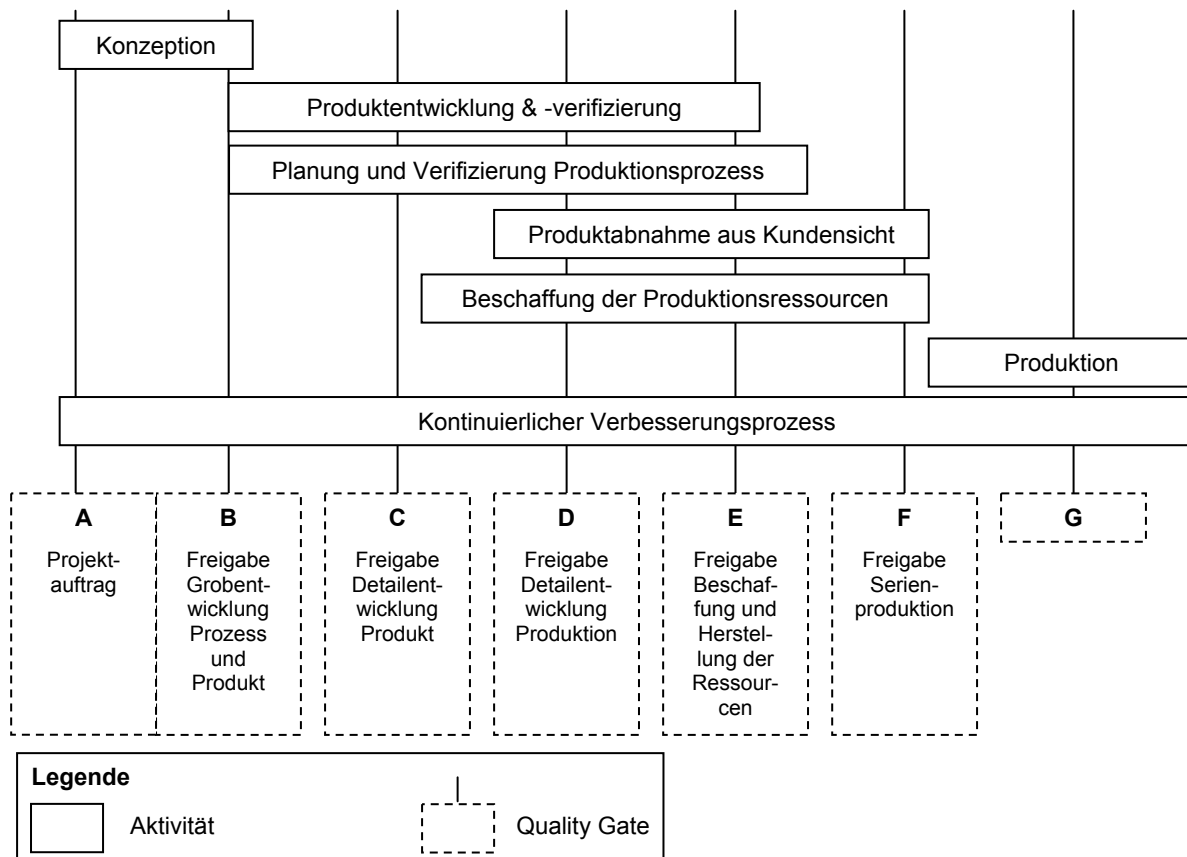


Abbildung 28: Entwicklungsprozess des VDA (nach [VDA '98])

Eine Trendwende könnte sich insofern hinsichtlich der Forschung im Bereich des Software Engineering abzeichnen, als das Quality-Gate-Konzept schon in vielen Software-Unternehmen genutzt wird, jedoch der theoretische Unterbau fehlt.

Bezogen auf die Software-Entwicklung sind die folgenden Fragestellungen von besonderem Interesse:

- Wie kann das Konzept bei iterativem oder in extremer Form bei agilem Vorgehen genutzt werden?
- Wie sind Quality-Gate-Referenzprozesse für die Software-Entwicklung hinsichtlich der notwendigen Aktivitäten, Dokument, Rollen und Hilfsmittel auszugestalten?
- An welchen Stellen sollten Quality Gates idealerweise im Entwicklungsprozess platziert werden?

4.3 Empirische Erhebung

Im Rahmen einer empirischen Erhebung, die als Teilleistung dieser Dissertation erbracht wurde, wurde der Einsatz von Quality Gates in Unternehmen untersucht, die zumindest einen Teil ihrer Wert-

schöpfung mit der Erstellung von Software erzielen. Die Erhebung fand im Zeitraum vom Oktober 2006 bis Januar 2007 statt. Vorgehen und Ergebnisse dieser Erhebung werden im Folgenden dargestellt.

4.3.1 Zielsetzungen der empirischen Erhebung

Das Hauptziel der empirischen Erhebung bestand darin, sich dem Quality-Gate-Konzept durch die Untersuchung verschiedener, in Software-Unternehmen praktizierter, Quality-Gate-Referenzprozesse induktiv zu nähern. Hierbei bestanden verschiedene Ziele mit zugehörigen Fragen (vgl. Tabelle 11).

Ziel	Fragen
Ermittlung des Grundverständnisses	<ul style="list-style-type: none"> - Wodurch lassen sich Quality Gates charakterisieren? - Welche Funktion haben Quality Gates? - Welche alternativen Bezeichnungen und verwandten Konzepte existieren?
Ermittlung der Möglichkeiten zur Ausgestaltung	<ul style="list-style-type: none"> - Wie können Quality Gates in der Praxis ausgestaltet werden? - An welchen anderen Referenzprozessen orientieren sich die vorhandenen Quality-Gate-Referenzprozesse? - In welchem Maße sind Quality Gates anpassbar an verschiedene Projektsituationen?
Verbesserungspunkte	<ul style="list-style-type: none"> - Welche Problempunkte treten bei der Verwendung von Quality Gates auf? - Welche Softwarewerkzeuge werden zur Unterstützung eingesetzt?

Tabelle 11: Ziele der empirischen Erhebung

Die Ergebnisse wurden in verschiedener Weise genutzt:

- Das Verständnis für das Quality-Gate-Konzept wurde geschärft. Dazu gehört insbesondere die Abgrenzung des Konzeptes von den Konzepten des Meilensteins und des Reviews.
- Genannte wichtige Referenzprozesse wurden hinsichtlich ihrer Ausgestaltung untersucht.
- Richtlinien, Empfehlungen und das Spektrum zur bzw. der Ausgestaltung konnten ermittelt werden.
- Ansatzpunkte für Verbesserungen wurden identifiziert.

4.3.2 Durchführung und Inhalte der empirischen Erhebung

Die Datenerhebung wurde durch eine schriftliche Umfrage realisiert. Dazu wurde ein Fragebogen konzipiert und mehrfach validiert und revidiert. Die Prüfung der Validität umfasste die Verständlichkeit und die Vollständigkeit der Fragen bezüglich einer adäquaten Abdeckung des Umfeldes. Mängel bezüglich der Validität flossen in Revisionen des Fragebogens ein. Letztlich wurde der Fragebogen an verschiedenen Stellen in deutscher und englischer Sprache angeboten:

- Auf zwei Konferenzen, die ihren Fokus auf Software Engineering legen:
 - Conference on Quality Engineering in Software Technology (*CONQUEST*) 2006
 - European Software Process Improvement Initiative (*EuroSPI*) 2006
- In der Mailingliste der *AISWorld* (Association for Information Systems) (<http://www.isworld.org>).

- In einer englischsprachigen Newsgroup, die sich mit dem Thema Software Engineering beschäftigt.
- Über die XING-Plattform (<http://www.xing.com>), über die soziale Kontakte professionell und themenbezogen aufgebaut werden können.

Darüber hinaus wurden Projekt- und Qualitätsmanager direkt per E-Mail angeschrieben, die sich mit dem Quality-Gate-Konzept beschäftigen.

Im Folgenden wird auf die inhaltliche Ausgestaltung des Fragebogens näher eingegangen. Der Fragebogen bestand aus 23 Fragen, die auf vier Frageblöcke aufgeteilt waren (vgl. Abbildung 37). Aufgrund einer vorhandenen Unsicherheit bezüglich des Quality-Gate-Konzeptes und der induktiven Vorgehensweise, enthielt ein Block ausschließlich offene Fragen. Hierdurch sollten mögliche Antworten nicht eingeschränkt werden. Ein weiterer Block des Fragebogens bestand aus geschlossenen Fragen (teilweise mit der Möglichkeit zur Ergänzung in offener Form). Ein einleitender Block umfasste die Frage nach der Position im Unternehmen sowie nach dem verwendeten Quality-Gate-Konzept. Wobei an dieser Stelle der Ansatz nur grob umrissen wurde (ein Gate als Sinnbild der „formalen“ Schranke zwischen Phasen oder Iterationen), falls kein Quality-Gate-Konzept im Unternehmen etabliert ist:

„Durch welchen Ansatz wird bei Ihnen im Unternehmen formal geprüft, ob die Eingangskriterien von Dokumenten für Phasen und Iterationen erfüllt sind?“

Im abschließenden Block wurde eine Möglichkeit zur Ergänzung gewährt, falls ein Aspekt des etablierten Quality-Gate-Referenzprozesses durch den Fragebogen nicht abgedeckt wurde. Zusätzlich konnten Kontaktdaten für eventuelle Rückfragen angegeben werden.

Abbildung 29 zeigt die Struktur des Fragebogens. Im linken Teil werden die einzelnen Frageblöcke dargestellt. Im rechten Teil werden stark verkürzt die Frageninhalte dargestellt. Der vollständige Fragebogen befindet sich im Anhang A.

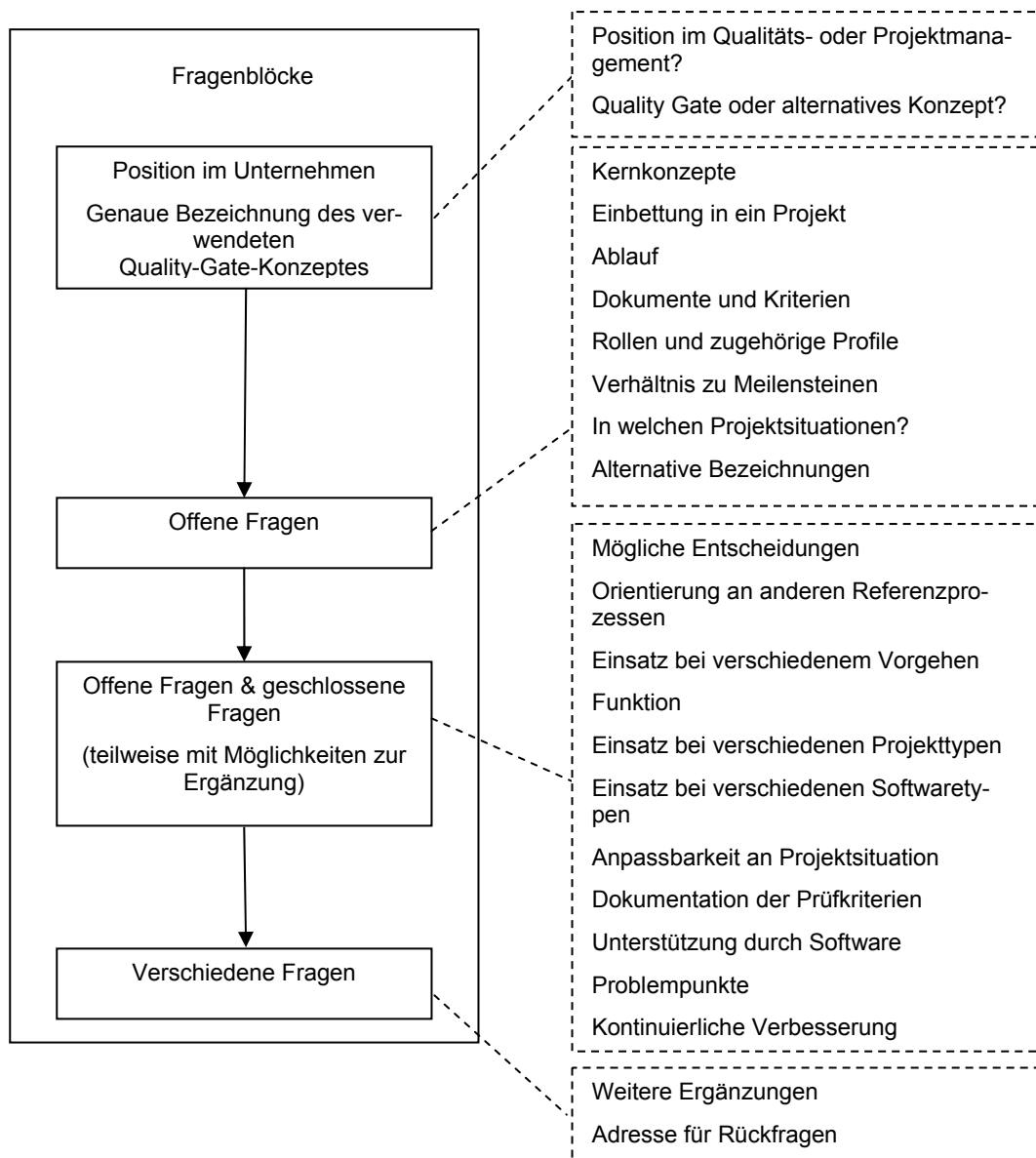


Abbildung 29: Struktur des Fragebogens

4.3.3 Auswertung

Insgesamt wurden 11 Fragebögen per E-Mail zurückgeschickt. Für die niedrige Beteiligung existieren drei mögliche Hauptursachen:

- Online-Umfragen haben häufig einen geringen Rücklauf [Richter '06]. Darüber hinaus besteht eine generelle Tendenz, Fragebögen nicht auszufüllen, selbst wenn die Umfrage in anderer Weise stattfindet.
- Der Umfang des Fragebogens war zu hoch und wirkte daher abschreckend. Insbesondere hinsichtlich der Vielzahl an offenen Fragen, deren Beantwortung wahrscheinlich zu viel Zeit beanspruchte.
- Quality Gates sind weniger weit verbreitet als angenommen oder es werden häufig unternehmenseigene Bezeichnungen verwendet. Als Konsequenz fühlten sich Unternehmen nicht angesprochen, an der Umfrage teilzunehmen.

Im Folgenden wird auf die wichtigsten Resultate der empirischen Erhebung näher eingegangen. Aus Gründen der Geheimhaltung können die Resultate nur stark anonymisiert und verdichtet dargestellt werden. Insbesondere werden keine synonymen Bezeichnungen für das Quality-Gate-Konzept wie-

dergegeben, da dies Rückschlüsse auf die Unternehmen zulässt, die an der Umfrage teilgenommen haben. Eine vollständige Datentabelle befindet sich im Anhang A.

4.3.3.1 Verwendete Referenzprozesse und kontinuierliche Verbesserung

Fünf Teilnehmer gaben direkt an, dass in ihrem Unternehmen Quality Gates eingesetzt werden. Die verbleibenden sechs Teilnehmer nannten ein – ihrer Ansicht nach – synonymes Konzept. Hierbei wurde einmal der Stage-Gate-Prozess [Cooper '01] und einmal das Meilenstein-Review [Leszak et al. '02] genannt. Die verbleibenden vier Konzepte sind hinsichtlich ihrer Bezeichnung als eigenständig anzusehen.

Einige Quality-Gate-Referenzprozesse orientieren sich an anderen Referenzprozessen. Die zugehörige Frage sollte klären, ob und, wenn ja, welche Referenzprozesse jeweils als Ausgangspunkt genutzt wurden. Insgesamt fünf Teilnehmer gaben an, dass in Ihrem Unternehmen ein eigenständiger Quality-Gate-Referenzprozess ohne Bezug zu einem anderen Referenzprozess genutzt wird. Weitere fünf Teilnehmer gaben an, dass Ihr Quality-Gate-Referenzprozess auf dem Stage-Gate-Prozess beruht. Ein weiterer Quality-Gate-Referenzprozess beruht auf dem des V-Modell XT.

Darüber hinaus wurde die Frage gestellt, ob der jeweils etablierte Quality-Gate-Referenzprozess einer kontinuierlichen Verbesserung unterliegt. Fünf Teilnehmer bejahten diese Fragen, die verbleibenden Teilnehmer machten hierzu keine Angabe.

4.3.3.2 Funktion und Einsatzgebiet

Hinsichtlich der Funktion von Quality Gates wurden verschiedene Antworten vorgegeben. Dabei konnten sich die Teilnehmer grob entscheiden, ob Quality Gates als Mittel des Projekt-, Qualitäts- oder Risikomanagements betrachtet werden können (Mehrfachnennungen waren möglich). Zusätzlich wurde Platz für Ergänzungen gewährt. Tabelle 12 zeigt die Verteilung der Ergebnisse.

Funktion	Anzahl der Nennungen
Qualitätsmanagement	7
Risikomanagement	6
Projektmanagement	5

Tabelle 12: Funktionen von Quality Gates

Darüber hinaus wiesen drei Teilnehmer auf den Synchronisationseffekt von Quality Gates hin. Ein weiterer Teilnehmer gab an, dass Quality Gates auch als Mittel des Multiprojektmanagements genutzt werden können.

Bezüglich des Einsatzgebietes wurden die Antworten „Standardsoftware“, „Eigenentwicklung“ und „Projekt mit externen Kunden“ vorgegeben. Darüber hinaus wurde nach der Art der Software gefragt. Hierbei wurden die Antworten „Anwendungssoftware“, „eingebettete Software“, „verteilte Software/Web-Anwendungen“ und „Systemsoftware“ vorgegeben.

Tabelle 13 und Tabelle 14 geben die Ergebnisse der Fragen nach dem Einsatzgebiet bzw. nach dem Softwaretyp wieder.

Einsatzgebiet	Anzahl der Nennungen
Eigenentwicklungen	5
Projekte mit externen Kunden	3
Standardsoftware	2

Tabelle 13: Einsatzgebiete, bei denen Quality Gates eingesetzt werden

Softwaretyp	Anzahl der Nennungen
Eingebettete Software	5
Verteilte Software/Web-Anwendungen	2
Anwendungssoftware	1
Systemsoftware	1

Tabelle 14: Softwaretypen, bei deren Entwicklung Quality Gates eingesetzt werden

4.3.3.3 Einbettung in den Prozess und Verhältnis zu Meilensteinen

Die Frage nach der Einbettung versucht die Lage von Quality Gates innerhalb eines Entwicklungsprozesses zu klären. Dabei spielt insbesondere das Verhältnis zu Meilensteinen eine Rolle. Bezüglich der Lage wurde das Quality Gate von zehn Teilnehmern als spezieller Meilenstein zwischen den Entwicklungsphasen eingeordnet. Zwei Teilnehmer gaben explizit an, dass ein Quality Gate auch innerhalb der Phasen (genauer zwischen Iterationen) platziert werden kann. Von den zehn Teilnehmern gaben neun Teilnehmer direkt an, dass in ihren Prozessen jeweils nur eine begrenzte Anzahl verschiedener Quality Gates zum Einsatz kommt.

Der verbleibende Teilnehmer setzte seinen genannten Ansatz mit Meilensteinen gleich, gab jedoch an, dass sich die Zeitpunkte für diese Meilensteine aus dem verwendeten Vorgehensmodell weitestgehend ergeben würden.

Von weiterem Interesse war die Frage, ob der genannte Ansatz bei wasserfallartigen oder bei iterativem Vorgehen genutzt wird. Tabelle 15 zeigt hierzu die Ergebnisse.

Vorgehensmodell	Anzahl der Nennungen
Wasserfall	4
Iterativ	2
Wasserfall und Iterativ	2
Ohne Angabe	3

Tabelle 15: Genannte Vorgehensmodelle bei denen Quality Gates genutzt werden

4.3.3.4 Ablauf

Die Frage nach dem Ablauf sollte klären, ob ein strukturiertes Vorgehen für die Erstellung von Kriterien, die Prüfung der Ergebnisse und die Entscheidungsfindung besteht. Die genannten Aktivitäten lassen sich klassifizieren in

- die *Erstellung von Kriterien* als Aktivität zur Bestimmung von relevanten Prüfkriterien und einzuhaltenden Schwellwerten,
- das *Kickoff* als einführendes Treffen für die weiteren Aktivitäten im Gate-Review,
- die *Individuelle Vorbereitung*, bei der Gutachter die Ergebnisse im Vorfeld einer Gate-Sitzung prüfen,
- die *Gate-Sitzung*, in der eine Entscheidung über das Projekt getroffen wird,
- die *Überwachung von Maßnahmen* nach einer getroffenen Entscheidung.

Tabelle 16 zeigt die Anzahl der Nennungen für die jeweiligen Aktivitäten.

Aktivitäten innerhalb des Ablaufs	Anzahl der Nennungen
Individuelle Vorbereitung	4
Gate-Sitzung	4
Überwachung von Maßnahmen	2
Erstellung von Kriterien	1
Kickoff	1

Tabelle 16: Genannte Antworten zum Ablauf

4.3.3.5 Ergebnisse und Kriterien

Mit dieser Frage sollte geklärt werden, welche Dokumente über welche Kriterien geprüft werden. Fünf Teilnehmer gaben an, dass die Durchführung von Aktivitäten und deren Ergebnisse geprüft werden. Aus drei weiteren Antworten kristallisierte sich heraus, dass es sich um keine technischen Kriterien handelt, die Kriterien formal sind und dass auch Kennzahlwerte eines Prozesses geprüft werden.

Fünf Teilnehmer nannten sehr unpräzise nur „Dokumente der aktuellen Entwicklungsphase“ als die Ergebnisse, die geprüft werden. Ein Teilnehmer enthielt sich der Antwort. Bezüglich der Kriterien wurden nur wenige verwertbare Antworten gegeben. Allerdings wurde dreimal darauf hingewiesen, dass Kriterien aus Geschäftszielen hergeleitet werden.

Hinsichtlich der Dokumentationsform der Kriterien wurde achtmal die Checkliste genannt. Drei Teilnehmer enthielten sich.

4.3.3.6 Beteiligte Rollen

Bei dieser Frage wurde auf die, an einem Quality Gate beteiligten Rollen und ihre Aufgaben und Profile eingegangen. Insgesamt neun Teilnehmer gaben Rollen an. Für jede Rolle wurden fast immer die Aufgaben genannt, jedoch nie das notwendige Profil.

Die genannten Rollen lassen sich grob unterteilen in:

- **Gate-Management:** Das Gate-Management beschäftigt sich mit der Pflege eines bereits ausgestalteten Quality-Gate-Referenzprozesses.
- **Gate-Verantwortlicher:** Der Gate-Verantwortliche ist verantwortlich für die ordnungsgemäße Durchführung aller Aktivitäten, die mit dem Gate-Review in Zusammenhang stehen.
- **Projektvertreter:** Ein Projektvertreter ist ein Repräsentant des Projektes, der am Gate-Review teilnehmen kann, um die Interessen des Projektes zu wahren. Zu seinen Aufgaben gehören unter anderem das Einreichen und die Vorstellung der Ergebnisse.
- **Gutachter:** Ein Gutachter bewertet die für das Quality Gate vorzulegenden Ergebnisse. Ein Gutachter kann keine Entscheidungen treffen, jedoch kann er eine Entscheidung oder Maßnahmen empfehlen.
- **Gatekeeper:** Gatekeeper sind für das Fällen der Entscheidung verantwortlich.

Tabelle 17 gibt einen Überblick über die Anzahl der Nennungen pro Rolle.

Rolle	Anzahl der Nennungen
Gutachter	8
Projektvertreter	7
Gatekeeper	6
Gate-Verantwortlicher	5
Gate-Management	1

Tabelle 17: Verteilung der genannten Rollen

4.3.3.7 Anpassbarkeit an Projektsituation

Ziel dieser Frage war die Klärung, bei welchen Projektsituationen der genannte Quality-Gate-Referenzprozess überhaupt eingesetzt wird. Drei Teilnehmer gaben an, dass ihr Quality-Gate-Referenzprozess immer eingesetzt wird. Drei weitere Teilnehmer nannten die Wichtigkeit und Größe als das entscheidende Kriterium für den Einsatz eines Quality Gates. Ein Teilnehmer nannte die Ressourcen Zeit und Geld als ausschlaggebende Kriterien.

Weiterhin sollte geklärt werden, inwieweit der jeweilig etablierte Quality-Gate-Referenzprozess an verschiedene Projektsituationen angepasst werden kann. Drei Anpassungsmöglichkeiten wurden vorgegeben, wobei Mehrfachnennungen und eine Ergänzung um individuelle Antworten möglich waren:

- Die Anzahl der Quality Gates ist projektspezifisch.
- Die Quality-Gate-Kriterien sind projektspezifisch.
- Die Rollen sind projektspezifisch besetzt.

Tabelle 18 gibt einen Überblick über die Anzahl der Nennungen.

Anpassungsmöglichkeit	Anzahl der Nennungen
Anzahl Quality Gates ist projektspezifisch	6
Kriterien jedes Quality Gates projektspezifisch	5
Die Rollen sind projektspezifisch besetzt	2
Ohne Angabe	2
Ansatz wird immer gleich verwendet	1

Tabelle 18: Anpassbarkeit an Projektsituation

4.3.3.8 Mögliche Entscheidungen

Hierbei sollte geklärt werden, welche Entscheidungen prinzipiell in einem Quality Gate getroffen werden können. Insgesamt wurden vier mögliche Entscheidungen angegeben (Mehrfachnennungen waren möglich). Zusätzlich wurde Freiraum zur Ergänzung gegeben. Folgende Antwortmöglichkeiten waren vorgegeben:

- **Uneingeschränkte Projektfortsetzung („go“):** Das Go bezeichnet die formelle und eingeschränkte Freigabe von Ressourcen für den nächsten Prozessschritt.
- **Projektabbruch („kill“):** Der Projektabbruch ist dann zu wählen, wenn eine Vielzahl sehr wichtiger Kriterien nicht erfüllt wurden und nicht absehbar ist, dass sie noch erfüllt werden können.
- **Zurückstellung des Projektes („hold“):** Bei der Zurückstellung handelt es sich um eine schwächere Form des Projektabbruchs. Es werden Ressourcen vom Projekt abgezogen, wobei aber damit zu rechnen ist, dass das Projekt zukünftig fortgesetzt wird.

- **Wiederholung des Quality Gates („repeat-gate“):** Die Wiederholung eines Quality Gates ist dann zu wählen, wenn ein Projekt wichtige Kriterien momentan nicht erfüllen kann, sie jedoch in absehbarer Zeit erfüllt werden können.

Tabelle 19 gibt einen Überblick über die Nennungen.

Entscheidung	Anzahl der Nennungen
Uneingeschränkte Projektfortsetzung	10
Projektabbruch	5
Zurückstellung des Projektes	5
Wiederholung des Quality Gates	4

Tabelle 19: Verteilung der genannten Ausgänge

4.3.3.9 Unterstützung durch Software

Unterstützende Software hilft bei der Anpassung, der Einplanung, Instanziierung, Durchführung und Überwachung von Quality Gates bzw. der Gate-Sitzung. Lediglich ein Teilnehmer gab ein Softwarewerkzeug an, das alle der genannten Aspekte unterstützt. Drei weitere Teilnehmer nannten Werkzeuge, die die Datensammlung, Verdichtung und Berichterstellung unterstützen. Ein weiterer Teilnehmer gab ein klassisches Projektmanagement-Werkzeug als unterstützende Software an. Zwei Teilnehmer gaben explizit an, dass sie kein Softwarewerkzeug nutzen. Ein Teilnehmer zweifelt hingegen sogar die Effizienz derartiger Software an. Die verbleibenden vier Teilnehmer nannten kein Werkzeug.

4.3.3.10 Problempunkte

Die Frage nach Problempunkten von Quality Gates wurde offen gestellt. Die folgende Aufzählung gibt die wesentlichen Schwachpunkte wider, wobei sich Problempunkte grob klassifizieren lassen nach Kriterien und Ergebnisse, Gate-Review, Steuerung und Prozess.

Kriterien und Ergebnisse

- Falls Kriterien projektunabhängig definiert sind, tritt das Problem auf, dass die Kriterien zu abstrakt sind. Die notwendige Operationalisierung vorläuft häufig subjektiv und unsystematisch.
- Es ist nur im begrenzten Maße möglich, formale Kriterien für alle zu prüfenden Aspekte zu finden. Daher ist der Einbindung von Experten notwendig, die Ergebnisse „auch notfalls“ ohne formale Kriterien prüfen können.
- Die Lieferung von Kennzahlwerten in der richtigen Detaillierungsstufe und Vollständigkeit ist entscheidend. Dabei erweisen sich zu stark verdichtete oder zu detaillierte Kennzahlwerte als unbrauchbar und resultieren häufig in ineffektiven Prüfungen. Es ist schwierig, das richtige Abstraktionsniveau für Kennzahlen zu finden.

Gate-Review

- Die Effektivität eines Quality Gates hängt entscheidend von einer systematischen Durchführung der Prüfungsaktivitäten ab. Leider werden die Prüfungsaktivitäten häufig nicht systematisch genug durchgeführt.
- Es besteht die Gefahr, dass Gate-Reviews uneinheitlich in unterschiedlichen Projekten durchgeführt werden, insbesondere wenn der Ansatz in verschiedenen Fachabteilungen durchgeführt wird. Infolgedessen kommt es zu unterschiedlichen Entscheidungen trotz vergleichbarer Ausgangslage.
- Beim Erreichen eines Quality Gates dürfen Entwicklungsaktivitäten nicht aufgrund der notwendigen Prüfaktivitäten angehalten werden. Leider wirkt ein Quality Gate psychologisch häufig als stoppende Schranke.

Steuerung

- Entscheidungsträger neigen dazu, Projekte in den späteren Phasen der Entwicklung nicht mehr abzubrechen, weil bereits ein Großteil der Ressourcen in das Projekt geflossen ist. Dies schränkt die Effektivität des Ansatzes ein.
- Ebenso ist es möglich, dass ein Projekt zu früh gestoppt wird, wenn Quality Gates bereits in den frühen Phasen eingesetzt werden. Das Problem tritt dann auf, wenn die Kriterien zu hart sind.
- Es ist nicht immer offensichtlich, wer über den weiteren Verlauf eines Projektes entscheiden darf. Dies betrifft insbesondere Quality-Gate-Referenzprozesse, die vergleichsweise abstrakt definiert sind und dann in den einzelnen Unternehmensbereichen angepasst und instanziiert werden.

Prozess

- Der Einsatz von Quality Gates wird häufig mit wasserfallartigem Vorgehen in Verbindung gebracht. Dies schränkt den Willen zum Einsatz bei Projekten ein, die nicht diesem Vorgehen folgen, obwohl dies möglich ist.
- Die Synchronisation von projektspezifischen Meilensteinen und Quality Gates ist nicht immer möglich. Dies führt zu redundanten Prüfkriterien oder zum Fehlen von Daten im Quality Gate, da bestimmte Prozessschritte noch nicht durchgeführt wurden, aber für ein Quality Gate verlangt werden.

4.3.4 Zusammenfassung und Bewertung der Ergebnisse

Aufgrund der recht niedrigen Rücklauf-Quote, können keine statistisch validen Aussagen getroffen werden. Dennoch können bestimmte Vermutungen getroffen werden. Bezüglich der Qualität der Daten ist festzustellen, dass ein Teil der Fragen nur unzureichend oder gar nicht beantwortet wurde. Dies stellt eine weitere Gefahr für die Validität der Ergebnisse dar. Wo immer möglich, wurden die Daten durch Rückfrage ergänzt. Im Folgenden werden die wesentlichen oder markanten Erkenntnisse zusammengefasst werden.

- **Funktion und Einsatzgebiet:** Hinsichtlich der Funktion ergibt sich kein deutliches Bild, obwohl der Begriff Quality Gate eine besondere Nähe zum Begriff Qualität suggeriert, spiegelt sich dies in den Ergebnissen der Umfrage nicht wider. Nahezu im gleichen Maße wird das Quality Gate dem Projekt-, Qualitäts- und Risikomanagement zugeordnet. Dabei ist jedoch unklar, mit welcher Intensität die verschiedenen Funktionen jeweils verfolgt werden, da keine Daten hierzu erhoben wurden. Eine Einschränkung der Verwendung des Quality Gates auf spezielle Projekttypen oder auf die Entwicklung bestimmter Softwaretypen ist nicht erkennbar. Die Untersuchung der Projekttypen ist daher interessant, weil Eigenentwicklungen und Projekte mit externen Kunden grundsätzlich andere Gatekeeper voraussetzen. Bei letzterem Projekttyp kann die Mitwirkung des Kunden als Gatekeeper nötig sein, da ohne seine Zustimmung ein Projekt nicht abgebrochen werden kann. Im weiteren Verlauf der Arbeit ist daher zu klären, wie die Rolle Gatekeeper besetzt werden muss, damit eine angemessene Entscheidung getroffen werden kann.
- **Referenzprozesse, Verbesserung und Verhältnis zum Prozess:** Fast alle genannten Quality-Gate-Referenzprozesse sind eigenständig oder orientieren sich am Stage-Gate-Prozess. Quality Gates werden bereits bei iterativem Vorgehen genutzt. Häufiger wurde dennoch der Einsatz bei wasserfallartigem Vorgehen genannt. Kontinuierliche Verbesserungsaktivitäten werden nur von rund der Hälfte der Unternehmen durchgeführt. Diese Aussage ist jedoch mit Vorsicht zu genießen, da die anderen Teilnehmer hierzu keine Angaben machten.
- **Ablauf:** Klare quantitative Aussagen bezüglich des Ablaufs können nicht getroffen werden, da die Anzahl der Antworten hierzu besonders niedrig ausfiel. Dennoch lassen sich fünf Aktivitäten identifizieren: Kriterienerstellung, Kickoff, individuelle Vorbereitung, Gate-Sitzung

und die Überwachung von Maßnahmen. Nach Anzahl der Nennungen sind die individuelle Vorbereitung und die Gate-Sitzung von besonderer Bedeutung.

- **Ergebnisse und Kriterien:** Eine Betrachtung der Prüfinhalte liefert keine deutlichen Resultate – jedoch wurde mehrfach darauf hingewiesen, dass die Kriterien aus Geschäftszielen abgeleitet werden sollten. Selten wurde erwähnt, dass die Kriterien formal und nicht technisch sind und sich auf Kennzahlwerte eines Entwicklungsprozesses beziehen.
- **Rollen:** Die Auswertung der genannten Rollen und Aufgaben zeigt ein klareres Bild. Vier Rollen sind hier von Wichtigkeit: der Gate-Verantwortliche, der Projektvertreter, der Gutachter sowie der Gatekeeper. Lediglich einmal wurde die Rolle Gate-Management genannt. Dies lässt die Vermutung zu, dass die Rolle des Gate-Managements vielerorts noch ungeklärt ist oder ein solches nicht existiert
- **Anpassbarkeit an Projektsituation:** Mehrfach wurde angegeben, dass der Einsatz von Quality Gates sich nach der Wichtigkeit des Projektes richtet. Ebenso häufig wurde angegeben, dass Quality Gates immer eingesetzt werden. Häufig sind die Anzahl der Quality Gates sowie ihre Kriterien projektspezifisch. Nur selten werden Rollen projektspezifisch besetzt.
- **Unterstützung durch Software:** Software, die die Anpassung, die Einplanung, Instanziierung, Durchführung und Überwachung von Gates bzw. des Gate-Reviews ganzheitlich unterstützt, wird nahezu nie eingesetzt. Gelegentlich wird Software jedoch zur Datenerhebung, Verdichtung und zur Erstellung von Berichten eingesetzt.
- **Mögliche Entscheidung:** Hinsichtlich der Entscheidungen, die im Quality Gate getroffen werden können, wurde besonders häufig die Freigabe von Ressourcen („go“) genannt. Rund die Hälfte nannte den Projektabbruch, die Zurückstellung eines Projektes sowie die Wiederholung eines Quality Gates als weitere mögliche Entscheidungen.

4.3.5 Ausgangspunkte für die weiteren Untersuchungen

Der wesentliche Ausgangspunkt für weitere Untersuchungen in dieser Arbeit sind die Referenzprozesse, die durch die Teilnehmer mehr oder weniger explizit genannt wurden:

- Der Quality-Gate-Referenzprozess des Stage-Gate-Prozesses nach Cooper aufgrund der häufigen Nennung und weiten Verbreitung.
- Der Quality-Gate-Referenzprozess des V-Modells XT, der ebenfalls genannt wurde und innerhalb Deutschlands eine bedeutende Rolle als Vorgehensmodell zur Software-Entwicklung einnimmt.
- Der Quality-Gate-Referenzprozess nach Pfeifer, der ganz besonders den Synchronisationseffekt von Quality Gates betont.

Diese drei Quality-Gate-Referenzprozesse bieten bereits eine gute Abdeckung hinsichtlich ihrer Bedeutung und Verbreitung, so dass eine Untersuchung dieser Referenzprozesse einen guten Rahmen für alle weiteren Betrachtungen bildet.

Hinsichtlich der Einbettung in bestimmte Entwicklungsprozesse ist von Relevanz, wie Quality Gates bei iterativem oder agilem Vorgehen genutzt werden können. Dazu ist die Untersuchung iterativer und agiler Vorgehensweisen notwendig.

Ein weiterer wichtiger Punkt ist die Untersuchung der Prüfkriterien. Hierzu lieferte die empirische Erhebung keine deutlichen Ergebnisse, insbesondere hinsichtlich der Abgrenzung von den Prüfinhalten eines technischen Reviews.

Weiterhin ist zu untersuchen, inwieweit der Einsatz von Software zur Anpassung, Einplanung, Instanziierung, Durchführung und Überwachung von Quality Gates bzw. des Gate-Reviews sinnvoll und möglich ist. Ebenso bedarf der Einsatz kontinuierlicher Verbesserungsmaßnahmen einer näheren Untersuchung.

Die genannten Problempunkte sind durch das zu entwickelnde Framework anzusprechen und zu validieren.

Die folgenden drei Abschnitte beschäftigen sich mit den drei (oben genannten) relevanten Quality-Gate-Referenzprozessen.

4.4 Quality-Gate-Referenzprozess bei interdisziplinären Projekten

Pfeifer et al. [Pfeifer et al. '03] definieren einen Quality-Gate-Referenzprozess, der insbesondere für den Einsatz in Projekten geeignet ist, die sich aus Teilprojekten verschiedener Fachdisziplinen zusammensetzen. Der Quality-Gate-Referenzprozess nach Pfeifer folgt der Strategie *Quality Gates als flexible Qualitätsstrategie*.

4.4.1 Hintergrund und Überblick

Der Fokus des Quality-Gate-Referenzprozess nach Pfeifer liegt auf der Entwicklung von softwareintensiven Systemen. Dabei geht es um die Einbettung von Software in komplexe technische Umgebungen, wie beispielsweise Maschinen und Fahrzeuge. Hierbei sind die Qualitätsanforderungen an Sicherheit und Zuverlässigkeit besonders hoch. Der Quality-Gate-Referenzprozess geht damit auf die rapide steigende Bedeutung von Software in technischen Systemen ein. Dabei ist besonders die Zusammenarbeit der Entwickler aus den beteiligten Fachdisziplinen Mechanik, Elektrotechnik und Software-Entwicklung ein entscheidender Erfolgsfaktor. Quality Gates dienen dabei als Synchronisationspunkte auf Phasenebene (vgl. Abbildung 30).

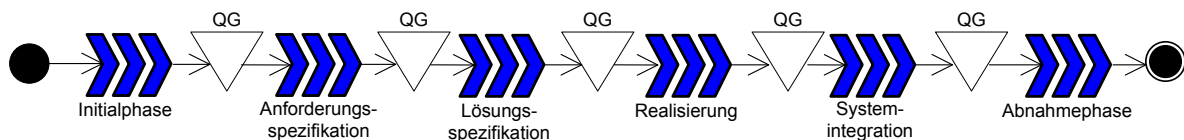


Abbildung 30: Quality Gates nach Pfeifer im Entwicklungsprozess

Ein besonderes Augenmerk liegt auf den frühen Entwicklungsphasen, da hier der Grundstein für die spätere Entwicklung gelegt wird. Bekanntermaßen liegen in den frühen Phasen nach Frühauf et al. [Frühauf et al. '04] das größte Potential und damit auch die größten Risiken verborgen. Aus diesem Grund strukturiert der Quality-Gate-Referenzprozess nach Pfeifer bereits die frühen Phasen der Entwicklung. Besonders der Umgang mit Anforderungen an Qualität und Funktionalität und die Festlegung finanzieller Rahmenbedingungen sind nach Pfeifer von entscheidender Bedeutung. Da also nicht nur qualitätsorientierte Kriterien geprüft werden, können Quality Gates nach Pfeifer zu unfokussierten Entscheidungspunkten verschwimmen.

4.4.2 Ausgestaltung des Quality-Gate-Referenzprozesses nach Pfeifer

Quality Gates im Quality-Gate-Referenzprozess nach Pfeifer dienen der Synchronisation verschiedener parallel verlaufender Entwicklungsstränge. Dabei stellt jeder Strang die Entwicklungsaktivitäten einer Fachdisziplin dar. Abbildung 31 zeigt exemplarisch die ersten beiden Phasen mit jeweils synchronisierenden Quality Gates in einem Projekt, an dem die drei Fachdisziplinen Elektronik, Mechanik und Software beteiligt sind.

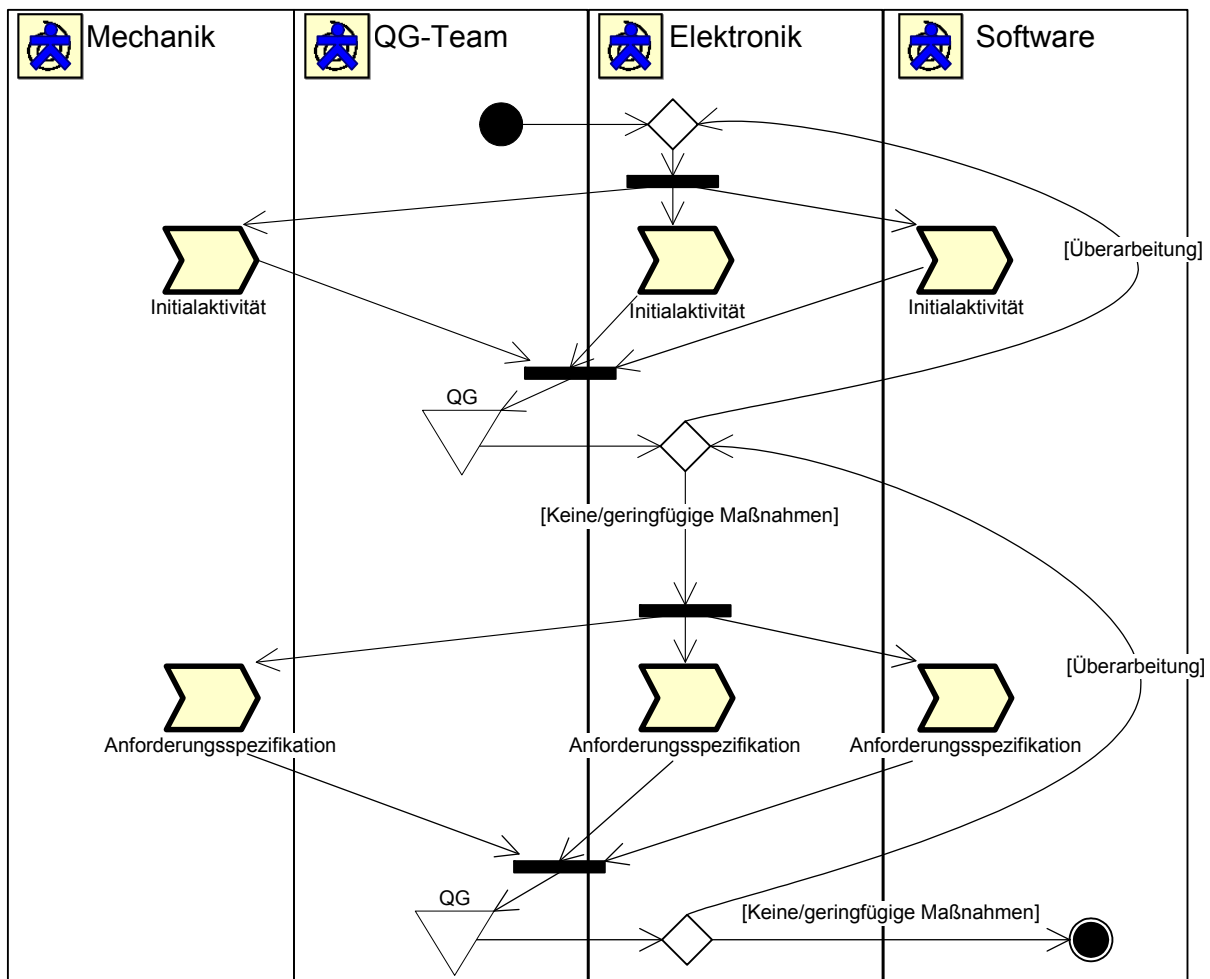


Abbildung 31: Ausschnitt eines Entwicklungsprojektes mit Quality Gates nach Pfeifer

Zwischen den Quality Gates innerhalb eines Entwicklungsstrangs lässt der Quality-Gate-Referenzprozess nach Pfeifer Spielraum für die Einplanung von projektspezifischen Meilensteinen und technischen Reviews. Darüber hinaus können weitere Synchronisationspunkte gesetzt werden, die die Abstimmung zwischen mindestens zwei der Fachdisziplinen bereits im Vorfeld eines Quality Gates ermöglichen.

Wird ein Quality Gate erreicht, so werden Ergebnisse von Vertretern aller beteiligten Fachdisziplinen zusammen mit dem Kunden innerhalb eines so genannten Gate-Reviews überprüft. Zusammen entscheiden sie über den weiteren Verlauf des betroffenen Projektes. Um die interdisziplinäre Prüfung zu erleichtern ist es wichtig, übergreifende Beschreibungstechniken zu vereinbaren und Fachvokabular verständlich aufzuschlüsseln. Die Prüfung findet auf Basis von zu Beginn der Phase definierten Kriterien statt. Eine Freigabe zur Fortführung des Projektes wird nur dann erteilt, wenn die Kriterien ausreichend erfüllt sind. Entscheidungsunterstützend wirkt hierbei eine dreistufige Ampel (Farben rot, gelb, grün), die verdichtet den Erfüllungsgrad aller Kriterien angibt.

Nach Pfeifer sind drei Aspekte für ein Projekt von besonderer Bedeutung:

- Die Orientierung der Projektbeteiligten an klar definierten und messbaren Kriterien.
- Die systematische und interdisziplinäre Synchronisation von Zwischenergebnissen.
- Die Fortschrittsbewertung aus Sicht des Projektmanagements.

Um diese Aspekte umzusetzen wird parallel zu den Entwicklungsphasen im Vorfeld eines Quality Gates dreistufig vorgegangen (vergleiche Abbildung 32):

1. **Ergebnisse vereinbaren:** In diesem Schritt werden die an einer Phase beteiligten Personen identifiziert. Dabei werden Lieferanten und interne und externe Kunden ermittelt. Diese Kun-

den definieren die Endergebnisse der Phase, die durch die Lieferanten zu erstellen sind. Nach Pfeifer sind die Ergebnisse (soweit wie möglich) feingranular festzulegen. Für jedes Ergebnis werden Kriterien und Messvorschriften festgelegt. Basierend darauf wird eine, für alle beteiligten Personen verständliche, Checkliste für das nächste Quality Gate erstellt.

2. **Weg darlegen:** In diesem Schritt wird eine Feinplanung der folgenden Aktivitäten durchgeführt. Dies beinhaltet auch die Einplanung von Meilensteinen und Synchronisationspunkten.
3. **Fortschritt synchronisieren:** Die Synchronisation des Projektfortschritts wird an Synchronisationspunkten im Vorfeld eines Quality Gates durchgeführt. Mit Hinblick auf die Checkliste des nächsten Quality Gates, wird zu diesem Zeitpunkt der Erfüllungsgrad der Kriterien eingeschätzt. Die Einschätzung wird von den Entwicklern vorgenommen und dient dem rechtzeitigen Erkennen von Risiken.

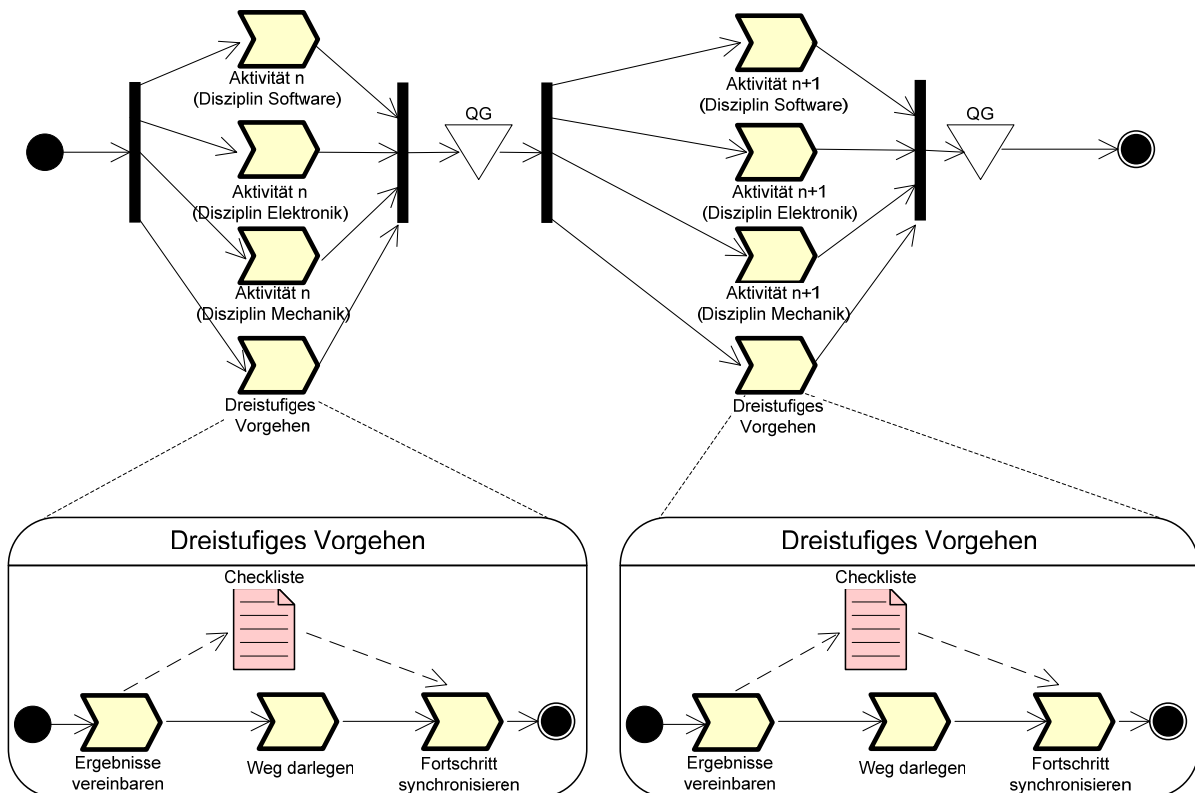


Abbildung 32: Dreistufiges Vorgehen als Ergänzung zu den Entwicklungsphasen

Dem dreistufigen Vorgehen folgt ein Quality Gate mit seinem Gate-Review. Innerhalb des Gate-Reviews wird eine Prüfung der Ergebnisse durch die internen oder externen Kunden und die Vertreter aller Fachdisziplinen durchgeführt. Dabei wird jedes vereinbarte Kriterium auf seine Erfüllung hin überprüft. Gegebenfalls werden Maßnahmen verpflichtend festgesetzt, die zeitnah erfüllt werden müssen. Das Quality Gate kann in diesem Fall passiert und das Projekt fortgesetzt werden. Können Abweichungen nicht zeitnah beseitigt werden, so ist eine größere Überarbeitung notwendig. Das betreffende Quality Gate muss dann wiederholt werden.

4.5 Quality-Gate-Referenzprozess des V-Modells XT

Das V-Modell ist ein von der Bundesrepublik Deutschland seit 1986 entwickeltes Vorgehensmodell zur Planung und Durchführung von Entwicklungsprojekten, die sich inhaltlich mit IT-Systemen befassen. Es wird von der Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung (KBSt) bereitgestellt.

4.5.1 Hintergrund und Überblick

Inzwischen liegen drei Versionen des V-Modells vor: das V-Modell, das V-Modell 97 und das V-Modell XT [VMXT '06] (XT steht dabei für „extreme tailoring“). Seit Anfang 2005 ersetzt das V-Modell XT in der Version 1.0 und seit Anfang 2006 in der Version 1.2 alle vorhergehenden Versionen. Da das ursprüngliche V-Modell und das V-Modell 97 inzwischen veraltet sind und auch kein Konzept für Quality Gates anbieten, werden sie im Weiteren nicht betrachtet. Alle weiteren Untersuchungen beziehen sich daher auf die aktuelle Veröffentlichung (Version 1.2) des V-Modells XT.

Grob gesprochen ist das V-Modell XT die verbindliche Grundlage des Vorgehens bei IT-Entwicklungsprojekten, die sowohl Bundesbehörden als auch Unternehmen der Privatwirtschaft als Geschäftspartner einbeziehen. Dabei werden alle Aktivitäten von der behördlichen Genehmigung eines Projektes, der Ausschreibung, der Projektbeauftragung, der eigentlichen Entwicklung bis hin zum Projektabschluss abgedeckt.

Namens gebend für das V-Modell XT ist die charakteristische V-förmig angeordnete Darstellung von Quality Gates (schematisch dargestellt in Abbildung 35 rechts). Im V-Modell XT bildet eine Projektdurchführungsstrategie den groben Rahmen für die Durchführung eines konkreten Projektes, indem sie die Anordnung, Anzahl und Auswahl der Quality Gates vorgibt. Eine Projektdurchführungsstrategie wird im späteren Verlauf als ein Gate-Netzwerk identifiziert und weiter formalisiert werden (vgl. Abschnitt 6.1). Die zu wählende Projektdurchführungsstrategie hängt von der Projektsituation ab. Um den verschiedenen Bedürfnissen eines Projektes gerecht zu werden, ist es möglich, ein weites Spektrum vom wasserfallartigen bis zum agilen Vorgehen abzudecken. Weiterhin deckt das V-Modell XT verschiedene Projekttypen der Systementwicklung mit Auftraggeber, Auftragnehmer und einer Kombination aus Auftraggebern und Auftragnehmer ab.

4.5.2 Ausgestaltung des Quality-Gate-Referenzprozesses im V-Modell XT

Das V-Modell XT kennt die Begriffe Entscheidungspunkt, Quality Gate und Qualitätsmesspunkt. In der Terminologie des V-Modells XT werden diese Begriffe synonym verwendet. Der Quality-Gate-Referenzprozess im V-Modell XT folgt der Strategie *Quality Gates als flexible Qualitätsstrategie*, da die Wahl der Projektdurchführungsstrategie von der Projektsituation abhängt.

Durch die Verwendung von Quality Gates werden der Projektfortschritt und das Gesamtrisiko für das Management sichtbar gemacht. Risikominimierend wirkt sich die ergebnisorientierte Ausrichtung der Quality Gates aus, da bei Abweichungen entsprechend reagiert werden kann. Die Anordnung, Anzahl und Auswahl der Quality Gates wird durch eine Projektdurchführungsstrategie vorgegeben. Neben der Festlegung von Quality Gates durch die Projektdurchführungsstrategie, kann der Projektleiter weitere projektspezifische Planungen vornehmen. Dies beinhaltet insbesondere die Definition von projektspezifischen Meilensteinen.

Jedes Quality Gate verlangt eine Menge von Produkten, die den Bearbeitungszustand „fertig gestellt“ aufweisen. Befinden sich alle Produkte in diesem Status, so kann eine Entscheidung über den weiteren Verlauf des Projektes getroffen werden. Die Entscheidung wird durch den Projektmanager in Absprache mit verschiedenen (entscheidungsbefugten) Stakeholdern und dem Projektleiter in der Aktivität „Projektfortschrittsentscheidung herbeiführen“ (dem Gate-Review) auf Grundlage von objektiven Kriterien getroffen (Abbildung 33). Ausgangsbasis für die Entscheidung sind die für ein Quality Gate durch den Projektleiter vorzulegenden Produkte. Auf diese Weise gibt eine Projektdurchführungsstrategie die Reihenfolge der zu erstellenden Dokumente vor. Im V-Modell XT definierte Aktivitäten arbeiten auf die Erstellung dieser Dokumente hin. Diese Aktivitäten sind in so genannten Vorgehensbausteinen definiert.

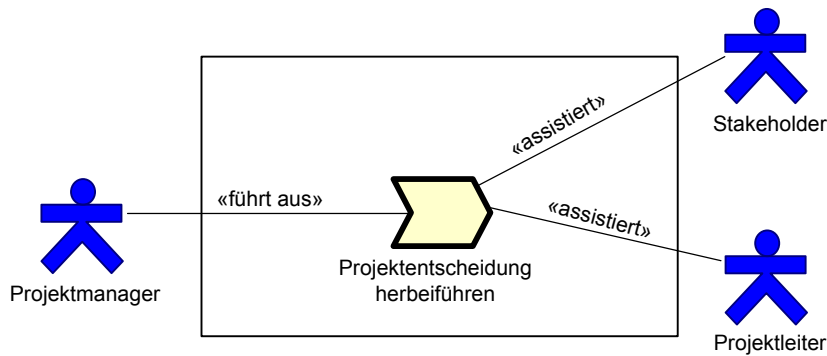


Abbildung 33: Beteiligte Rollen an der Aktivität „Projektfortschrittsentscheidung herbeiführen“

Die vorzulegenden Produkte werden anhand von vorher festgelegten Kriterien geprüft. Das V-Modell XT gibt nicht vor, welcher Art die Kriterien sein dürfen. Daher kann ein Quality Gate leicht zum unfokussierten Entscheidungspunkt ausarten. Festgelegt sind hingegen die Produkte, die für jedes Quality Gate einzureichen sind. Sie hängen jedoch zum Teil von der Projektsituation ab.

Je nach Bewertung des Fortschritts und des Risikos kann ein Quality Gate erneut durchgeführt werden. In diesem Fall sind die geforderten Produkte erneut vorzulegen. Ebenfalls kann das Projekt abgebrochen werden. Fällt die Entscheidung positiv aus, so kann das Projekt uneingeschränkt oder mit Auflagen fortgesetzt werden. Letzteres bedeutet, dass die Produkte nebenläufig geprüft werden. Bei einer positiven Entscheidung werden auf jeden Fall das Budget und weitere Ressourcen für den nächsten Projektschritt freigegeben. Eine positive Entscheidung ist gleichbedeutend mit dem Erreichen einer neuen Fortschrittsstufe im Projekt. Daher sind alle Quality Gates mit dem Status bezeichnet, der bei einer positiven Entscheidung erreicht wird (z.B. „System spezifiziert“).

Wie bereits angedeutet, wird die Entscheidung über die Fortsetzung eines Projekts in der Aktivität „Projektfortschrittsentscheidung herbeiführen“ getroffen. Diese unterteilt sich in verschiedene Aktivitäten (vgl. Abbildung 34). Eingeleitet wird die Aktivität durch die Vorlage der notwendigen Produkte durch den Projektleiter. In einer Agenda werden die notwendigen Teilnehmer bestimmt und zu einer Sitzung eingeladen, in der die bisherigen Projektergebnisse präsentiert werden. Es ist ebenfalls möglich, die Entscheidung komplett ohne Sitzung zu treffen, z.B. per Umlauf. Die Entscheidung wird durch den Lenkungsausschuss gefällt. Ihm gehören mindestens der Projektmanager und der Projektleiter an. In dieser Minimalbesetzung ist der Lenkungsausschuss bereits beschlussfähig. Sollte jedoch der Abbruch des Projektes zur Diskussion stehen, so müssen dem Lenkungsausschuss weitere Stakeholder mit weiter reichenden Entscheidungsbefugnissen angehören.

Letztlich wird das Dokument „Projektfortschrittsentscheidung“ erstellt. Es beinhaltet Angaben zur Bewertung des Projektes, zur inhaltlichen und zeitlichen Planung, zur Ressourcenplanung und zu Vorgaben und Rahmenbedingungen. Letztere gehen dabei auf vorgeschriebene technische Lösungen und auf die Einhaltung von Sicherheitsrichtlinien und Gesetzen ein. In der inhaltlichen und zeitlichen Planung wird unter anderem auf die weiteren Quality Gates und die jeweils dann vorzulegenden Dokumente eingegangen.

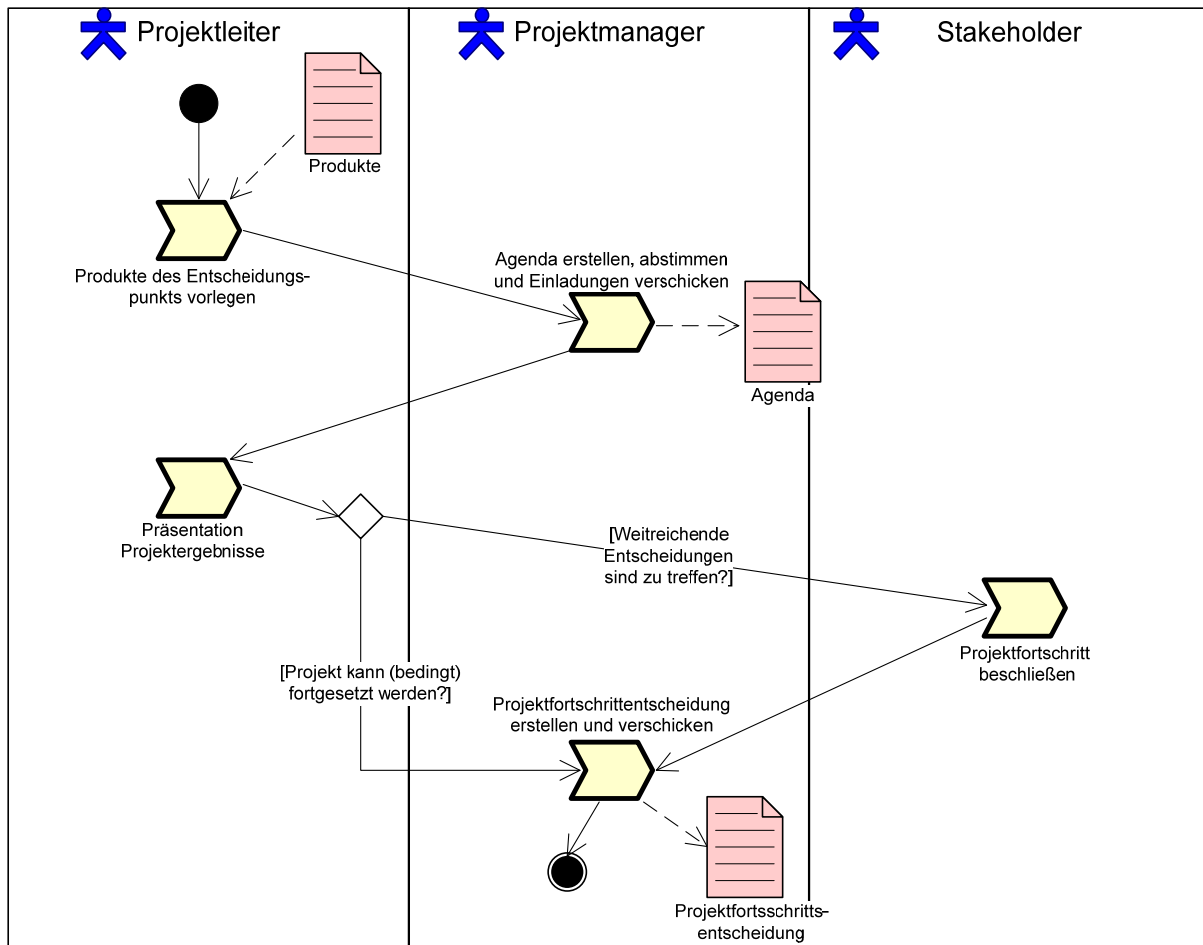


Abbildung 34: Aktivität „Projektfortschrittsentscheidung herbeiführen“

4.5.3 Anpassung des Quality-Gate-Referenzprozesses im V-Modell XT

Das V-Modell XT definiert insgesamt 21 verschiedene Quality Gates, allerdings werden je nach Projektsituation bestimmte Quality Gates gewählt und andere ausgelassen. Einige Quality Gates gelten hingegen für alle Projektsituationen. Für diese Arbeit sind nur solche Projektsituationen von Interesse, die sich mit der Systementwicklung (und damit mindestens zu einem Teil mit der Software-Entwicklung) beschäftigen.

Abbildung 35 zeigt schematisch die Anordnung der Quality Gates in der charakteristischen V-Form im rechten Teil. Weiß eingefärbte Quality Gates sind stets vorhanden. Hellgrau eingefärbte Quality Gates liegen auf der Schnittstelle zwischen Auftragnehmer und Auftraggeber. Schwarz eingefärbte Quality Gates betreffen nur die Entwicklung. Die letzteren beiden Typen von Quality Gates sind in Abhängigkeit von der Projektsituation vorhanden.

Eine Projektdurchführungsstrategie legt die Anordnung von Quality Gates fest, wobei es bei der iterativen Projektdurchführungsstrategie möglich ist, Quality Gates mehrfach und damit beliebig häufig durchlaufen zu können. An einem Quality Gate kann das Projekt ebenfalls in verschiedene Entwicklungsstränge verzweigen. Ebenso können verschiedene Entwicklungsstränge zusammengeführt werden.

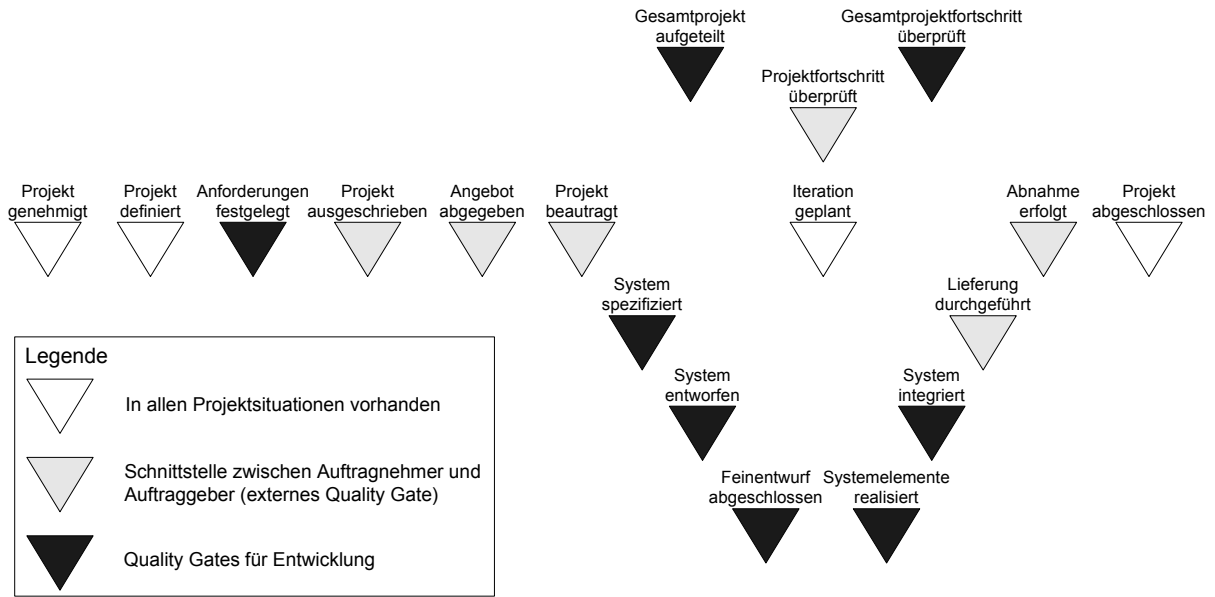


Abbildung 35: Quality Gates im V-Modell XT

Die Projektdurchführungsstrategie muss spätestens zum Quality Gate „Projekt definiert“ festliegen. Das Ergebnis der Anpassung wird im Projekthandbuch festgehalten, welches durch die Schlüsselrollen eines Projektes erstellt wird (vgl. Abbildung 36). Die Anpassung kann mit Hilfe des Softwarewerkzeugs „V-Modell XT Projektassistent“ [VMXT-PA '06] vorgenommen werden.

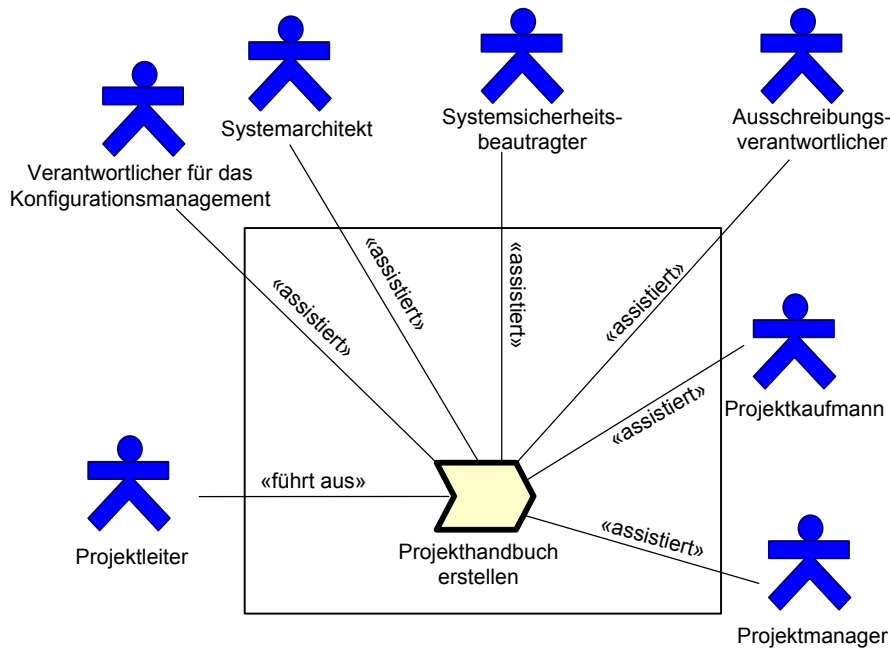


Abbildung 36: Beteiligte Rollen an der Aktivität „Projekthandbuch erstellen“

Das Projektmodell des V-Modells XT definiert neun Projektmerkmale mit verschiedenen Merkmalswerten. Für die Auswahl der Quality Gates und Projektdurchführungsstrategien sind allerdings nur fünf Merkmale relevant. Die verbleibenden Projektmerkmale werden für die Auswahl der so genannten Vorgehensbausteine des V-Modells XT benötigt. Allerdings haben letztere Projektmerkmale einen Einfluss auf die für das Quality Gate vorzulegenden Produkte.

Eines der Projektmerkmale des Projektmodells ist die Projektrolle. Die Projektmerkmalswerte des Projektmerkmals Projektrolle können grob unterschieden werden nach:

- **Auftraggeber:** Hierbei handelt es sich um ein Projekt, das sich im Wesentlichen mit der Ausschreibung, der Auswahl von Angeboten von potentiellen Auftragnehmern und der Abnahme des Produktes durch den Auftraggeber beschäftigt.
- **Auftragnehmer:** Hierbei handelt es sich um ein Projekt, das sich mit der Angebotserstellung und der Entwicklung (im Falle des Zustandekommens eines Vertrages) beschäftigt. Ferner muss das Produkt dem Auftraggeber geliefert und von diesem auch abgenommen werden.
- **Auftraggeber und Auftragnehmer:** Dieser Projekttyp ist dann zu wählen, wenn eine Auftrennung des Projektes in ein Auftraggeber- und ein Auftragnehmerprojekt nicht sinnvoll erscheint. Dies ist beispielsweise der Fall, wenn der Auftraggeber unternehmensintern ist.

Tabelle 20 fasst die für die Auswahl relevanten Projektmerkmale mit ihren möglichen Werten zusammen. Es ist erkennbar, dass nicht alle Projektmerkmale die Anpassung des Quality-Gate-Referenzprozesses beeinflussen.

Projektmerkmal	Relevante Merkmalswerte
Projektgegenstand*	Eingebettetes System, Komplexes System ¹ , Softwaresystem, Systemintegration
Projektrolle* (Rolle des Projektes im Zusammenspiel mit anderen Rollen)	Auftraggeber mit einem Auftragnehmer, Auftraggeber mit mehreren Auftragnehmern, Auftraggeber und Auftragnehmer mit Unterauftragnehmern, Auftraggeber und Auftragnehmer ohne Unterauftragnehmer, Auftragnehmer mit Unterauftragnehmern, Auftragnehmer ohne Unterauftragnehmer
Systemlebenszyklusausschnitt*	Entwicklung, Wartung und Pflege, Weiterentwicklung und Migration
Fertigprodukte*	ja, nein
Hohe Realisierungsrisiken*	ja, nein
Safety und Security**	ja, nein
Kaufmännisches Projektmanagement***	ja, nein
Quantitative Projektkennzahlen***	ja, nein
Benutzerschnittstelle***	ja, nein
Legende	
* Einfluss auf Auswahl der Quality Gates und auf einzureichenden Produkte	
** Nur Einfluss auf einzureichenden Produkte	
*** Nur Einfluss auf Auswahl der Vorgehensbausteine (für weitere Betrachtungen nicht relevant)	

Tabelle 20: Projektmerkmale und zugehörige Werte im V-Modell XT (Auswahl)

Das V-Modell XT nennt zehn relevante Projektdurchführungsstrategien. Ausgehend von der Projektsituation empfiehlt das V-Modell XT mindestens eine dieser Projektdurchführungsstrategien. Allerdings sind einige Projektmerkmale für bestimmte Projektdurchführungsstrategien irrelevant, so dass nicht jede Kombination von Wertebelegungen zwangsläufig zu einer neuen Projektdurchführungsstrategie führt. Abbildung 37 zeigt exemplarisch die agile Projektdurchführungsstrategie. Dabei wird lediglich die Anordnung der Quality Gates über den Kontrollfluss gezeigt und nicht die zwischen den Quality

¹ Ein komplexes System besteht aus Hard- und Softwarekomponenten und ist von größerer Komplexität (z.B. ein Militärflugzeug oder ein Automobil).

Gates liegenden Prozessschritte, durch die die Produkte entstehen. Diese Darstellung wird als Gate-Netzwerk bezeichnet und in Kapitel 6 weiter formalisiert.

Die agile Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer ergibt sich beispielsweise durch die folgende Projektsituation:

- **Projektgegenstand:** Softwaresystem
- **Projektrolle:** Auftraggeber und Auftragnehmer ohne Unterauftragnehmern
- **Systemlebenszyklusausschnitt:** Entwicklung
- **Fertigprodukte:** nein
- **Hohe Realisierungsrisiken:** ja

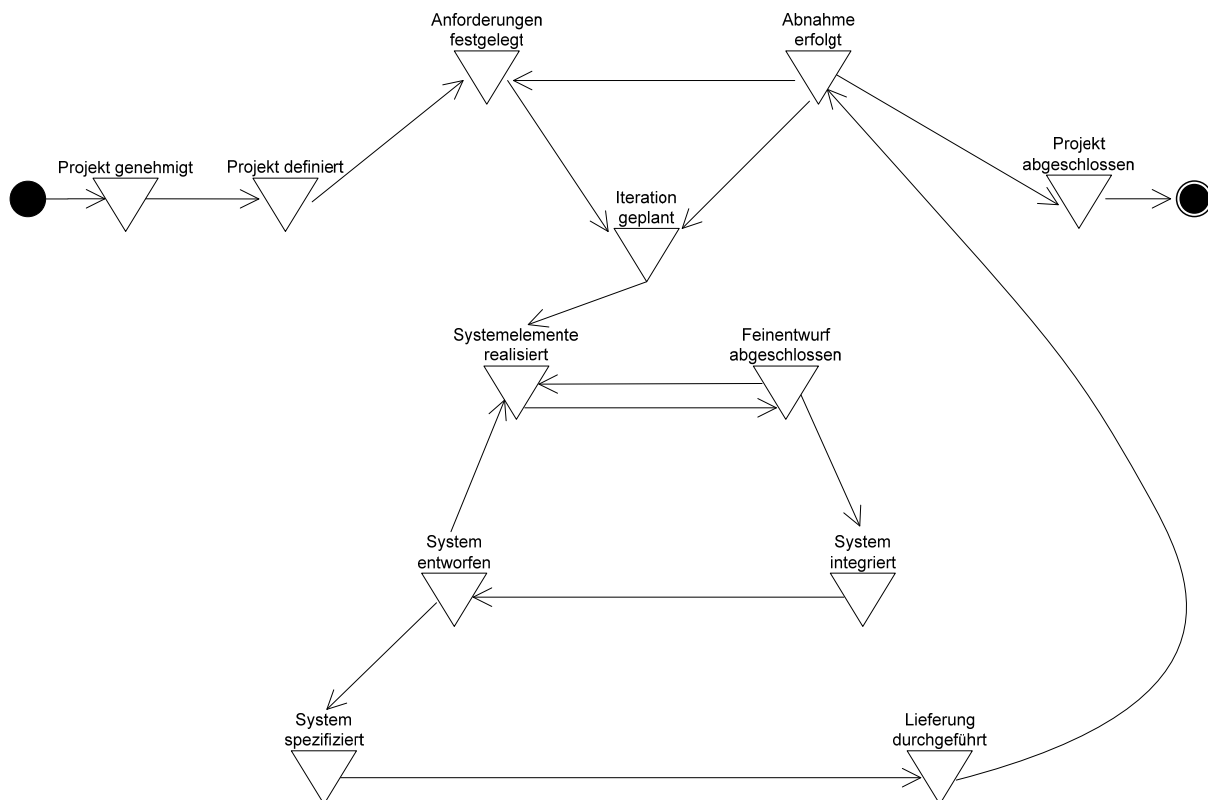


Abbildung 37: Agile Projektdurchführungsstrategie im V-Modell XT

Die agile Projektdurchführungsstrategie geht davon aus, dass Anforderungen instabil sind. Außerdem ist anfangs unklar, ob die Software realisiert werden kann. Es besteht bezüglich der Realisierung also ein hohes Risiko. Aus genannten Gründen müssen Anforderungen ständig neu festgelegt, Risiken bestimmt und in der Iterationsplanung geprüft werden. Im extremen Fall kann das Projekt im Quality Gate „Iteration geplant“ gestoppt werden. Um diese Risiken weitestgehend zu umgehen, setzt die agile Projektdurchführungsstrategie auf eine enge Kooperation zwischen Auftragnehmer und Auftraggeber. Durch häufig an den Auftraggeber erfolgende Lieferungen, erhält dieser stets ein lauffähiges System („Lieferung durchgeführt“) und kann durch seine Entscheidung im Quality Gate „Abnahme erfolgt“ steuernd in ein Projekt eingreifen.

4.6 Quality-Gate-Referenzprozess des Stage-Gate-Prozesses

Der Stage-Gate-Prozess nach Cooper [Cooper '01] ist ein Referenzprozess, der bei der Entwicklung verschiedener Produkte eingesetzt wird. Dazu gehört auch die Entwicklung von Software. Wie bereits in der empirischen Erhebung festgestellt, orientieren sich die meisten Quality-Gate-Referenzprozesse am Stage-Gate-Prozess. Ebenso stellt die US-amerikanische *Product Development and Management Association (PDMA)* 1997 fest, dass nahezu 60 % der US-amerikanischen Unternehmen eine Variante

des Stage-Gate-Prozesses verwenden [Griffin '97]. Vergleichbare Daten liegen für den europäischen Raum nicht vor.

4.6.1 Hintergrund und Überblick

Der Stage-Gate-Prozess deckt den gesamten Entstehungsprozess von der Bewertung einer Produktidee bis zur Markteinführung eines Produktes ab. Die Entwicklung stellt hierbei nur einen Teil dar. Das Ziel des Stage-Gate-Prozesses ist es, das Risiko des Scheiterns eines Projektes frühzeitig zu erkennen, um rechtzeitig gegenzusteuern. Ebenso sind die Chancen eines Projektes von Interesse.

Der Stage-Gate-Prozess untergliedert den Entstehungsprozess in konstruktive Phasen (*Stages*), sowie in *Gates*. Gates liegen zwischen den Phasen. Gates sind unfokussierte Entscheidungspunkte (vgl. Abbildung 1); der Fokus kann jedoch auf die Prüfung qualitätsorientierter Kriterien gelegt werden, so dass Quality Gates realisiert werden können. Phasen enthalten Aktivitäten, die abgeschlossen sein müssen, um von einer Phase des Prozesses zur nächsten Phase zu gelangen. Die Fortsetzungsentscheidung wird von Gatekeepern im Gate getroffen. Aufgrund des Erfüllungsgrads der definierten Kriterien wird das Risiko des Projektes innerhalb des Gate-Reviews sichtbar. Ressourcen werden freigegeben, falls das Risiko klein genug erscheint (vgl. Abbildung 38).

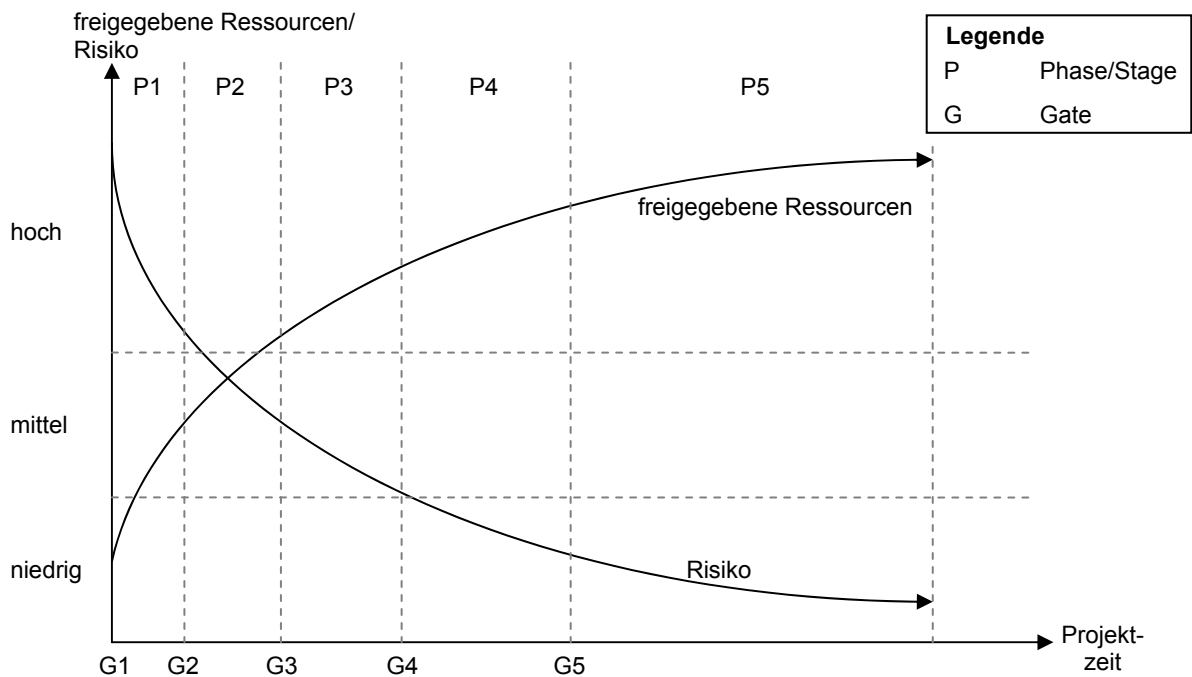


Abbildung 38: Entwicklung des Risikos und der resultierender Ressourcen (nach [TQU '05])

Abbildung 39 zeigt den klassischen Stage-Gate-Prozess. Viele Unternehmen haben jedoch schon eigene Varianten des Stage-Gate-Prozesses entwickelt, die sich vor allem in der Anzahl der Gates unterscheiden. So nutzt Motorola beispielsweise eine Variante des Stage-Gate-Prozesses mit 16 Gates [CGISS '02].

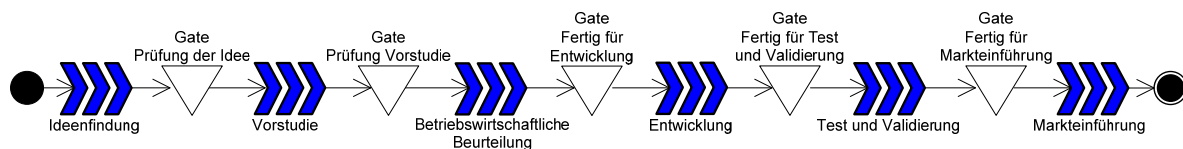


Abbildung 39: Klassischer Stage-Gate-Prozess nach Cooper

Da der Stage-Gate-Prozess eine Vielzahl von Projekttypen abdeckt, sind viele Vorgaben nur abstrakt formuliert. Cooper definiert ein Gate als speziellen Meilenstein. Nach Cooper werden im Rahmen von Meilensteinen – so wie sie im Projektmanagement genutzt werden – keine Entscheidungen über den Abbruch oder die Fortsetzung des Projektes getroffen. Vielmehr bilden die Meilensteine nach Cooper

das Mikromanagement innerhalb der Phasen, während durch die Gates der grobe Rahmen vorgeben wird. Hinsichtlich der Entscheidungen, die im Gate im Gegensatz zum Meilenstein getroffen werden können, gilt folgendes:

“Milestones are not gates per se; go/kill decisions are not made here...“, [Cooper '01]

Ein Hauptaugenmerk des Stage-Gate-Prozesses liegt im Projektportfoliomanagement. Der Stage-Gate-Prozess hilft dabei

- ein Neuentwicklungs-Projekt systematisch und effizient durchzuführen
- und unter konkurrierenden Projekten, die Erfolg versprechenden auszuwählen.

Mittlerweise lassen sich drei Generationen des Stage-Gate-Prozesses unterscheiden. Die erste Generation reicht bis in die 1960er Jahre zurück und wird als Phasen-Review-Prozess bezeichnet. Ein Review prüft dabei die Ergebnisse wichtiger Aktivitäten einer Phase. Der Fokus liegt auf der Prüfung der technischen Kriterien. Demnach liegen die Nachteile darin, dass weder die Marktsituation noch verfügbare oder fehlende Ressourcen in der Prüfung beachtet werden. Ebenso fehlt dem Management die Möglichkeit, steuernd und konstruktiv einzugreifen zu können.

Stage-Gate-Prozesse der zweiten Generation unterscheiden sich insofern, als dass sie die Möglichkeit zum steuernden und konstruktiven Eingriff ermöglichen. Neben technischen Kriterien werden auch Kriterien geprüft, die die Marktsituation und Produktionskapazitäten betreffen. Da insbesondere die ersten Phasen des Stage-Gate-Prozesses über den späteren Erfolg oder Misserfolg eines Projektes entscheiden, legen Stage-Gate-Prozesse der zweiten Generation einen besonderen Fokus auf diese Phasen und ihre abschließenden Gates. Ein Quality-Gate-Referenzprozess in Stage-Gate-Prozessen der zweiten Generation folgt der Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie*.

Inzwischen haben viele Unternehmen Stage-Gate-Prozesse der zweiten Generation etabliert und auf ihre Bedürfnisse hin verändert. Die Veränderungen gehen soweit, dass es gerechtfertigt ist, von Stage-Gate-Prozessen der dritten Generation zu sprechen [Cooper '98b, Cooper '98a]. Cooper fasst Erweiterungen und Anpassungen in den sechs F-Konzepten zusammen, die in den nachfolgenden sechs Abschnitten vorgestellt werden. Ein Quality-Gate-Referenzprozess in Stage-Gate-Prozessen der dritten Generation folgt der Strategie *Quality Gates als flexible Qualitätsstrategie*.

4.6.1.1 Flexibilitäts-Konzept

Der Stage-Gate-Prozess lässt es in Abhängigkeit vom Projektrisiko zu, Phasen oder Gates auszulassen, hinzuzufügen, zu teilen oder zu verschmelzen. Auf diese Weise können in einem Projekt mit geringerem Risiko beispielsweise Phasen verschmolzen und Gates ausgelassen werden, um das Projekt nicht unnötig zu verteuern. Dementsprechend kann ein Projekt mit höherem Risiko mehr Phasen und Gates aufweisen, um häufiger steuernd und konstruktiv auf das Projekt einwirken zu können. Üblicherweise wird man jedoch eine obere und untere Grenze definieren, die angibt welche und wie viele Phasen und Gates ein Projekt mindestens aufweisen muss und höchstens aufweisen darf.

Abbildung 40 zeigt einen verkürzten Stage-Gate-Prozess mit verschmolzenen Phasen und ausgelassenen Gates, der in einem Projekt mit geringem Risiko eingesetzt werden könnte.

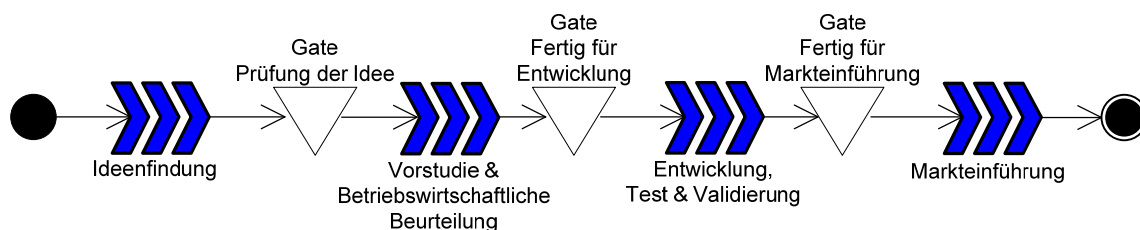


Abbildung 40: Verkürzter Stage-Gate-Prozess für ein Projekt mit geringem Risiko

4.6.1.2 Fuzzy-Konzept

Die Idee des Fuzzy-Konzeptes fußt auf der Grundidee der Fuzzy-Logik. Die Fuzzy-Logik verallgemeinert die klassische binäre Logik, um unscharfes Wissen auszudrücken. Innerhalb der Fuzzy-Logik existieren nicht nur die Werte *wahr* (= 1) und *falsch* (= 0), sondern auch beliebige Zwischenwerte im Intervall [0;1]. Damit ist es möglich, Aussagen wie „etwas“, „ziemlich“ oder „ein bisschen“ zu formalisieren.

Mit Hilfe der Fuzzy-Logik kann nun der Gesamt-Erfüllungsgrad aller Kriterien im Gate beschrieben werden. Basierend darauf lassen sich Schwellwerte definieren, die ausdrücken, wann welche Entscheidung von den Gatekeepern zu treffen ist. Auf diese Weise lassen sich Maßnahmen zwischen dem extremen Abbruch und der Fortsetzung des Projektes definieren. Zum Beispiel ist es möglich, dass beim Fehlen eines Ergebnisses ein Gate als „fast“ bestanden gilt, was zum Beispiel durch den Fuzzy-Wert 0,8 ausgedrückt werden kann. Ausgehend von diesem Wert lässt sich dann eine Entscheidung ableiten. Dies kann zum Beispiel eine bedingte Fortsetzung des Projektes sein, statt einer Wiederholung eines Gates. Die Prüfung der nicht erfüllten Kriterien kann dann nebenläufig zum Entwicklungsprozess erfolgen, um keine zu größeren Verzögerungen zu verursachen.

4.6.1.3 Fluss-Konzept

Das Fluss-Konzept besagt, dass Aktivitäten nicht mehr strikt zu einer Phase gehören müssen. Aktivitäten können in andere Phasen hineinragen, wobei der Intensitätsschwerpunkt einer Aktivität jedoch immer in einer Phase liegt. Phasen können sich damit überlappen, was entsprechend bei der Prüfung im Gate zu berücksichtigen ist, da die Erstellung einiger Produkte unter Umständen noch nicht komplett abgeschlossen ist. Eine Aktivität kann schon kurz vor Beginn einer Phase starten und erst in der nächsten beendet werden. Dies ist insbesondere bei länger andauernden Aktivitäten sinnvoll, um Verzögerungen am Ende einer Phase zu vermeiden. Unabhängig von diesen Änderungen, werden die Gate-Reviews wie gewohnt durchgeführt. Das heißt, dass Ressourcen nur dort freigegeben werden können und daher früher begonnene Aktivitäten wieder abgebrochen werden können. Zum Zeitpunkt des Gates müssen die Ergebnisse der zurückliegenden Phase die wichtigsten Kriterien jedoch erfüllen. Nicht erfüllte Kriterien können im Rahmen des Fuzzy-Konzeptes behandelt werden.

4.6.1.4 Fokus-Konzept

Das Fokus-Konzept besagt, dass die Ergebnisse auf Grundlage von Muss- und Soll-Kriterien bewertet werden. Dies hat den Vorteil, dass ungeeignete Projekte anhand der Muss-Kriterien aussortiert werden können. Anhand der Soll-Kriterien können Projekte priorisiert werden. Viel versprechende Projekte können dann fortgeführt werden, während die etwas schlechteren (die ja die Muss-Kriterien bestanden haben) zurückgestellt werden, weniger Ressourcen erhalten oder unter Auflagen weitergeführt werden. Abbildung 41 stellt den daraus resultierenden zweistufigen Entscheidungsprozess dar.

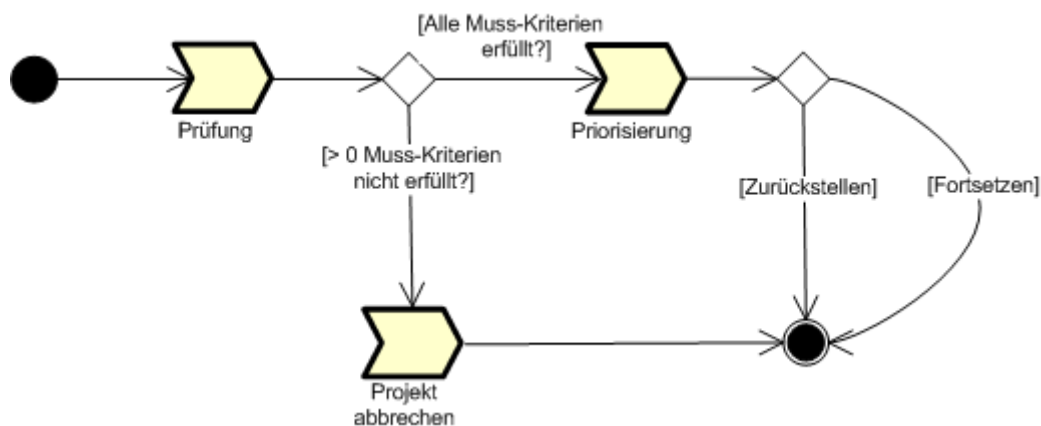


Abbildung 41: Zweistufiger Entscheidungsprozess innerhalb eines Gate-Reviews

4.6.1.5 Unterstützungs-Konzept

Um für den reibungslosen Ablauf eines etablierten Stage-Gate-Prozesses zu sorgen, ist es sinnvoll, ein Management zur Unterstützung zu institutionalisieren. Dieses Management kann Teil des Prozessmanagements sein und durch die Rolle eines Prozessmanagers für den Stage-Gate-Prozess besetzt werden. Mögliche Aufgaben liegen in der Unterstützung und der Überwachung des Ablaufs eines Gate-Reviews. Ebenso kann der Prozessmanager eine Empfehlung über den Fortgang des Projektes aussprechen oder entscheiden, falls es zu keiner Entscheidung durch die Gatekeeper gekommen ist. Letztlich ist der Prozessmanager für die Einleitung und Durchführung von Schulungsmaßnahmen für Gatekeeper, Entwickler und Gutachter zuständig.

4.6.1.6 Konzept der kontinuierlichen Verbesserung

Ebenso wie andere Prozesse, sollte der Stage-Gate-Prozess einer kontinuierlichen Verbesserung unterliegen. Dies kann beispielsweise durch die Anwendung des *Plan-Do-Check-Act-Kreislaufs (PDCA)* nach Deming [Walton '86] oder des *Quality Improvement Paradigmas (QIP)* im Rahmen einer *Experience Factory* nach Basili [Basili '93] geschehen. Typische Verbesserungsmaßnahmen umfassen die Etablierung der fünf vorgenannten F-Konzepte sowie weiterer Punkte:

- Einführung einer Workflow-Software zur verteilten Durchführung eines Gate-Reviews, falls Gatekeeper und Entwickler sich an verschiedenen Orten befinden.
- Intensivierung der vorderen Gates, um frühzeitig Informationen über die Erfolgsaussichten eines Projektes zu erhalten.
- Etablierung von Aktivitäten in den vorderen Phasen (und Prüfung der Ergebnisse dieser im Gate), die sich mit der Suche nach Fertigprodukten beschäftigen, um den späteren Entwicklungsaufwand und das Risiko zu minimieren.

4.6.2 Verwendung von Gates im Stage-Gate-Prozess

Nach Cooper weist der typische Stage-Gate-Prozess vier bis sechs Phasen und ebenso viele Gates auf. Jede Phase umfasst parallele und sequenzielle Aktivitäten, die Personen verschiedener Fachabteilungen involvieren. Vor jeder Phase (mit Ausnahme der Ideenfindung) befindet sich ein Gate, das den Übergang in die nächste Phase regelt. Im Rahmen eines Gates wird folgendes geprüft:

- **Prüfung der Qualität:** Liegen alle Ergebnisse der zurückliegenden Phase in ausreichender Qualität vor?
- **Prüfung der Attraktivität:** Ist das Projekt hinsichtlich benötigter Ressourcen und der Marktattraktivität interessant?
- **Prüfung des weiteren Vorgehens:** Sind das weitere Vorgehen, sowie die dafür notwendigen Ressourcen angemessen?

Jedes Gate fordert Ergebnisse, die anhand von Kriterien geprüft werden. Basierend darauf wird durch die Gatekeeper eine Entscheidung gefällt. Die folgende Auflistung geht näher auf Ergebnisse, Kriterien und die Entscheidung ein:

- *Ergebnisse* werden durch den Projektleiter oder auch durch die Ersteller für jedes Gate eingereicht. Eingereicht werden können nur elektronisch oder physisch vorhandene Dokumente. Die notwendigen Ergebnisse werden im vorherigen Gate bestimmt.
- *Kriterien* werden unterteilt in Muss- und Soll-Kriterien (vgl. Abschnitt 4.6.1.3). Kriterien werden in einer Checkliste festgehalten und sind sichtbar für alle Projektteilnehmer. Muss-Kriterien sind idealerweise auf einer Nominalskala mit den Werten ja und nein zu bewerten. Soll-Kriterien sind hingegen auf einer Ordinalskala zu bewerten, die den Vergleich (und damit eine Priorisierung) zwischen verschiedenen Projekten ermöglicht.
- Die *Entscheidung* besteht aus drei Komponenten:

1. der eigentlichen Entscheidung über die Fortsetzung, den Abbruch, die Zurückstellung des Projektes oder über die wiederholte Durchführung eines Gates,
2. Einem Freigabeplan für die Ressourcen,
3. Einer Liste der geforderten Ergebnisse, die im nächsten Gate vorzulegen sind. Darüber hinaus ist ein Termin für das nächste Gate zu setzen.

Tabelle 21 gibt eine kurze Auflistung typischer Prüfinhalte der einzelnen Gates eines Stage-Gate-Prozesses. Grundlage der Auflistung ist ein Stage-Gate-Prozess der zweiten Generation (vgl. Abbildung 39).

Gate	Typische Prüfinhalte
Gate <i>Prüfung der Idee</i>	<ul style="list-style-type: none"> - Entsprechung des Produktes mit der strategischen Ausrichtung des Unternehmens - Erfolgsaussicht z.B. unter Berücksichtigung technischer Möglichkeiten - Gewinnaussichten
Gate <i>Prüfung der Vorstudie</i>	<ul style="list-style-type: none"> - Strengere Prüfung der Kriterien des ersten Gates - Prüfung der Nutzerakzeptanz - Prüfung der benötigten Ressourcen
Gate <i>Fertig für Entwicklung</i>	<ul style="list-style-type: none"> - Prüfung des Projektplanes - Prüfung der Produktspezifikation
Gate <i>Fertig für Test und Validierung</i>	<ul style="list-style-type: none"> - Prüfung der benötigten finanziellen Ressourcen - Überprüfung, ob Marktakzeptanz unverändert
Gate <i>Fertig für Markteinführung</i>	<ul style="list-style-type: none"> - Prüfung der Verifikations- und Testaktivitäten - Prüfung des Return on Investment (ROI) - Prüfung der Marketingaktivitäten

Tabelle 21: Übersicht über Gates und typischer Prüfaspekte des Stage-Gate-Prozesses

4.6.3 Ausgestaltung effektiver und effizienter Gates nach Cooper

Cooper schlägt verschiedene Maßnahmen zur Ausgestaltung effektiver und effizienter Gates vor. Diese Maßnahmen werden in diesem Abschnitt vorgestellt.

Abbildung 38 zeigt, dass im Rahmen eines Gates mehr Ressourcen freigegeben werden, als bei vorherigen Gates. Folglich stellen spätere Gates höhere Anforderungen an das Urteilsvermögen der Gatekeeper, da bei einer Fehlentscheidung mehr Ressourcen fehlgeleitet würden. Aus dem gleichen Grund sind härtere Kriterien zu bestimmen. Zu bedenken ist weiterhin, dass Gatekeeper späterer Gates entsprechende Entscheidungsbefugnis besitzen müssen, da sie unter Umständen ein Projekt abrechnen müssen, für das bereits erhebliche Ressourcen aufgewendet worden sind.

Die Auswahl der Kriterien und die Intensität eines Gate-Reviews sind durch das Management sorgfältig zu bestimmen, um zu verhindern, dass zu viele aussichtsreiche Projekte abgebrochen und zu viele ungeeignete Projekte fortgesetzt werden. Das erste Problem tritt vor allem in den frühen Gates auf, in denen meist nur Visionen vorhanden sind, jedoch kaum Konzepte zur Realisierung. Bei dieser Ausgangslage können zu harte Kriterien oder eine zu rigorose Prüfung zum Abbruch eines viel versprechenden Projektes führen. Durch die Umsetzung des Fokus-Konzepts und des Fuzzy-Konzeptes gemäß Cooper kann dieses Problem abgeschwächt werden. Das Fokus-Konzept hilft dabei, die richtigen Projekte durch Muss-Kriterien zu filtern und durch Soll-Kriterien zu priorisieren. Das Fuzzy-Konzept leitet dann aus dem Erfüllungsgrad dieser Kriterien eine Entscheidung ab, die nicht zwangsläufig zum

Projektabbruch bei Nichterfüllung der Kriterien führen muss, sondern beispielsweise nur zur Zurückstellung eines Projektes führt.

Methoden zur Entscheidungsfindung sind so zu konzipieren, dass sie wiederholbar und effizient zu gleichen Ergebnissen führen. Ersteres kann zum Beispiel durch die Forderung nach bestimmten, für alle Projekte gleichen Ergebnissen realisiert werden, letzteres durch die Verdichtung auf Kennzahlen.

Eine effiziente und effektive Gestaltung stellt besondere Anforderungen an die Rolle des Gatekeepers. Ein Gatekeeper besitzt auf jeden Fall Entscheidungsbefugnis über die Freigabe von Ressourcen. Jedes Gate erfordert Gatekeeper mit den entsprechenden Fachkenntnissen. Da ein Gate als Synchronisationspunkt verschiedener parallel verlaufender Aktivitätsstränge der Teilprojekte der verschiedenen Fachabteilungen wirkt, reicht ein Gatekeeper in der Regel nicht aus. Vielmehr sollte ein Gatekeeper jeder Fachabteilung in jedem Gate-Review vertreten sein.

Mit zunehmender Verantwortung aufgrund steigender notwendiger Ressourcen, sollten spätere Gates mit einer größeren Anzahl erfahrener Gatekeeper besetzt werden. Eine ähnliche Anforderung gilt für wichtigere Projekte, da sie grundsätzlich erfahrene Gatekeeper benötigen, um eine möglicherweise teure Fehlentscheidung zu vermeiden.

Cooper schlägt weiterhin vor, dass Gatekeeper von Gate zu Gate innerhalb eines Projekts möglichst durch die gleichen Personen besetzt werden sollten, um eine gewisse Kontinuität der Entscheidungen zu wahren. Noch effektiver ist die Etablierung einer Gruppe von Gatekeepern aus den betroffenen Fachbereichen, die für alle Projekte immer gleich besetzt ist, um auf diese Weise Kontinuität projektübergreifend zu sichern.

4.7 Bezug zwischen Prozessreifegradmodellen und Quality-Gate-Referenzprozessen

Die Prozessreifegradmodelle *CMMI* (Capability Maturity Model Integration) [CMMI '06] und *SPICE* (Software Process Improvement and Capability Determination)² [Hörmann et al. '06] dienen der Beurteilung und Verbesserung der Reife von Software-Entwicklungsprozessen. Prozessreifegradmodelle geben vor, welche Charakteristika ein Software-Entwicklungsprozess aufweisen muss, um eine bestimmte Reife erlangen zu können. Die Art und Weise der Umsetzung wird hingegen nicht vorgegeben. Die Ausgangshypothese ist dabei, dass Unternehmen mit reiferen Prozessen, Projekte mit erheblich höherer Wahrscheinlichkeit zum Erfolg führen können. Innerhalb dieses Abschnittes wird der Bezug von Quality-Gate-Referenzprozessen zu den Prozessreifegradmodellen CMMI und SPICE aufgezeigt. Dadurch, dass Prozessreifegradmodelle nur vorgeben, was ein reifer Prozess aufweisen muss, nennt keines der beiden betrachteten Prozessreifegradmodelle explizit Quality-Gate-Referenzprozesse als Lösungsmöglichkeit.

Ein Bezug zwischen Quality Gates und einem Prozessreifegradmodell ist von genau einer der zwei folgenden Arten:

- Das Prozessteilgebiet des Prozessreifegradmodells kann mit Hilfe von Quality Gates (teilweise) umgesetzt werden.
- Das Prozessteilgebiet des Prozessreifegradmodells unterstützt die Etablierung von Quality Gates.

Einfacher gesprochen bedeutet dies erstens, dass die Einführung eines Quality-Gate-Referenzprozesses einem Unternehmen ermöglicht, einen reiferen Prozess zu erhalten und zweitens, dass ein reiferer Prozess die Etablierung eines Quality-Gate-Referenzprozesses erleichtert.

² Auch ISO/IEC 15504.

4.7.1 Bezug zwischen CMMI und Quality-Gate-Referenzprozessen

Das Prozessreifegradmodell CMMI definiert verschiedene Prozessteilgebiete (sog. *Process Areas*), in denen sich ein Prozess verbessern kann. Für jedes dieser Prozessteilgebiete sind spezielle und generische Ziele (sog. *Specific* und *Generic Goals*) definiert, die erfüllt sein müssen, damit das Prozessteilgebiet als umgesetzt gilt. Spezielle Ziele gelten nur für ein Prozessteilgebiet, während generische Ziele für mehrere Prozessteilgebiete gelten. Für die Erfüllung der Ziele wird die Etablierung bestimmter spezieller und generischer Praktiken (sog. *Specific* und *Generic Practices*) erwartet.

Ein Unternehmen kann nun eine in CMMI definierte Teilmenge der Prozessteilgebiete verbessern. Der Entwicklungsprozess erreicht auf diese Weise eine höhere Reifestufe (ein sog. *Maturity Level*). Alternativ kann ein Prozessteilgebiet einzeln verbessert werden und der Entwicklungsprozess dort eine bestimmte Fähigkeitsstufe (einen sog. *Capability Level*) erlangen. Insgesamt existieren fünf Reifestufen und sechs Fähigkeitsstufen. Höhere Reifestufen weisen auf einen reiferen Prozess hin. CMMI definiert insgesamt 22 Prozessteilgebiete, die sich entweder auf das Prozess- oder Projektmanagement, den Entwicklungsprozess oder auf unterstützende Prozesse beziehen.

Tabelle 22 zeigt Bezüge von Quality-Gate-Referenzprozessen zu einzelnen Prozessteilgebieten des Prozessreifegradmodells CMMI. In der Spalte „Bezug“ wird in eckigen Klammern jeweils der Bezug charakterisiert. Nachfolgend wird eine detaillierte Beschreibung gegeben.

Prozessteilgebiet (Reifestufe)	Bezug
Project Monitoring and Control (2)	<p>[Prozessteilgebiet kann durch Quality Gates teilweise umgesetzt werden] Überwachung des Projektfortschritts, um ggf. steuernd einzugreifen.</p> <p>Wichtige generische Praktiken (Generisches Ziel „Managed“):</p> <ul style="list-style-type: none"> - Identifizierung von Stakeholdern für Vereinbarungen und Steuerung. (GP 2.7) - Status des Projektes für das Management sichtbar an definierten Punkten, damit dieses steuernd eingreifen kann. (GP 2.10)
Measurement and Analysis (2)	<p>[Unterstützt die Etablierung von Quality Gates] Etablierung einer systematischen Möglichkeit zur objektiven Messung und Analyse. Unterstützt die Prozessteilgebiete „Quantitative Project Management“ und „Project Monitoring and Control“.</p> <p>Spezielle Ziele:</p> <ul style="list-style-type: none"> - Zielgerichtetes Messen (SG 1) - Durchführung der Messung (SG 2)
Decision Analysis and Resolution (3)	<p>[Prozessteilgebiet kann durch Quality Gates teilweise umgesetzt werden] Etablierung eines Prozesses zur Entscheidungsfindung aufgrund formaler Kriterien, um die Subjektivität von Entscheidungen zu minimieren. Anwendbar auf technische und nichttechnische Kriterien. Häufig angewendet, um sich für eine Alternative zu entscheiden (z.B. Make-or-Buy-Entscheidungen).</p> <p>Wichtige spezielle Praktiken (Spezielles Ziel „Evaluate Alternatives“):</p> <ul style="list-style-type: none"> - Erstellung von Kriterien. (SP 1.2) - Identifikation alternativer Lösungen. (SP 1.3)
Risk Management (3)	<p>[Prozessteilgebiet kann durch Quality Gates teilweise umgesetzt werden] Identifikation von Risiken und Einleitung von Gegenmaßnahmen, um die Eintretenswahrscheinlichkeit oder den Schaden zu verringern.</p> <p>Wichtige spezielle Praktiken (Spezielles Ziel „Prepare for Risk Management“):</p> <ul style="list-style-type: none"> - Etablierung einer Risikostrategie (SP 1.3) <p>Wichtige spezielle Praktiken (Spezielles Ziel „Identify and Analyze Risks“):</p> <ul style="list-style-type: none"> - Identifikation von Risiken (SP 2.1) - Bewertung und Priorisierung der Risiken (SP 2.2) <p>Wichtige spezielle Praktiken (Spezielles Ziel „Mitigate Risks“):</p> <ul style="list-style-type: none"> - Bestimmung und Einleitung von Gegenmaßnahmen (SP 3.1)
Quantitative Project Management (4)	<p>[Prozessteilgebiet kann durch Quality Gates umgesetzt werden] Überwachung der Qualität eines Produktes und von Prozesskennzahlen, um ggf. steuernd eingreifen zu können.</p> <p>Wichtige generische Praktiken (Generisches Ziel „Quantitatively managed process“):</p> <ul style="list-style-type: none"> - Qualitätsorientierte Kriterien systematisch ermittelt, z.B. auf Grundlage statistisch ermittelter Sollwerte, aus Kundenwünschen oder Geschäftszielen. (GP 4.1).

Tabelle 22: Bezug von Quality Gates zu den Prozessteilgebieten von CMMI

4.7.2 Bezug zwischen SPICE und Quality-Gate-Referenzprozessen

Das Prozessreifegradmodell SPICE definiert – ähnlich zu CMMI – verschiedene Prozessteilgebiete (sog. *Processes*), die zu verschiedenen Prozessgruppen gehören (sog. *Process Groups*). Ein Prozess kann in Prozessteilgebieten verbessert werden. Durch die Erfüllung bestimmter Prozessattribute (sog. *Process Attributes*) kann der Prozess eines Unternehmens eine höhere Reifestufe erlangen (ein sog. *Capability Level*). Die Prozessattribute verlangen die Umsetzung verschiedener Prozessteilgebiete. Für jedes Prozessteilgebiet werden Basispraktiken und generische Praktiken (sog. *Base Practices* und *Generic Practices*) empfohlen, die die Umsetzung erleichtern, aber nicht zwingend erfüllt sein müssen.

Ähnlich zu CMMI kann ein Unternehmen bestimmte Prozessattribute und damit eine definierte Menge von Prozessteilgebieten erfüllen und somit eine höhere Reifestufe erlangen. Dabei sind sechs verschiedene Reifestufen möglich, wobei höhere Reifestufen einen reiferen Prozess signalisieren. Alternativ können auch einzelne Prozessteilgebiete erfüllt werden.

Tabelle 23 zeigt Bezüge von Quality-Gate-Referenzprozessen zu einzelnen Prozessteilgebieten des Prozessreifegradmodells SPICE. In der Spalte „Bezug“ wird in eckigen Klammern jeweils der Bezug charakterisiert. Nachfolgend wird eine detaillierte Beschreibung gegeben.

Prozesteilgebiet (Reifestufe) [Prozessgruppe]	Bezug
Joint Review (2) [Supporting Life Cycle Processes]	[Prozesteilgebiet kann durch Quality Gates teilweise umgesetzt werden] - Durchführung von Reviews unter Beteiligung des Managements.
Quality Management (4) [Management Pro- cess Group]	[Prozesteilgebiet kann durch Quality Gates teilweise umgesetzt werden] - Projektübergreifende Sicherung der Software-Qualität.
Project Management (2 und 3) [Management Pro- cess Group]	[Prozesteilgebiet kann durch Quality Gates teilweise umgesetzt werden] - Überwachung des Projektfortschritts auf verschiedenen Ebenen unter Be- teiligung des Managements.
Risk Management (1 und 4) [Management Pro- cess Group]	[Prozesteilgebiet kann durch Quality Gates teilweise umgesetzt werden] Systematisches Management von Risiken. Wichtige Basispraktiken: - Identifikation von Risiken (BP 3) - Bewertung von Risiken (BP 4) - Bestimmung und Einleitung von Gegenmaßnahmen (BP 5)
Measurement (3 und 5) [Management Pro- cess Group]	[Unterstützt die Etablierung von Quality Gates] Objektive Bewertung der Software-Qualität zur Unterstützung des Mana- gements. Wichtige Basispraktiken: - Ermittlung des Informationsbedarfs (BP 3) - Bestimmung von Maßen auf Grundlage des Informationsbedarfs (BP 4) - Durchführung der Messung (BP 5) - Analyse der Messwerte (BP 6) - Nutzung der Analyse für Entscheidungen (BP 7)

Tabelle 23: Bezug von Quality Gates zu den Prozesteilgebieten von SPICE

4.7.3 Fazit der Untersuchung von Prozessreifegradmodellen

Die Untersuchung der Prozessreifegradmodelle CMMI und SPICE zeigt, dass ein Unternehmen mit der Einführung von Quality Gates bereits verschiedene geforderte Prozesteilgebiete (zumindest teilweise) erfüllen kann. Dabei können die Projektkontrolle und Projektsteuerung, das Risikomanagement, die Entscheidungsfindung und die projektübergreifende Qualitätssicherung (also das Qualitätsmanagement) anhand (formaler) Kriterien durch Quality Gates (teilweise) umgesetzt werden. Gleichzeitig kann durch die Unterstützung und Systematisierung von Messprogrammen, die gleichermaßen durch CMMI und SPICE nahegelegt werden, die Etablierung von Quality Gates erleichtert werden.

4.8 Abschließende Betrachtung

Im Folgenden werden die Betrachtungen dieses Kapitels hinsichtlich des Quality-Gate-Konzeptes zusammengefasst. Dazu gehört eine Zusammenfassung der Charakteristika von Quality Gates, eine

Abgrenzung des Konzepts von den Konzepten des Meilensteins und des Reviews, sowie eine Zusammenfassung der wichtigsten Problempunkte.

4.8.1 Charakteristika von Quality Gates

Die folgende Auflistung fasst die Charakteristika von Quality Gates zusammen.

- **Spezieller Meilenstein:** Quality Gates sind spezielle Meilensteine, die zwischen den herausragenden Prozessschritten (fast immer Phasen) innerhalb eines Prozesses liegen. Zwischen diesen Prozessschritten liegt zumeist ein Disziplinbruch vor oder es handelt sich um besonders kritische Stellen im Prozess. Durch die Zusammenführung verschiedener Aktivitäten zum Phasenende kann ein Quality Gate synchronisierend wirken. Im Falle eines Disziplinbruchs kann ein Quality Gate die Funktion einer Schnittstelle übernehmen, da durch ein Quality Gate Ein- oder Ausgangskriterien vorgegeben werden können. Als spezieller Hauptmeilenstein sind Quality Gates nahezu unabhängig von der Projektsituation.
- **Ergebnisorientierung:** Ein Quality Gate ist ergebnisorientiert, da es sich an den Prozessschritten orientiert und deren Ergebnisse im Gate-Review geprüft werden. Ereignisorientierte Quality Gates, also solche Quality Gates, die in äquidistanten zeitlichen Abständen durchgeführt werden, sind **keine** Quality Gates im Verständnis dieser Arbeit.
- **Vereinbarung im Referenzprozess:** Quality Gates werden bereits im Referenzprozess vereinbart. Eine Anpassung an eine Projektsituation findet auf Grundlage mehr oder weniger definierter Algorithmen oder Heuristiken statt. Damit liegen mindestens die Anordnung, Anzahl und Auswahl der Quality Gates zum Ende des Projektstarts fest, insbesondere also vor der spezifischen Planungsphase. Kriterien und Ergebnisse können bereits zu diesem Zeitpunkt definiert sein, können jedoch auch später ermittelt werden. Idealerweise stehen sie jedoch vor dem jeweiligen Gate-Review fest. Die Kriterien sind qualitätsorientiert. Sie leiten sich daher aus den Qualitätszielen eines Projektes ab. Nicht qualitätsorientierte Kriterien (beispielsweise wirtschaftliche Kriterien) sind möglich und auch üblich, lassen ein Quality Gate jedoch bei übermäßiger Nutzung zum unfokussierten Entscheidungspunkt ausarten.
- **Verwendung als Fortschrittsmaß:** Durch die Ergebnisorientierung und die Ausrichtung an den herausragenden Prozessschritten, kann festgestellt werden, ob ein wesentlicher Stand innerhalb eines Projektes erreicht wurde. Bei Verfolgung der grundlegenden Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* kann der Projektfortschritt auch mit anderen Projekten verglichen werden.
- **Freigabepunkt im Projekt:** An einem Quality Gate (als speziellem Entscheidungspunkt) wird eine formale Entscheidung auf Basis der Kriterien über die Fortsetzung des Projektes getroffen. Ressourcen werden formal freigeschaltet. Gegebenfalls sind Maßnahmen zu treffen. Die Mitwirkung von Entscheidungsträgern (Gatekeepern) ist daher notwendig. Die Option zur Eskalation zum Lenkungsausschuss bzw. zur Geschäftsleitung ist ausdrücklich vorgesehen.

4.8.2 Einordnung als Meilenstein-Konzept

Quality Gates können als spezielle Meilensteine betrachtet werden. Die folgende Auflistung gibt Aufschluss über Unterschiede und Gemeinsamkeiten zum projektbezogenen Meilenstein.

- **Zeitpunkt der Einplanung:** Anordnung, Anzahl und Auswahl der Quality Gates liegen bereits am Ende des Projektstarts fest. Projektbezogene Meilensteine werden erst in der Planungsphase ermittelt – sie können daher viel genauer und spezifischer an die Projektsituation angepasst werden. Projektbezogene Meilensteine können aus diesem Grund nicht schon in einem Referenzprozess vereinbart werden.
- **Lage im Projekt:** Quality Gates werden an besonders kritischen Stellen eingeplant. Projektbezogene Meilensteine orientieren sich an den Ergebnissen der Arbeitspakete.
- **Auftretenshäufigkeit:** Quality Gates sind spezielle Hauptmeilensteine in einem Projekt. Sie treten somit erheblich seltener auf als projektbezogene Meilensteine.

- **Abstraktheit:** Das Ziel eines Quality Gates, also der Projektstand, der nach erfolgreichem Durchschreiten erreicht wird, ist häufig abstrakt formuliert. Projektbezogene Meilensteine geben konkretere und viel projektspezifischere Ziele vor.
- **Entscheidung:** Die Option zur Eskalation zum Lenkungs Ausschuss bzw. zur Geschäftsleitung bei Quality Gates ist ausdrücklich vorgesehen. Die Möglichkeit zur Eskalation ist bei projektbezogenen Meilensteinen weniger institutionalisiert. Der Abbruch eines Projektes zu einem Meilenstein ist daher unüblich.
- **Freigabe von Ressourcen:** Quality Gates und projektbezogene Meilensteine sind gleichermaßen Zeitpunkte, an denen formal Ressourcen freigegeben werden.
- **Risikomanagement:** Zu Meilensteinen und Quality Gates können gleichermaßen Gegenmaßnahmen beschlossen werden. Sie wirken daher risikominimierend.

Abbildung 42 fasst das beschriebene Verhältnis zum projektbezogenen Meilenstein zusammen und stellt darüber hinaus das Verhältnis zu verschiedenen anderen Meilensteinkonzepten dar.

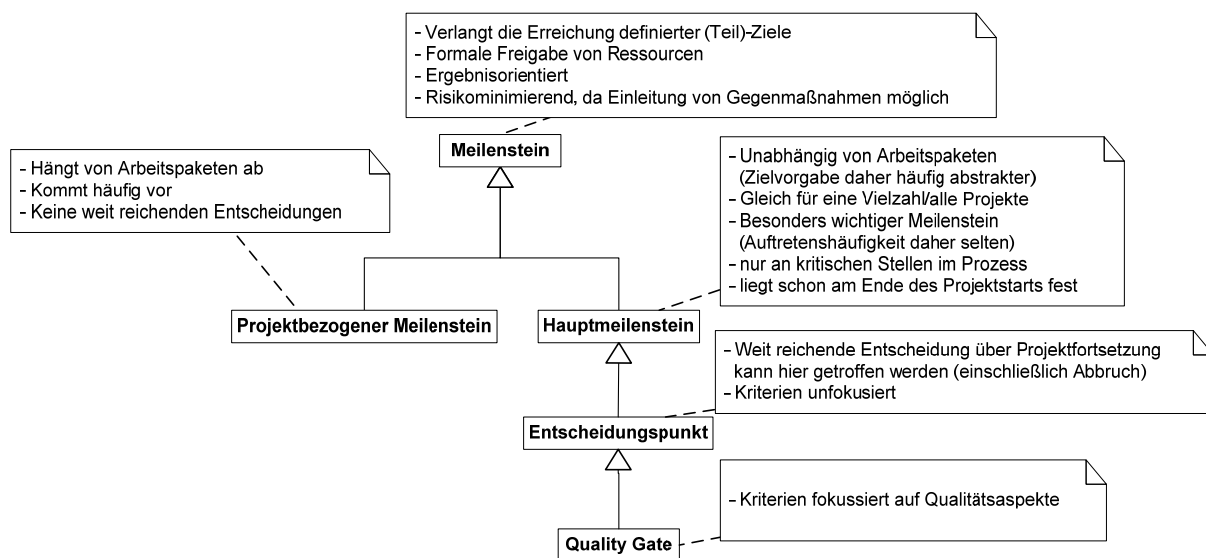


Abbildung 42: Ein Quality Gate als spezieller Meilenstein

4.8.3 Einordnung als Review-Konzept

Das Gate-Review eines Quality Gates lässt sich als spezielles Review einordnen (vgl. Abbildung 43). Im Folgenden werden das technische Review und das Gate-Review auf Unterschiede und Gemeinsamkeiten hin untersucht.

- **Strukturiertheit des Ablaufs:** Im Vergleich zum formalen Vorgehen bei der Inspektion, ist ein Gate-Review (als spezielles Management-Review) weniger strukturiert.
- **Einordnung eines Gate-Reviews als Projektbesprechung:** Ein Gate-Review als Projektbesprechung ist wegen seiner Möglichkeit zur Eskalation und aufgrund seiner Häufigkeit zwischen dem Meilenstein-Review und der Besprechung des Lenkungs Ausschusses einzuordnen (vgl. Tabelle 3).
- **Inhaltliche Ausrichtung der Prüfung:** Innerhalb eines technischen Reviews werden Dokumente gelesen und inhaltlich nachvollzogen. Innerhalb eines Gate-Reviews werden (Teil-)Produkte und Kennzahlwerte geprüft.
- **Abstraktionsgrad der Ergebnisse:** Die Prüfung im Gate-Review erfolgt auf Grundlage von Kennzahlwerten, die die Ausgangssituation des Projektes, Prozesses oder Produktes abstrahiert widerspiegeln. Die Prüfgegenstände innerhalb eines technischen Reviews sind unaufbereitete – oder zumindest nur minimal – aufbereitete Dokumente des Projektes.

- **Formalität der Kriterien:** Innerhalb eines Gate-Reviews wird geprüft, ob Kennzahlwerte bestimmte Sollwerte einhalten. Die Prüfung ist dabei sehr formal, es gibt zumeist nur wenige Möglichkeiten zur Interpretation. Die Prüfung innerhalb eines Reviews ist stark inhaltlich. Befunde im technischen Review sind häufig zu diskutieren.
- **Entscheidung:** Die Entscheidung, die innerhalb eines technischen Reviews getroffen wird, legt im schlimmsten Fall die Überarbeitung der inspizierten Dokumente nahe. Im Gate-Review wird über die Fortsetzung eines Projektes entschieden. Im ungünstigsten Fall ist der Abbruch des Projektes möglich.
- **Auswirkung der Kriterien:** Die Kriterien eines Quality Gates wirken analytisch, jedoch gleichzeitig auch steuernd, da die Kriterien für alle sichtbar sind. Die Kriterien eines technischen Reviews sind weniger sichtbar für die Lieferanten und sehr viel stärker interpretierbar – sie wirken daher häufig nur analytisch.

Abbildung 43 fasst das beschriebene Verhältnis zum technischen Meilenstein zusammen und zeigt darüber hinaus das Verhältnis zum Management-Review.

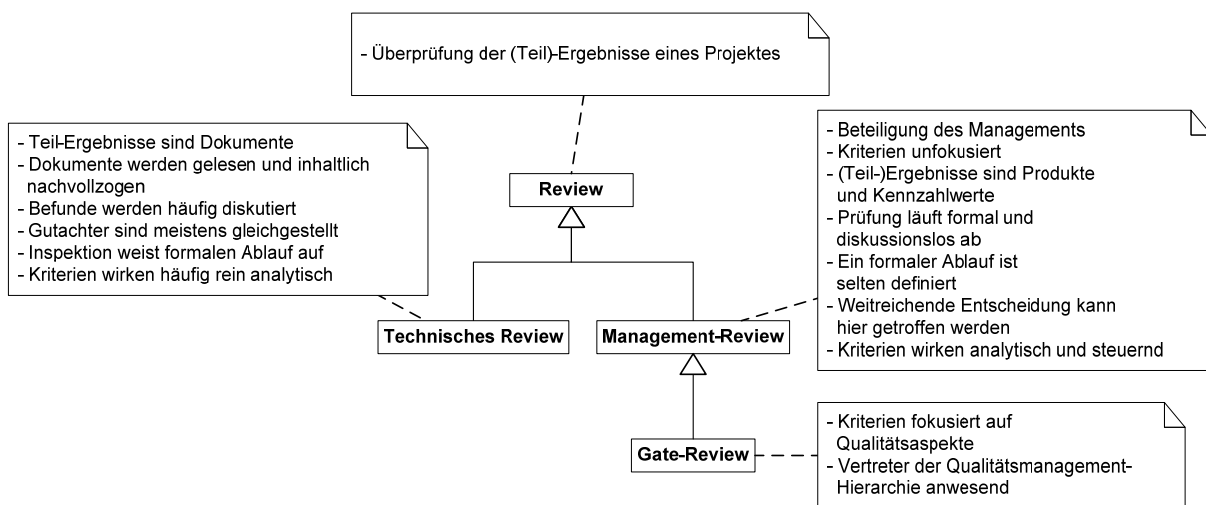


Abbildung 43: Ein Quality Gate als spezielles Review

4.8.4 Weitere identifizierte Problempunkte

Neben den in Abschnitt 4.3.3.10 identifizierten Problempunkten, konnten weitere Problempunkte in den betrachteten Quality-Gate-Referenzprozessen identifiziert werden:

- Die meisten Quality-Gate-Referenzprozesse verfügen über keine formalen Tailoring-Regeln (Ausnahme V-Modell XT), die die Anpassung an eine Projektsituation erleichtert.
- Abläufe, die der Entscheidungsfindung und zur Herleitung von notwendigen Maßnahmen dienen, fehlen häufig oder sind nur abstrakt beschrieben.
- Abläufe, die der Erstellung von Kriterien dienen, fehlen meistens. Ausnahme ist hier der Quality-Gate-Referenzprozess nach Pfeifer.
- Auf inkrementelles, iteratives und agiles Vorgehen wird nur wenig eingegangen (Ausnahme V-Modell XT).

5 Abstrahiertes Domänenmodell für Quality-Gate-Referenzprozesse

Im vorherigen Kapitel wurden Quality-Gate-Referenzprozesse untersucht. Dabei wurden verschiedene Teilkonzepte genannt, die mit Quality Gates in Beziehung stehen. In dieser Arbeit stellt ein Konzept gemäß [Wikipedia '08] eine gedankliche Zusammenfassung von Sachverhalten dar, die sich durch gemeinsame Merkmale auszeichnen. Ein Teilkonzept ist ein Teil eines größeren Konzeptes – im Sinne dieser Arbeit also des Quality-Gate-Konzeptes.

Ziel dieses Kapitels ist eine Identifikation und Abstrahierung dieser Teilkonzepte. Diese abstrahierten Teilkonzepte können in einem Domänenmodell dargestellt werden – dem abstrahierten Domänenmodell. Das abstrahierte Domänenmodell bildet einen wesentlichen und grundlegenden Teil dieser Arbeit. Es liefert einen Überblick aller Teilkonzepte, die ein Unternehmen bei Ausgestaltung eines Quality-Gate-Referenzprozesses berücksichtigen muss. Ein derartiges Modell ist gegenwärtig nicht in der Literatur vorhanden und wurde daher als Teilleistung dieser Arbeit von Grund auf entwickelt.

Das abstrahierte Domänenmodell hat die folgenden Zielsetzungen:

- Es zeigt die Teilkonzepte der Domäne und benennt sie einheitlich. Da hierbei ein Prozess betrachtet wird, stellt jedes Teilkonzept entweder einen Prozessschritt, ein Artefakt oder eine Ressource dar.
- Es beschreibt die Beziehungen der Teilkonzepte zueinander. Hierzu gehören insbesondere Kardinalitäten und die Art der Beziehung.
- Es zeigt Vererbungsbeziehungen (Generalisierungen und Spezialisierungen) zwischen Teilkonzepten.

Das abstrahierte Domänenmodell ist für die weiteren Betrachtungen in zweierlei Hinsicht notwendig:

- Bestehende Quality-Gate-Referenzprozesse können leichter eingeordnet werden und miteinander verglichen werden, da das Modell eine einheitliche Terminologie schafft.
- Das Modell zeigt unter anderem diejenigen Teilkonzepte, die ausgestaltet werden können. Es bildet zusammen mit der Charakterisierung (die quasi das axiomatische Wissen über die Domäne bildet) das Rahmenwerk zur Ausgestaltung der einzelnen Teilkonzepte und ist Grundlage der weiteren Betrachtungen in den folgenden Kapiteln.

Da der Begriff *Ausgestaltung* essentiell für das weitere Verständnis dieser Arbeit ist, wird dieser Begriff zunächst definiert:

Definition: Die *Ausgestaltung* liefert auf Basis des abstrahierten Domänenmodells einen nutzbaren Quality-Gate-Referenzprozess. Dazu muss jedes Teilkonzept des abstrahierten Domänenmodells ausgestaltet werden.

Definition 10: Ausgestaltung

Durch den Vorgang der Ausgestaltung werden beispielsweise Teilkonzepte, die Aktivitäten darstellen, vor allem durch die Angabe von Unteraktivitäten konkretisiert. Rollen erhalten ein Profil und werden auf Rollen innerhalb eines Unternehmens abgebildet. Artefakte werden benannt und strukturell und inhaltlich beschrieben. Hilfsmittel werden konkretisiert und anwendbar gestaltet.

Zur Erstellung des abstrahierten Domänenmodells wurde in vier Schritten vorgegangen:

1. **Identifikation der Teilkonzepte:** Dazu wurden beteiligte Prozessschritte, Ressourcen und Artefakte identifiziert, die mit Quality Gates in Beziehung stehen.
2. **Abstraktion der Teilkonzepte:** Die identifizierten Teilkonzepte wurden durch Generalisierung zusammengefasst.

3. **Bestimmung der Beziehungen:** Hierbei wurde die Art und Kardinalität der Beziehung zwischen den verschiedenen Teilkonzepten ermittelt.
4. **Paketbildung:** Dazu wurden die Teilkonzepte thematisch auf verschiedene Pakete aufgeteilt. Jedes Paket widmet sich einem zentralen Aspekt eines Quality-Gate-Referenzprozesses. Zu beachten ist hierbei, dass keines der Pakete ausgelassen werden kann. Nur die Ausgestaltung der Teilkonzepte aller Pakete bildet letztlich einen nutzbaren und vollständigen Quality-Gate-Referenzprozess. Die Pakete dienen daher ausschließlich der Übersicht. Die Aufteilung der Teilkonzepte auf Pakete erfolgt auf Grundlage der Dissertation von Hawlitzky [Hawlitzky '02], der allerdings nur strukturelle Konzepte, inhaltliche Konzepte und Steuerungskonzepte unterscheidet. Anpassungs- und Reviewkonzepte blendet Hawlitzky hingegen aus.

Im folgenden Abschnitt wird das abstrahierte Domänenmodell paketweise beschrieben.

5.1 Elemente des abstrahierten Domänenmodells

Dieser Abschnitt beschreibt paketweise die Teilkonzepte, die mit Quality Gates in Beziehung stehen. Nach der Ausgestaltung können diese Teilkonzepte in einem Quality-Gate-Referenzprozess zusammengefasst werden. Er bildet einen eigenständigen optionalen Teilprozess des Entwicklungs-Referenzprozesses (vgl. Abbildung 44) mit eigenen Prozessschritten, Rollen, Artefakten, Hilfsmitteln und eigenen Regeln zur Anpassung. Dies bedeutet, dass ein Entwicklungs-Referenzprozess auch ohne Quality-Gate-Referenzprozess vollständig ist und in einem Projekt angewendet werden kann. Da ein Quality-Gate-Referenzprozess jedoch von den Prozessschritten, den Artefakten und Ressourcen des Entwicklungsprozesses abhängt, kann er nicht unabhängig vom Entwicklungs-Referenzprozess ausgestaltet und angepasst werden. Hierauf wird näher in Kapitel 10 eingegangen.



Abbildung 44: Beziehung zwischen Entwicklungsprozess und Quality-Gate-Referenzprozess

5.1.1 Strukturelle Konzepte

Strukturkonzepte befassen sich mit der strukturellen Einbettung von Quality Gates in mindestens einen Entwicklungsprozess. Sie klären die Frage nach der Anzahl, Auswahl und Anordnung von Quality Gates. Strukturkonzepte sind im Paket *Struktur* zusammengefasst, das die folgenden wichtigen Teilkonzepte enthält:

- **Quality Gate:** Das Teilkonzept Quality Gate steht im Mittelpunkt des Struktur-Paketes. Ein Quality Gate liegt stets zwischen mindestens zwei Prozessschritten. Im Falle eines synchronisierenden Quality Gates sind auch mehrere eingehende und ausgehende Prozessschritte möglich.
- **Gate-Netzwerk:** Ein Gate-Netzwerk definiert eine bestimmte Anzahl, Auswahl oder Anordnung von Gates. Ein Gate-Netzwerk, das aus keinem Gate besteht, ist ebenfalls ein gültiges Gate-Netzwerk. Jeder Quality-Gate-Referenzprozess definiert mindestens ein Gate-Netzwerk.

Abbildung 45 fasst die Teilkonzepte des Paketes *Struktur* und ihre Beziehung untereinander zusammen.

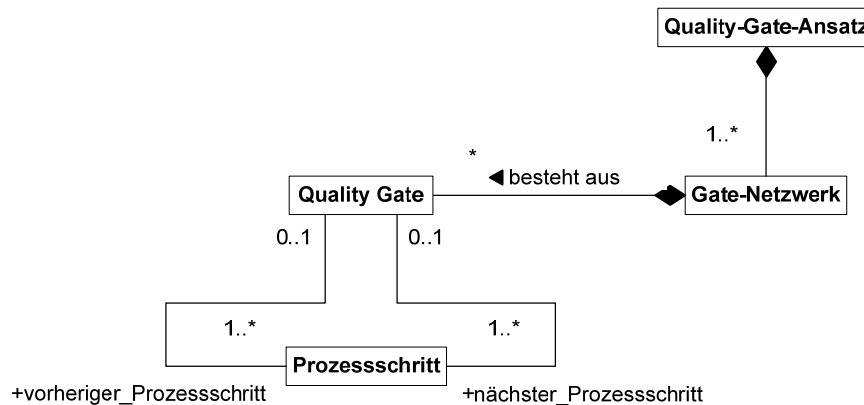


Abbildung 45: Teilkonzepte des Paketes Struktur

5.1.2 Inhaltliche Konzepte

Die inhaltlichen Konzepte umfassen alle inhaltlichen Belange eines Quality-Gate-Referenzprozesses. Die entsprechenden Teilkonzepte werden im Paket *Inhalt* zusammengefasst. Das Paket *Inhalt* enthält die folgenden wichtigen Teilkonzepte:

- **Kriterienerstellung:** Die Kriterienerstellung fasst die verschiedenen Aktivitäten zusammen, die der (systematischen) Erstellung von Kriterien dienen.
- **Ersteller:** An der Kriterienerstellung wirken Ersteller mit; sie unterstützen die Kriterienerstellung aktiv und entscheiden über die notwendigen Kriterien.
- **Kriterium:** Kriterien werden für jedes Quality Gate eines Projektes in einer Checkliste zusammengefasst. Projektübergreifend können Kriterien in einem Kriterienkatalog festgehalten werden, der bei der Kriterienerstellung verwendet werden kann oder muss.
- **Ergebnis:** Ein Ergebnis ist ein (physikalisch oder elektronisch vorhandenes) Produkt oder ein Kennzahlwert und besitzt einen Lieferanten und mindestens einen Kunden. Der Lieferant eines Produkts ist der Ersteller und ein Kunde der Empfänger des Produktes. Lieferanten eines Kennzahlwertes sind diejenigen Projektbeteiligten, die den Kennzahlwert beeinflussen können. Beispielsweise kann ein Tester die Testüberdeckung beeinflussen. Ein Kunde eines Kennzahlwertes ist ein Stakeholder, der ein Interesse am Kennzahlwert hat.

Abbildung 46 fasst die Teilkonzepte des Paketes *Inhalt* und ihre Beziehung untereinander zusammen.

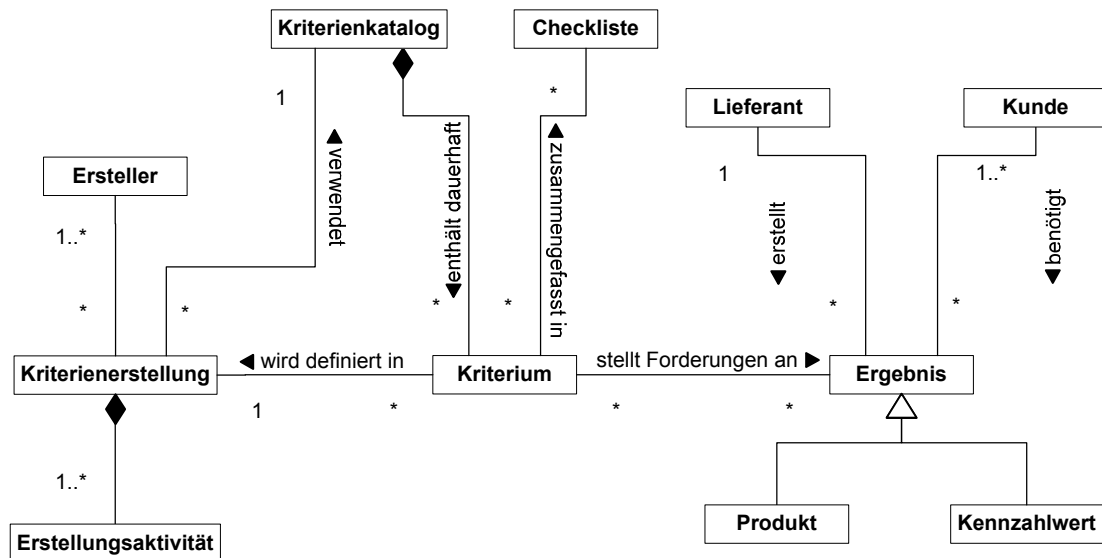


Abbildung 46: Teilkonzepte des Paketes Inhalt

5.1.3 Reviewkonzepte

Reviewkonzepte beschäftigen sich mit der Prüfung der Ergebnisse gegen Kriterien. Diese Teilkonzepte werden im Paket *Review* zusammengefasst. Das Paket *Review* enthält die folgenden wichtigen Teilkonzepte:

- **Gate-Review:** Das Gate-Review bildet das zentrale Teilkonzept im Paket *Review*. Ein Gate-Review besteht aus verschiedenen (mindestens aber einer) Reviewaktivitäten, die zusammengekommen den systematischen Ablauf des Gate-Reviews bilden. Eine Reviewaktivität kann entweder Teil der Organisation, der Prüfung oder des Abschlusses eines Gate-Reviews sein.
- **Protokoll:** Ein Protokoll dokumentiert neben der getroffenen Entscheidung weitere Informationen.
- **Protokollant:** Ein Protokollant erstellt das Protokoll im Gate-Review.
- **Gutachter:** Ein Gutachter bewertet die für ein Quality Gate vorzulegenden Ergebnisse auf Grundlage von Kriterien. Ein Gutachter kann keine Entscheidungen treffen, jedoch kann er eine Entscheidung und Maßnahmen empfehlen.
- **Projektvertreter:** Ein Projektvertreter ist ein Repräsentant des Projektes, der am Gate-Review teilnehmen kann, um die Interessen des Projektes zu wahren.
- **Gate-Verantwortlicher:** Ein Gate-Verantwortlicher ist verantwortlich für die ordnungsgemäße Durchführung aller Aktivitäten, die mit einem Gate-Review in Zusammenhang stehen.

Abbildung 47 fasst die Teilkonzepte des Paketes *Review* und ihre Beziehung untereinander zusammen.

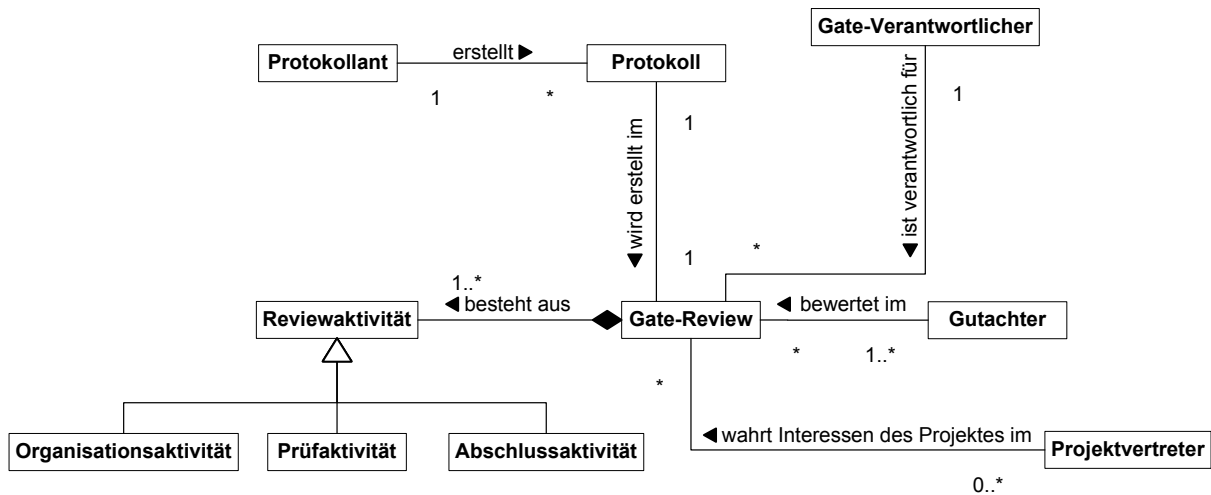


Abbildung 47: Teilkonzepte des Paketes Review

5.1.4 Steuerungskonzepte

Steuerungskonzepte umfassen diejenigen Teilkonzepte, die mit der Projektfortsetzung – also mit der unmittelbaren Entscheidung – in Beziehung stehen. Diese Teilkonzepte werden im Paket *Steuerung* zusammengefasst. Das Paket *Steuerung* enthält die folgenden wichtigen Teilkonzepte:

- **Entscheidung:** Das Gate-Review bildet das zentrale Teilkonzept im Paket *Steuerung*. Eine Entscheidung kann entweder eine uneingeschränkte bzw. bedingte Projektfortsetzung, eine Projektzurückstellung, eine Abbruchentscheidung oder Entscheidung über die Wiederholung eines Quality Gates sein. Die Entscheidung beinhaltet ebenfalls zu treffende Maßnahmen.
- **Gatekeeper:** Gatekeeper fällen die Entscheidung im Gate-Review. Eine Entscheidungsunterstützung (dies kann zum Beispiel eine Verdichtung auf die „Ampel“ sein) erleichtert dabei das Treffen der Entscheidung.

Abbildung 48 fasst die Teilkonzepte des Paketes *Steuerung* und ihre Beziehung untereinander zusammen.

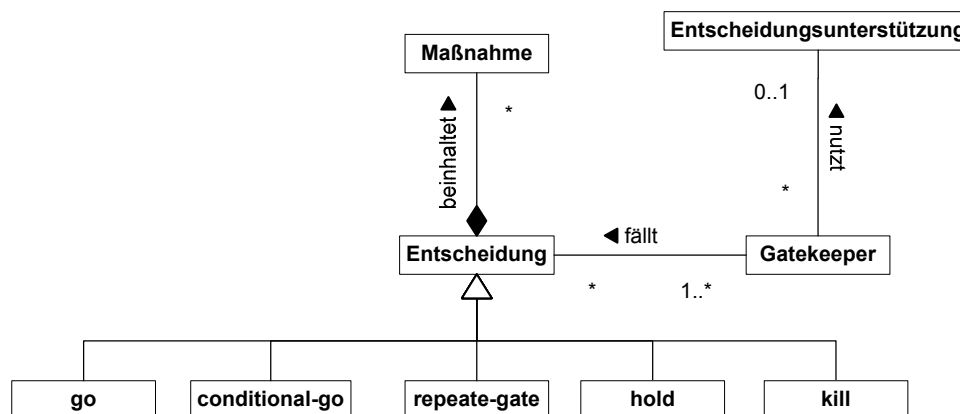


Abbildung 48: Teilkonzepte des Paketes Steuerung

5.1.5 Anpassungskonzepte

Anpassungskonzepte beschäftigen sich mit der Anpassung eines Quality-Gate-Referenzprozesses an eine gegebene Projektsituation. Darüber hinaus gehören Teilkonzepte dazu, die sich mit der Pflege eines Quality-Gate-Referenzprozesses beschäftigen. Teilkonzepte, die die Anpassung betreffen, werden im Paket *Anpassung* zusammengefasst.

Im Folgenden wird der Begriff *Pflege* definiert.

Definition: Die *Pflege* bezeichnet den Vorgang, durch den ein **bereits** ausgestaltetes Teilkonzept anders ausgestaltet wird. Das Ziel ist dabei eine Verbesserung des Quality-Gate-Referenzprozesses.

Definition 11: Pflege

Das Paket *Anpassung* enthält die folgenden wichtigen Teilkonzepte:

- **Regel:** Eine Regel bewertet ein auswählbares Element auf Grundlage einer Projektsituation. Regeln werden in einem Regelwerk zusammengefasst.
- **Projektmodell:** Ein Projektmodell fasst die für die Anpassung eines Quality-Gate-Referenzprozesses relevanten Projektmerkmale und zugehörigen Projektmerkmalswerte zusammen. Es existiert genau ein Projektmodell innerhalb eines ausgestalteten Quality-Gate-Referenzprozesses. Regeln sowie ein Tailoring-Verfahren beziehen sich auf ein Projektmodell.
- **Tailoring-Verfahren:** Ein Prozess-Tailorer nutzt ein Tailoring-Verfahren, um für eine gegebene Projektsituation einen geeigneten Quality-Gate-Referenzprozess zu bestimmen. Ein Tailoring-Verfahren ist unterschiedlich stark formalisiert. Das Spektrum reicht über Algorithmen, Heuristiken bis hin zum intuitiven Vorgehen. Abhängig vom Grad der Formalisierung eines Tailoring-Verfahrens, sind die notwendigen Regeln ebenfalls unterschiedlich stark formalisiert.
- **Gate-Management:** Das Gate-Management eines Unternehmens beschäftigt sich im Wesentlichen mit der Pflege des etablierten Quality-Gate-Referenzprozesses, insbesondere mit der Pflege des Regelwerks und des Tailoring-Verfahrens.
- **Prozess-Tailorer:** Ein Prozess-Tailorer ist für die Anpassung eines Quality-Gate-Referenzprozesses an eine konkrete Projektsituation verantwortlich. Je nach Formalisierung und Verbindlichkeit des Tailoring-Verfahrens besitzt er eine unterschiedlich starke Einflussnahme auf den Vorgang der Anpassung.

Abbildung 49 fasst die Teilkonzepte des Paketes *Anpassung* und ihre Beziehung untereinander zusammen.

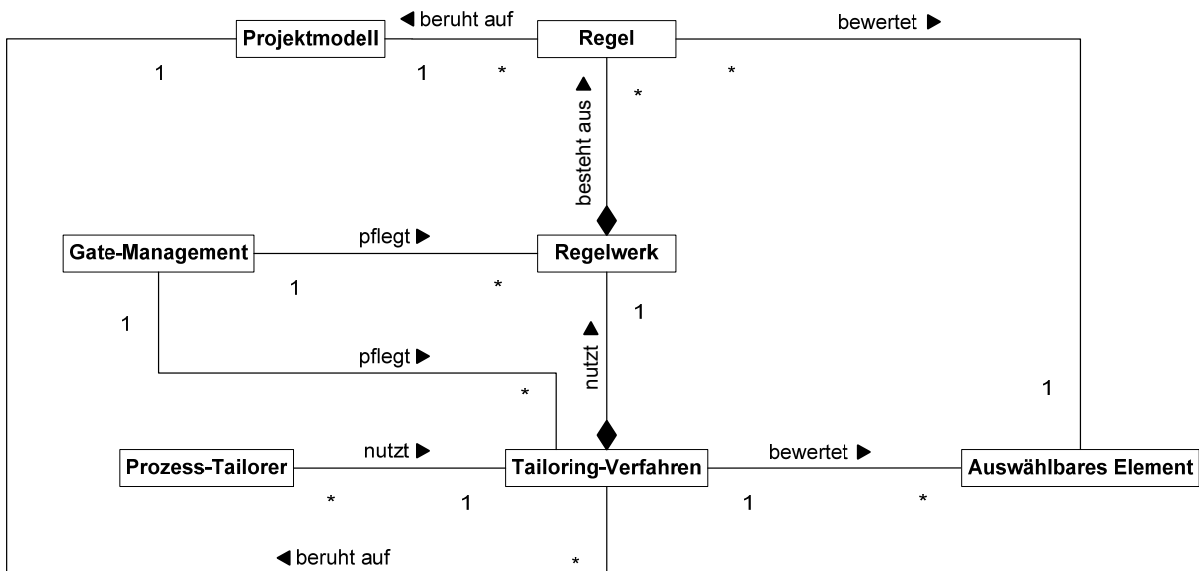


Abbildung 49: Teilkonzepte des Paketes Anpassung

5.2 Paketübergreifende Beziehungen

Einige Teilkonzepte haben Bezug zu Teilkonzepten anderer Pakete. Diese Beziehungen sollen in diesem Abschnitt betrachtet werden.

Die Beziehung des Quality Gates mit dem Gate-Review ist so stark, dass eine Teil-von-Beziehung gerechtfertigt ist (Abbildung 50). Ebenso ist die Entscheidung elementarer Bestandteil des Quality Gates. Die Entscheidung wird im Gate-Review gefällt, so dass eine Teilnahme von Gatekeepern im Gate-Review erforderlich ist. Eine getroffene Entscheidung wird im Protokoll festgehalten.

Eine Checkliste leitet das Review an, da sie zu prüfende Kriterien enthält, die die geforderten Ergebnisse erfüllen müssen.

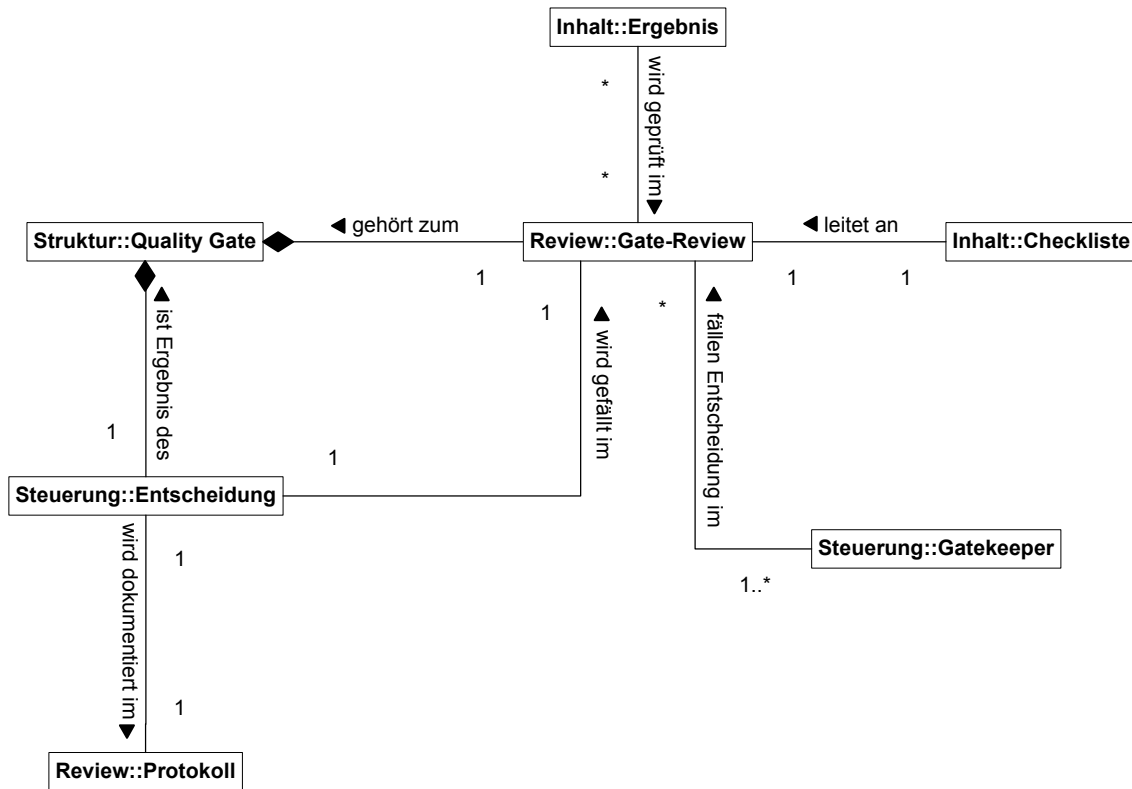


Abbildung 50: Beziehungen von Teilkonzepten untereinander

5.3 Überblick über das weitere Vorgehen

Innerhalb dieses Kapitels wurden die Teilkonzepte vorgestellt, die mit Quality Gates in Beziehung stehen. Diese Teilkonzepte wurden auf verschiedene Pakete aufgeteilt und im abstrahierten Domänenmodell zusammengefasst. Zusammen mit der Charakterisierung und Abgrenzung bildet es die Grundlage für die weiteren Betrachtungen. In den folgenden Kapiteln 6 bis 10 wird auf Richtlinien und Empfehlungen zur Ausgestaltung, sowie auf das Spektrum der Ausgestaltung der einzelnen Teilkonzepte eingegangen. Strukturell orientiert sich die Aufteilung der Kapitel dabei an den Paketen des abstrahierten Domänenmodells.

Die Kapitel 7, 8 und 9 stellen in dieser Reihenfolge die Verknüpfung zum Projektmanagement-Regelkreis (vgl. Abbildung 15) her. Dabei können durch die Ausgestaltung der inhaltlichen Konzepte die Planungsaktivitäten, durch die Ausgestaltung der Reviewkonzepte die Kontrollaktivitäten und durch die Ausgestaltung der Steuerungskonzepte die Steuerungsaktivitäten realisiert werden.

Nach Abschluss der Ausgestaltung erhält ein Unternehmen einen nutzbaren Quality-Gate-Referenzprozess. Dieser kann im Rahmen der Pflege durch ein Gate-Management optimiert werden. Durch die Anpassung erhält das Unternehmen einen Quality-Gate-Prozess der für eine bestimmte Projektsituation geeignet ist (Abbildung 51). Letztlich kann der Quality-Gate-Prozess instanziiert werden, indem Rollen Personen zugewiesen werden, Ergebnisse und Kriterien konkretisiert werden, Termine für Quality Gates festgelegt werden etc.

Wie ein Unternehmen die einzelnen Teilkonzepte für sich ausgestaltet, hängt maßgeblich von der gewählten grundlegenden Strategie ab (vgl. Kapitel 0):

Quality Gates als einheitliche Qualitäts-Leitlinie oder *Quality Gates als flexible Qualitätsstrategie*.

Hierauf wird gegebenenfalls in den nachfolgenden Kapiteln eingegangen.

Bei der Ausgestaltung existiert ein Spektrum an Alternativen, das ebenfalls berücksichtigt werden muss. Eine aufwändige Ausgestaltung ist beispielsweise nicht in jedem Kontext sinnvoll. Will ein Unternehmen beispielsweise erstmalig einen Quality-Gate-Referenzprozess ausgestalten und nutzen, so ist eine leichtgewichtige Ausgestaltung der Teilkonzepte unter Umständen sinnvoller. Im Rahmen der Pflege kann dieser Quality-Gate-Referenzprozess dann weiter ausgebaut und optimiert werden.

Die Untersuchungen der nächsten Kapitel beziehen sich nahezu ausschließlich auf Quality Gates in Entwicklungsprozessen. Insbesondere werden daher die weiteren Phasen des Entstehungsprozesses nicht mit in die Untersuchungen einbezogen – wengleich die meisten Ergebnisse auch auf sie übertragen werden können.

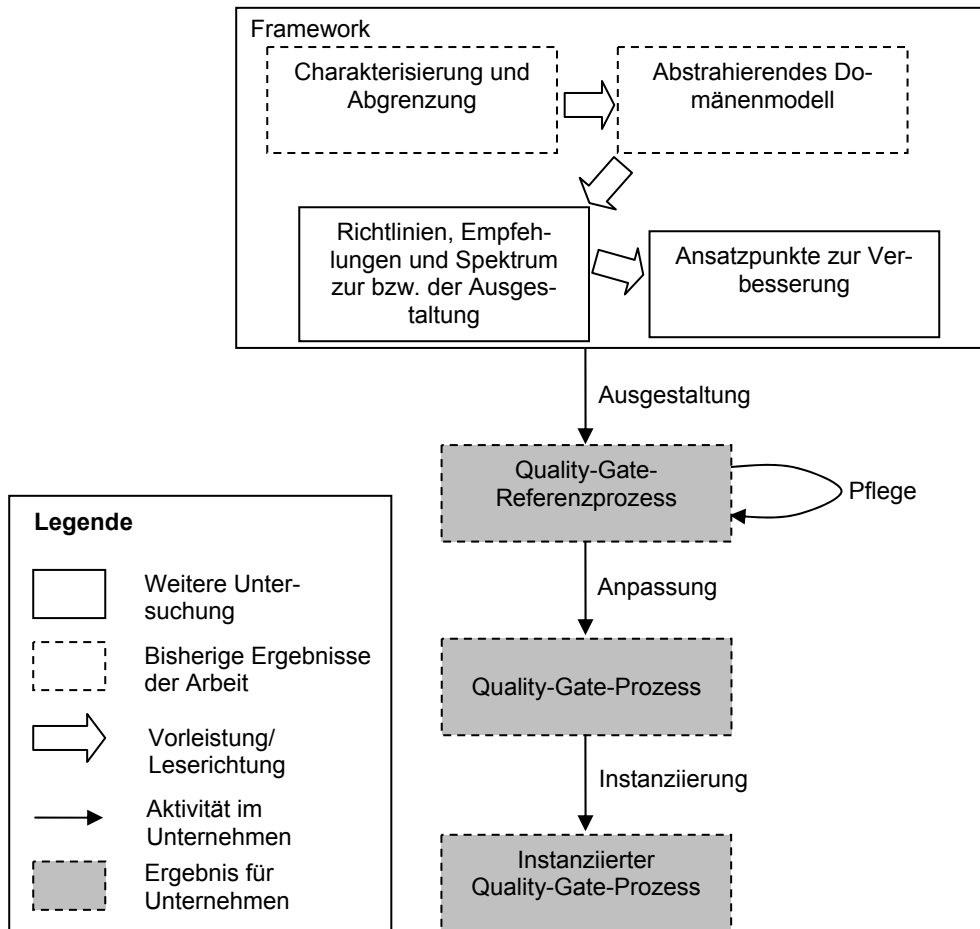


Abbildung 51: Übersicht der Ausgestaltung und Anpassung

6 Ausgestaltung der strukturellen Konzepte

Die Ausgestaltung der strukturellen Konzepte beschäftigt sich mit der Anordnung, der Anzahl und Auswahl von Quality Gates; sie zeigen grob gesprochen, wann welche Quality Gates zu passieren sind. Eine Anordnung, Anzahl und Auswahl von Quality Gates werden in den verschiedenen Gate-Netzwerken eines Quality-Gate-Referenzprozesses festgehalten.

In diesem Abschnitt werden Gate-Netzwerke für verschiedene Vorgehensweisen untersucht. Dabei sind die wasserfallartige, die iterative und die inkrementelle Vorgehensweise von besonderem Interesse. Zusammen mit der iterativen Vorgehensweise wird auch die agile Vorgehensweise betrachtet. Weiterhin werden Gate-Netzwerke für Projekte untersucht, die sich aus mehreren parallel verlaufenden Teilprojekten zusammensetzen oder verschiedene Handlungsstränge aufweisen.

An dieser Stelle wird kurz auf die Herkunft des Begriffs Gate-Netzwerk eingegangen. Betrachtet man die Anordnung der Prozessschritte eines Entwicklungsprozesses, so kann parallel dazu die Anordnung von Quality Gates betrachtet werden. Die Anordnung von Quality Gates legt sich dann wie ein Netz über die Anordnung der Prozessschritte. Diese Anordnung kann aus diesem Grund als Gate-Netzwerk³ betrachtet werden. Dies bedeutet jedoch, dass ein Entwicklungsprozess mindestens auf Phasenebene beschrieben sein muss, um ein geeignetes Gate-Netzwerk mit (ergebnisorientierten) Quality Gates ausgestalten zu können.

Zunächst wird das Teilkonzept Gate-Netzwerk weiter formalisiert.

6.1 Formalisierung des Gate-Netzwerk-Konzeptes

Entwicklungsprozess und Gate-Netzwerk können im gewissen Rahmen unabhängig von einander definiert werden. Die Einschränkungen ergeben sich wie folgt:

- Die Anzahl der Quality Gates in einem Gate-Netzwerk kann nicht größer sein als die Anzahl der (herausragenden) Prozessschritte im zugehörigen Entwicklungsprozess. Herausragende Prozessschritte sind dabei Phasen, Iterationen, Inkremente und in Einzelfällen auch Aktivitäten.
- Die Auswahl entscheidet, welche Quality Gates ein Projekt passieren muss. Es können nur solche Quality Gates gewählt werden, die sich auf einen Prozessschritt des Entwicklungsprozesses beziehen, dem das Projekt folgt.
- Die Anordnung der Quality Gates im Gate-Netzwerk muss mit der Anordnung der Prozessschritte im Entwicklungsprozess übereinstimmen.

Für einen Entwicklungsprozess sind daher verschiedene Gate-Netzwerke möglich. Ebenso passt jedes Gate-Netzwerk zu mindestens einem Entwicklungsprozess. Das geeignete Gate-Netzwerk für eine Projektsituation eines gegebenen Projektes wird über den Vorgang der Anpassung bestimmt. Gate-Netzwerk und Entwicklungsprozess ergeben zusammen einen Entwicklungsprozess mit Quality Gates.

Im Folgenden wird das Teilkonzept des Gate-Netzwerkes formal definiert.

³ Die Herkunft des Begriffes Gate-Netzwerk geht auf Hawlitzky [Hawlitzky '02] zurück.

Definition: Ein Gate-Netzwerk ist ein Vier-Tupel $(V_{\text{QG}}, E_{\text{control}}, V_{\text{start}}, V_{\text{end}})$, das einen speziellen Graphen bildet. Die einzelnen Elemente des Tupels sind wie folgt definiert:

- V_{QG} ist eine Menge von Quality Gates – sie bildet die Menge der Knoten des Graphen.
- E_{control} enthält gerichtete Kontrollflüsse, die die Quality Gates aus V_{QG} miteinander verbinden.
- $V_{\text{start}} \subseteq V_{\text{QG}}$ ist eine nichtleere Menge von Quality Gates, mit denen das Gate-Netzwerk starten kann.
- $V_{\text{end}} \subseteq V_{\text{QG}}$ ist eine nichtleere Menge von Quality Gates, mit denen das Gate-Netzwerk enden kann.

Des Weiteren hat das Gate-Netzwerk die Eigenschaften, dass jedes Quality Gate über Kontrollflüsse von mindestens einem der „startenden“ Quality Gates aus erreicht werden kann.

Definition 12: Definition Gate-Netzwerk

Hinsichtlich der Semantik gilt folgendes:

- Ein Gate-Netzwerk kann im Falle von Iterationen und Inkrementen im Entwicklungsprozess Schleifen enthalten. Dies bedeutet, dass ein Quality Gate gleichen Typs mehrfach durchgeführt werden kann. Ein Kontrollfluss führt in diesem Fall zu einem vorherigen Quality Gate (oder zum gleichen Quality Gate) zurück. Innerhalb einer Schleife befindet sich also mindestens ein Quality Gate. Die Kontrollfluss-Schleife kann mit einer Kardinalität versehen werden, die angibt, bei welchen Iterationen oder Inkrementen, die in der Schleife befindlichen Quality Gates durchlaufen werden müssen. Ein Wert $n > 1$ ($n \in \mathbb{N}$) bedeutet, dass die Quality Gates in der Schleife nur bei jedem n-ten Schleifendurchlauf durchlaufen werden müssen. Fehlt diese Angabe, so werden die Quality Gates in der Schleife bei jedem Schleifendurchlauf durchlaufen (dies ist gleichbedeutend mit $n = 1$).
- Jedes Quality Gate hat einen eindeutigen Bezeichner, der die Lage relativ zu einem vorhandenen Prozessschritt des Referenzprozesses angibt. Dabei gilt, dass die Bezeichnung „Fertig für x “, die Positionierung vor einem Prozessschritt x und „ x abgeschlossen“, die Positionierung nach einem Prozessschritt x kennzeichnet. Es sind auch andere Bezeichnungen erlaubt, solange die Positionierung relativ zu einem Prozessschritt eindeutig ist. Für Iterationen und Inkremente gibt es eine spezielle Regel, da ein Quality Gate bei jeder Iteration ausgeführt werden kann oder nur vor der ersten und nach der letzten Iteration bzw. vor und nach dem letzten Inkrement. Hierfür existieren jeweils die Bezeichnungen „Fertig für nächste/s Iteration/Inkrement x “, „Fertig für erste/s Iteration/Inkrement x “ und „Letzte/s Iteration/Inkrement x abgeschlossen“.
- Sind x und y zwei aufeinander folgende Prozessschritte, so bezeichnen „ x abgeschlossen“ und „Fertig für y “ ein und dasselbe Quality Gate. Inhaltlich kann jedoch ein Unterschied gemacht werden: beim Quality Gate „ x abgeschlossen“ werden stärker Ausgangskriterien des Prozessschrittes x geprüft, während beim Quality Gate „Fertig für y “ stärker Eingangskriterien des Prozessschrittes y geprüft werden.

Zur Beschreibung von Gate-Netzwerken wird die folgende Notation verwendet:


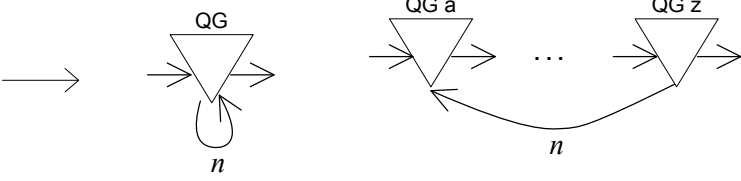
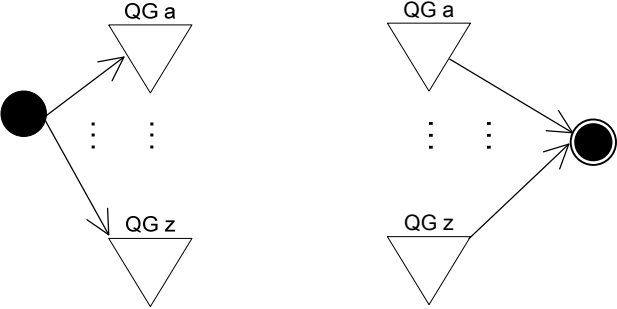
Notationselement	Symbol
Quality Gate	
Einfacher Kontrollfluss und Kontrollfluss-Schleifen mit Kardinalität (für Iterationen und Inkremente)	
Start- und Endmarkierung	

Tabelle 24: Notationselemente und jeweilige Symbole für Gate-Netzwerke

Gate-Netzwerke können nicht beliebig aufgebaut sein, sondern müssen stets Bezug zu einem Entwicklungsprozess haben (siehe auch obige Einschränkungen). Abbildung 53 zeigt jeweils ausschnittsweise den Bezug zwischen einem Entwicklungsprozess mit Quality Gates (rechte Seite) und verschiedenen Gate-Netzwerken (linke Seite). Zur Illustration dient dabei der folgende Entwicklungsprozess:

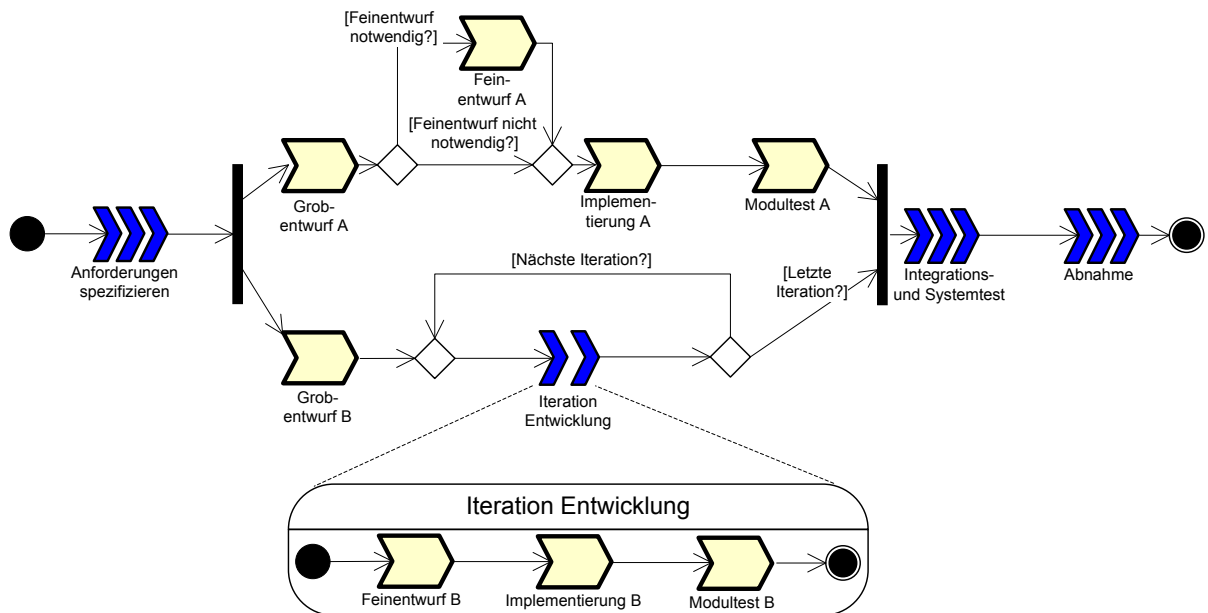


Abbildung 52: Entwicklungsprozess zur Illustration verschiedener Gate-Netzwerke

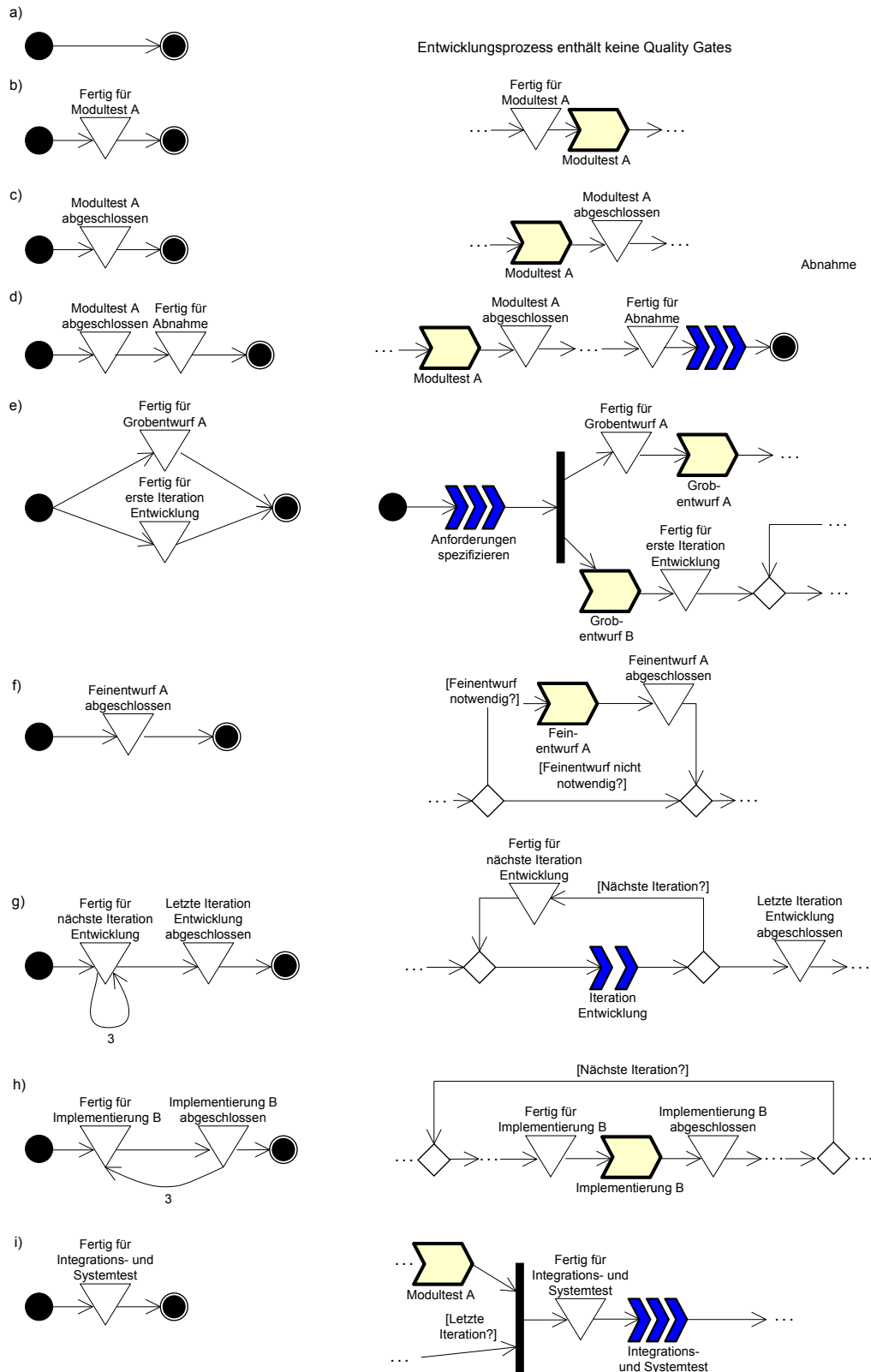


Abbildung 53: Beispielhafter Bezug verschiedener Gate-Netzwerke zum Entwicklungsprozess

Gate-Netzwerke lassen sich nun aus den verschiedenen in Abbildung 53 beschriebenen Situationen zusammensetzen.

6.2 Allgemeine Richtlinien zur Ausgestaltung von Gate-Netzwerken

Für die Ausgestaltung eines Gate-Netzwerkes existieren verschiedene Richtlinien:

- Die Anzahl der Quality Gates sollte nicht zu hoch sein, da die Durchführung der mit einem Quality Gate verbundenen Aktivitäten Ressourcen benötigt und sich ein aufwändiges Gate-Review verzögernd auf das Projekt auswirkt. Eine Höchstgrenze von 15 ergibt sich aus der Untersuchung verschiedener Ansätze hinsichtlich der Anzahl der Quality Gates. Üblich sind allerdings Gate-Netzwerke mit einer Anzahl von fünf bis sieben Quality Gates. Kleinere Projekte (beispielsweise Ein-Personen-Projekte) können hingegen auch ganz ohne Quality Gates auskommen. Eine weitere wichtige Größe ist das Risiko eines Projektes. Risikoreichere Projekte sollten stets eine größere Anzahl von Quality Gates aufweisen als risikoärmere Projekte. Abbildung 54 fasst diese Erkenntnisse zusammen. Insbesondere wird die maximale Anzahl von Quality Gates schnell überschritten, wenn nach jeder Iteration oder jedem Inkrement ein Quality Gate durchgeführt werden soll. In diesem Fall ist es sinnvoller, beispielsweise nur nach jeder zweiten oder dritten Iteration ein Quality Gate durchzuführen. Dies entspricht in etwa einem Zeitraum von zwei bis drei Monaten, wenn man eine übliche Iterationslänge von 30 Tagen zu Grunde legt.
- Es existiert nur eine begrenzte Anzahl von verschiedenen Quality Gates, die maximal der Anzahl der verschiedenen Prozessschritte im Entwicklungsprozess entspricht. Jedes Gate-Netzwerk wird aus diesen Quality Gates zusammengesetzt. Die verfügbaren Quality Gates werden in einer Menge M_{QG} zusammengefasst.
- Quality Gates werden hinter den herausragenden Prozessschritten (in den meisten Fällen Phasen) platziert (vgl. Abschnitt 4.8.1). Insbesondere heißt dies auch, dass ein Unternehmen zumindest auf Phasenebene einen definierten Entwicklungsprozess besitzen muss.
- Quality Gates als Abschluss von Prozessschritten, die sich mit Anforderungen beschäftigen und Quality Gates vor einem Abnahme-Prozessschritt sind besonders wichtig. Erstere, da qualitative Probleme bei den Anforderungen erhebliche negative Auswirkungen auf nachfolgende Prozessschritte haben können. Letztere, da die Abnahme einen Prozessschritt mit Außenwirkung darstellt, bei dem qualitativ hochwertige Software an einen Kunden auszuliefern ist. Ein Gate-Netzwerk für ein Projekt mit einer typischen Projektsituation enthält daher mindestens ein Quality Gate nach der Anforderungsphase und ein Quality Gate vor der Abnahme.
- Es können Quality Gates mit ausschließlich interner Wirkung und mit externer Wirkung unterschieden werden. Ein Quality Gate mit interner Wirkung fordert keine Ergebnisse, die für den externen Kunden direkt relevant bzw. nutzbar sind, wie zum Beispiel das Entwurfsdokument. Quality Gates mit externer Wirkung fordern Ergebnisse, die für den externen Kunden direkt relevant bzw. nutzbar sind, wie beispielsweise die Anforderungsspezifikation. Typische Quality Gates mit externer Wirkung sind beispielsweise die Quality Gates „Anforderungen spezifizieren abgeschlossen“ und „Abnahme abgeschlossen“. Die verbleibenden Quality Gates sind meistens intern (vgl. Abschnitt 4.5.3).

Einfluss der grundlegenden Strategie auf die Ausgestaltung

Ein Unternehmen legt durch die Wahl einer der beiden grundlegenden Strategien die Ausgestaltung der strukturellen Konzepte in folgender Weise fest:

- Fällt die Wahl auf die Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie*, so existiert lediglich ein Gate-Netzwerk, dass in allen Projekten verwendet wird. Auf diese Weise wird sichergestellt, dass zwei Projekte miteinander verglichen werden können:

„Die Projekt A und B haben beide die Quality Gates x und y erfolgreich durchschritten. Projekt B ist jedoch am Quality Gate z gescheitert und muss daher das Quality Gate z wiederholen.“

- Wählt das Prozessmanagement eines Unternehmens die Strategie *Quality Gates als flexible Qualitätsstrategie*, so ist ihm die Vergleichbarkeit zwischen Projekten hinsichtlich durchschrittener Quality Gates weniger wichtig. Folglich kann es verschiedene Gate-Netzwerke geben, die besser auf verschiedene Projektsituation zugeschnitten sind. Allerdings sollte in ähnlichen Projektsituationen auch ein ähnliches Gate-Netzwerk verwendet werden, um eine gewisse qualitative Vergleichbarkeit zu schaffen. Es ist somit nicht sinnvoll, eine Vielzahl an unterschiedlichen Gate-Netzwerken zuzulassen. Die Anzahl der Quality Gates eines Gate-Netzwerkes orientiert sich an der Wichtigkeit und dem Risiko eines Projektes (vgl. Abbildung 54). Die Auswahl der konkreten Quality Gates orientiert sich an den inhaltlichen Aspekten und die Anordnung am Entwicklungsprozess eines Projektes.

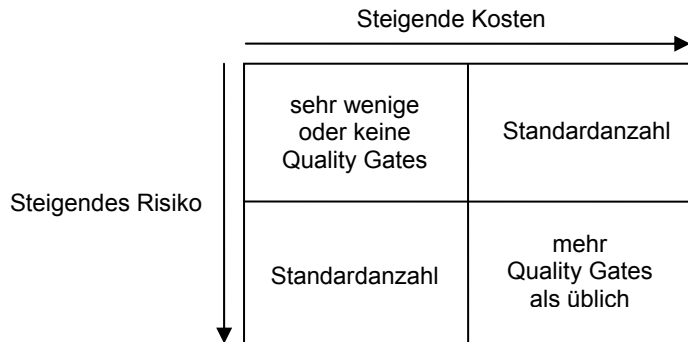


Abbildung 54: Anzahl der Quality Gates in Abhängigkeit von Wichtigkeit und Risiko

6.3 Gate-Netzwerke bei wasserfallartigem Vorgehen

Das wasserfallartige Vorgehen [Royce '98] zeichnet sich durch eine sequenzielle Anordnung der Entwicklungs-Aktivitäten aus. Jede Aktivität beschäftigt sich dabei mit einem Aspekt der Software-Entwicklung. Rücksprünge sind eingeschränkt in die vorherige Aktivität möglich, um Probleme dort beheben zu können. Das wasserfallartige Vorgehen eignet sich besonders gut für die Entwicklung von sicherheitskritischer Software. Es setzt dabei jedoch voraus, dass die Anforderungen stabil sind und die Risiken weitestgehend nach dem Projektstart bekannt sind, da späte Änderungen der Anforderungen erhebliche Kosten verursachen, da sie nur mit erheblichen Aufwand eingepflegt werden können. Abbildung 55 zeigt beispielhaft eine typische sequenzielle Anordnung von Aktivitäten ohne rückwärtigen Kontrollfluss. Aktivitäten können durch Quality Gates abgeschlossen werden, müssen dies jedoch nicht. Sind Unteraktivitäten definiert, so können auch Unteraktivitäten durch Quality Gates abgeschlossen werden.

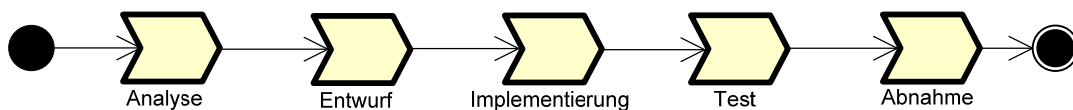


Abbildung 55: Beispiel für Quality Gates in einem wasserfallartigen Entwicklungsprozess

Abbildung 56 zeigt ein mögliches Gate-Netzwerk für das wasserfallartige Vorgehen.

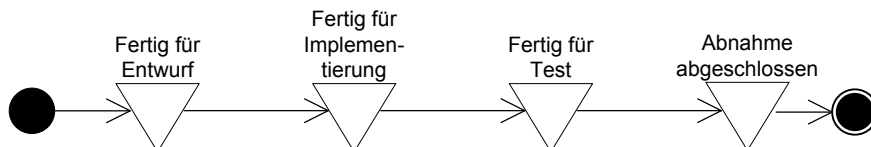


Abbildung 56: Beispiel für ein mögliches Gate-Netzwerk beim wasserfallartigen Vorgehen

6.4 Gate-Netzwerke bei iterativem Vorgehen

Im Gegensatz zum wasserfallartigen Vorgehen, ist das iterative Vorgehen besonders gut für Projekte mit eher instabilen Anforderungen und eher hohen Entwicklungsrisiken geeignet. Die Entwicklung erfolgt daher in Iterationen, die stets mit einer Verifikation oder Validierung der erstellten Produkte enden. Das auf diese Weise gewonnene Feedback fließt dann in die nächste Iteration hinein und verringert auf diese Weise das Risiko und wirkt sich stabilisierend auf die Anforderungen aus.

Allgemein lässt sich der iterative Entwicklungsprozess wie folgt definieren:

Definition: Ein *iterativer* Entwicklungsprozess ist ein Prozess, in dem eine bestimmte Abfolge von Aktivitäten mehr als einmal mit dem Ziel durchgeführt wird, sich dem eigentlichen Entwicklungsziel approximativ zu nähern. Jede Iteration schließt mit einer Verifikation oder Validierung ab. Die sich wiederholende Abfolge heißt *Iteration*.

Definition 13: Iterativer Entwicklungsprozess

Das iterative Vorgehen geht auf oberster Ebene von Phasen aus. Jede Phase widmet sich vorrangig einem Aspekt der Software-Entwicklung. Jede Phase kann sich in beliebig viele Iterationen unterteilen. Ein typischer Vertreter für das iterative Vorgehen ist der *Rational Unified Prozess (RUP)* [Kruchten '99] (vgl. Abbildung 57).

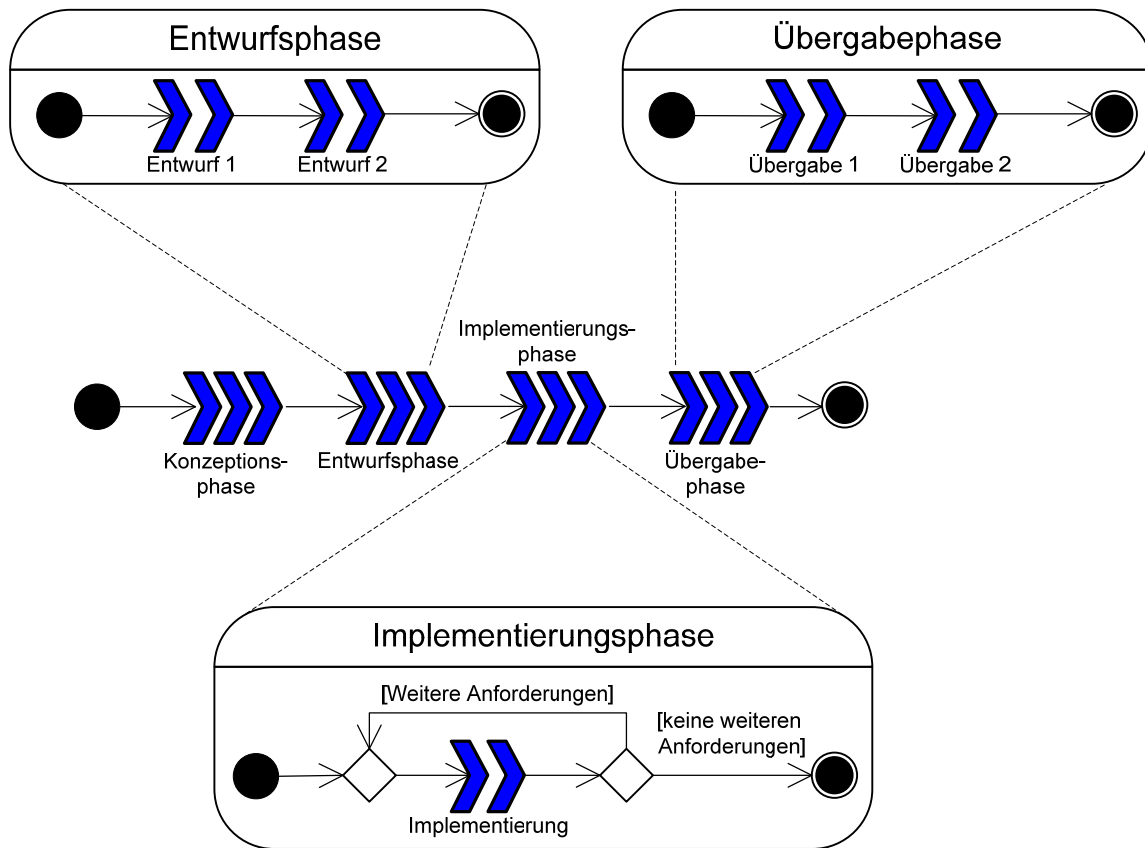


Abbildung 57: Phasen und Iterationen im RUP

Nach Wallin et al. [Wallin und Ekdahl et al. '02, Wallin und Larsson et al. '02] ist für den RUP auf Phasenebene das folgende Gate-Netzwerk sinnvoll:

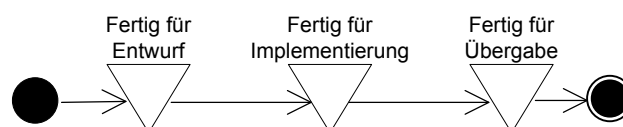


Abbildung 58: Gate-Netzwerk auf Phasenebene für den RUP

Allgemeiner betrachtet kann jede Phase durch ein Quality Gate abgeschlossen werden.

Die Iterationsebene stellt sich für jede Phase wie folgt dar:

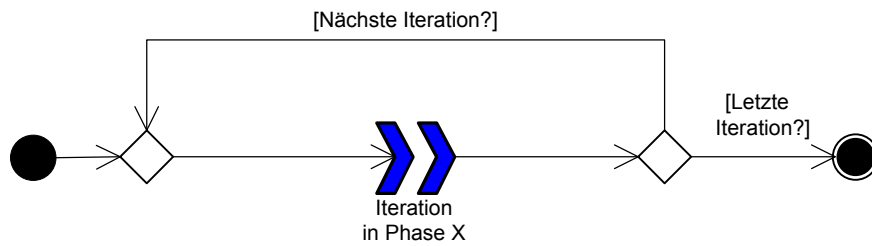


Abbildung 59: Aufbau einer Phase

Da jede Iteration für sich genommen ein vergleichsweise großer Block ist, kann die Einführung von Quality Gates vor Start oder zum Abschluss bestimmter Iterationen sinnvoll sein. Soll beispielsweise hinter jeder zweiten Iteration in der Implementierungs-Phase des RUP ein Quality Gate durchgeführt werden, so kann dies durch das folgende Gate-Netzwerk ausgedrückt werden:

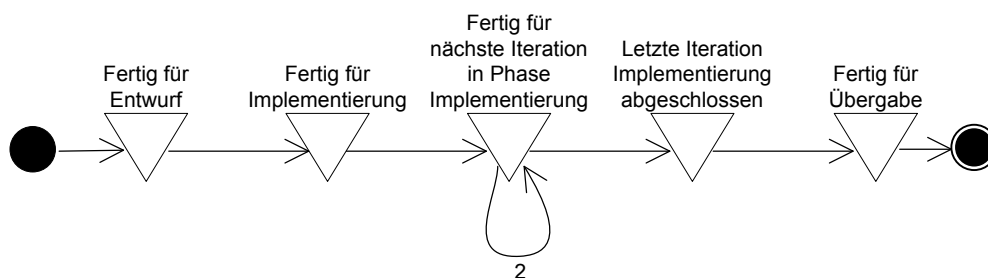


Abbildung 60: Mögliches Gate-Netzwerk für den RUP

6.5 Gate-Netzwerke bei agilem Vorgehen

Das agile Vorgehen eignet sich besonders gut, wenn sehr instabile Anforderungen vorliegen und die zu entwickelnde Software in keinem sicherheitsrelevanten Umfeld eingesetzt werden soll. Agile Entwicklungsprozesse legen ein iteratives Vorgehen mit häufigen Lieferungen (der in Teilen implementierten) Software an den Kunden nahe (sog. *Small Releases*), sie können daher als eine extreme Form des iterativen Vorgehens angesehen werden. Teillieferungen schließen eine Iteration ab und ermöglichen somit ein Feedback durch den Kunden. Beim agilen Vorgehen lassen sich ebenfalls Phasen bilden. Jede Phase endet mit der Lieferung eines großen und abgeschlossenen Teils der Software an den Kunden (sog. *Releases*).

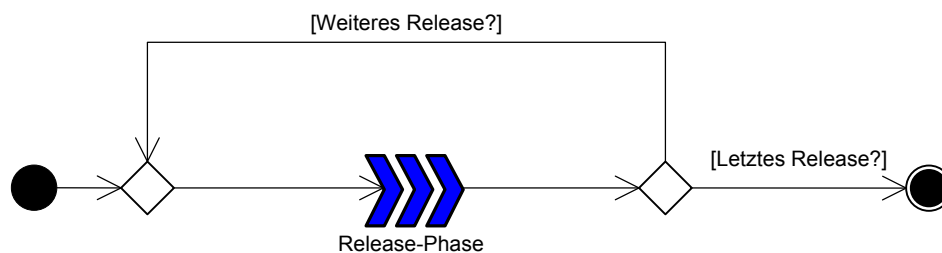


Abbildung 61: Phasenmodell beim Agilen Vorgehen

Jede Release-Phase besteht aus beliebig vielen Iterationen:

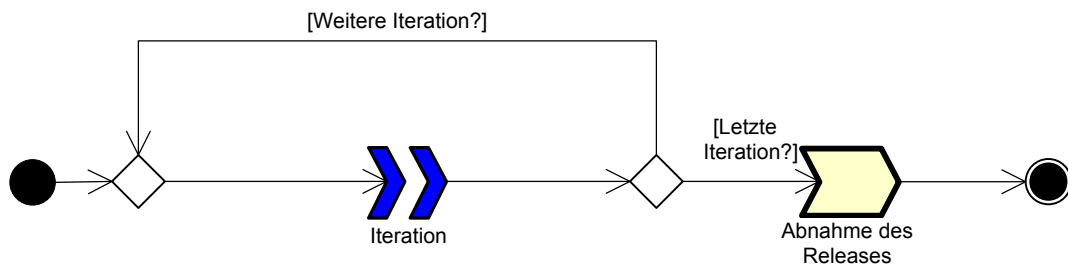


Abbildung 62: Aufbau einer Release-Phase

Jede Iteration beginnt mit der Iterationsplanung und wird mit einem Akzeptanztest abgeschlossen. Dazwischen findet die eigentliche Entwicklung statt, die sich jedoch nicht ohne weiteres in verschiedene Aktivitäten aufschlüsseln lässt (vgl. Abbildung 63).

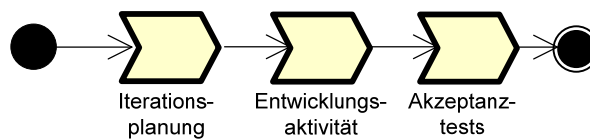


Abbildung 63: Aufbau einer Iteration beim agilen Vorgehen

Die Einführung eines Quality Gates beim agilen Vorgehen ist jeweils vor der Abnahme eines Releases gegen Ende einer Release-Phase sinnvoll. Weitere mögliche Positionen liegen zwischen den Iterationen oder vor dem Akzeptanztest einer Iteration. Abbildung 64 zeigt ein mögliches Gate-Netzwerk für das agile Vorgehen, bei dem nach jeder Iteration und vor jeder Abnahme eines Releases ein Quality Gate durchgeführt werden soll.

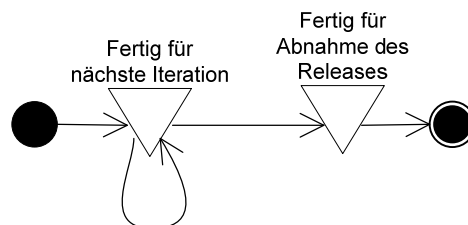


Abbildung 64: Mögliches Gate-Netzwerk beim agilen Vorgehen

6.6 Gate-Netzwerke bei inkrementellem Vorgehen

Das inkrementelle Vorgehen geht im Gegensatz zum iterativen Vorgehen von stabilen Anforderungen aus. Folglich werden die Anforderungsphase und die Phase Grobentwurf nur einmal durchlaufen. Implementierung und der Feinentwurf erfolgen in so genannten Inkrementen. Allgemein lässt sich der inkrementelle Entwicklungsprozess wie folgt definieren:

Definition: Ein *inkrementeller* Software-Entwicklungsprozess ist ein Prozess, bei dem Teile der Software nach und nach entwickelt und zusammengefügt werden. Die Erweiterung der Software durch einen weiteren Teil heißt *Inkrement*. Kennzeichnend ist, dass bereits erstellte Inkremente bereits eingesetzt und auf ihnen aufgebaut werden kann.

Definition 14: Inkrementeller Software-Entwicklungsprozess

Auf Phasenebene stellt sich der inkrementelle Entwicklungsprozess wie folgt dar:

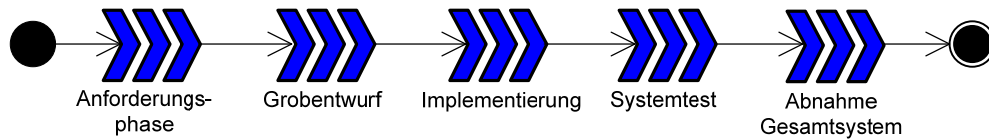


Abbildung 65: Phasenmodell beim inkrementellen Vorgehen

Die Implementierung unterteilt sich in beliebig viele Inkremente:

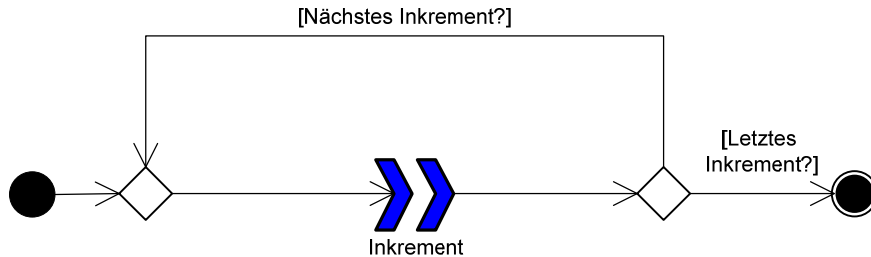


Abbildung 66: Aufbau der Phase Implementierung

Das Vorgehen in jedem Inkrement verläuft dabei strukturiert:

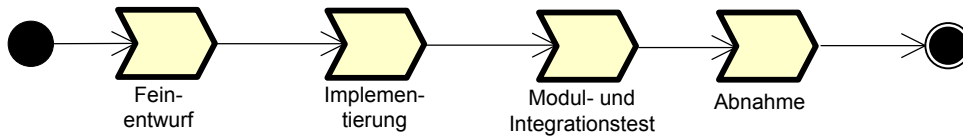


Abbildung 67: Aufbau eines Inkrementes

Auf Phasenebene können Quality Gates nach jeder Phase eingeführt werden. Weiterhin kann die Positionierung nach jedem Inkrement und nach jeder Aktivität eines Inkrementes sinnvoll sein. Ein mögliches Gate-Netzwerk besitzt jeweils ein Quality Gate nach jeder Phase und vor jeder Abnahmeaktivität (vgl. Abbildung 68).

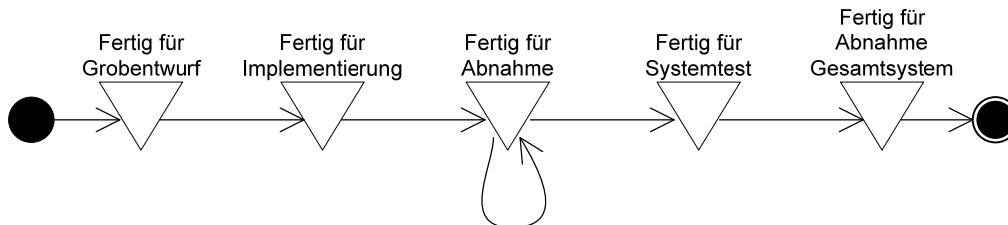


Abbildung 68: Mögliches Gate-Netzwerk für das inkrementelle Vorgehen

6.7 Gate-Netzwerke bei parallel verlaufenden Handlungssträngen

In einem Software-Entwicklungsprozess können Iterationen, Inkremente oder Aktivitäten parallel ausgeführt werden. In diesem Fall ergeben sich parallel verlaufende (fachlich jedoch verschiedene) Handlungsstränge, die jeweils über eigene Quality Gates verfügen können. An bestimmten Punkten werden diese Handlungsstränge wieder synchronisiert. An diesen Stellen sind Quality Gates sinnvoll, da hier die Qualität zwischen Disziplinen koordiniert werden muss. Ebenso existieren Punkte im Prozess, an denen die Entwicklung verzweigt. Diese Punkte sind ebenfalls geeignet für Quality Gates, da hier rückblickend die Erfüllung von Vorleistungen geprüft werden kann. Abbildung 69 zeigt beispielhaft einen Entwicklungsprozess mit zwei parallel verlaufenden Handlungssträngen.

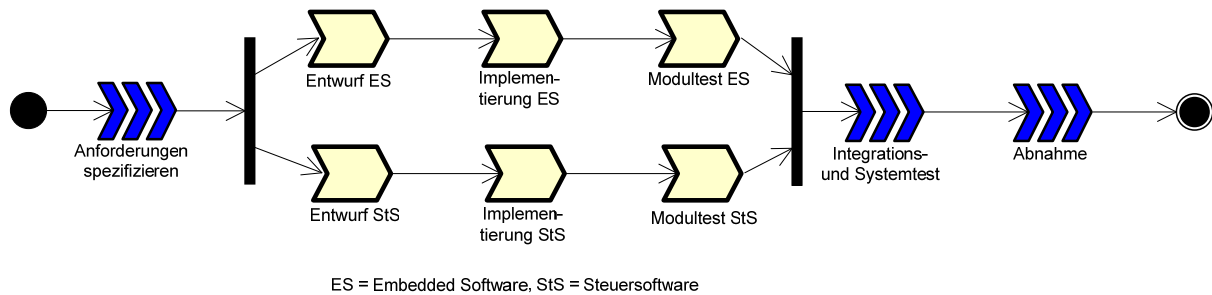


Abbildung 69: Entwicklungsprozess mit zwei parallel verlaufenden Handlungssträngen

Abbildung 70 zeigt ein dazu passendes Gate-Netzwerk mit jeweils einem speziellen Quality Gate in jedem Handlungsstrang und zwei Quality Gates, von denen eines als Verzweigungspunkt und eines als Synchronisationspunkt genutzt wird.

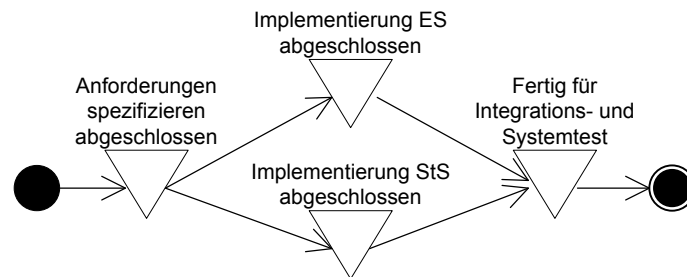


Abbildung 70: Mögliches Gate-Netzwerk mit parallel angeordneten Quality Gates

6.8 Gate-Netzwerke bei Projekten mit mehreren Teilprojekten

Häufig stellt die Entwicklung von Software nur ein Teilprojekt eines größeren Projektes dar. Das Gesamtprojekt folgt dann in Phasen, die jeweils mit Quality Gates abgeschlossen werden können (vgl. Abschnitt 4.4). Die Quality Gates auf Gesamtprojekt-Ebene dienen vor allem der Evaluierung der Qualität des Gesamtprojektes und der Synchronisation der Teilprojekte, jedoch weniger der Qualitätssicherung innerhalb der Teilprojekte. Daher ist die Ergänzung um Quality Gates im Teilprojekt „Software“ sinnvoll. Die Positionierung hängt dabei wiederum stark vom Vorgehen innerhalb des Teilprojektes „Software“ ab, wobei das Vorgehen in Phasen im Gesamtprojekt ein wasserfallartiges Vorgehen im Teilprojekt „Software“ nahe legen kann.

6.9 Operationen auf Basis-Gate-Netzwerken

Jedes Gate-Netzwerk lässt sich von Grund auf einzeln notieren – dies ist jedoch aufwändig, da es (selbst für einen einzelnen Entwicklungsprozess) eine Vielzahl möglicher Gate-Netzwerke geben kann. Einfacher ist es hingegen, von Basis-Gate-Netzwerken auszugehen. Ein Basis-Gate-Netzwerk spiegelt die übliche Anordnung, Auswahl und Anzahl von Quality Gates für einen Entwicklungsprozess wider.

Basis-Gate-Netzwerke können über verschiedene Operationen so angepasst werden, dass sie für eine Projektsituation besser geeignet sind, als eines der Basis-Gate-Netzwerke selbst. Diese Idee fußt auf dem Flexibilitäts-Konzept nach Cooper (vgl. Abschnitt 4.6.1.1). Die Operationen umfassen dabei das Hinzufügen und das Löschen von Quality Gates. Ferner stellt eine Modifikation der Kardinalität des Kontrollflusses bei Iterationen und Inkrementen eine mögliche Operation dar [Schulz '07]. Im Rahmen der Anpassung sind dann ein Basis-Gate-Netzwerk und die eventuell erforderlichen Operationen zu wählen.

Im Folgenden werden diese Operationen weiter formalisiert. Für jedes Gate-Netzwerk γ kann eine Menge

$$M_{\text{QG}}^{\gamma}$$

gebildet werden. M_{QG}^{γ} enthält die bereits im Gate-Netzwerk γ vorhandenen Quality Gates und die Quality Gates, um die das Gate-Netzwerk γ noch erweitert werden kann. Ein Gate-Netzwerk lässt sich jedoch nicht beliebig um Quality Gates erweitern, stattdessen müssen die Einschränkungen bezogen auf den zugehörigen Entwicklungsprozess berücksichtigt werden (vgl. Abschnitt 6.1). M_{QG}^{γ} enthält nur solche Quality Gates, um die sich γ unter Berücksichtigung der gegebenen Einschränkungen erweitern lässt.

Sei B die Menge aller Basis-Gate-Netzwerke innerhalb eines Quality-Gate-Referenzprozesses. Weiterhin sei Θ die Menge aller Quality Gates mit

$$\Theta = \bigcup_{b \in B} M_{\text{QG}}^b$$

und Γ die Menge der Gate-Netzwerke, die sich durch das beliebig häufige Hinzufügen oder Löschen eines Quality Gates oder durch die Modifikation der Kardinalität aus Basis-Gate-Netzwerken aus B ergeben. Dabei gilt insbesondere

$$B \subseteq \Gamma.$$

Die obigen Operationen werden im Folgenden formalisiert. Dabei ist zu beachten, dass bei den Operationen niemals angegeben werden muss, an welcher Stelle ein Quality Gate platziert werden soll. Die Stelle im Gate-Netzwerk ergibt sich stets durch den Bezug des jeweiligen Gate-Netzwerkes zu einem Entwicklungsprozess.

- **Hinzufüge-Operation:**

$$\varphi_{\text{add}} : \Gamma \times \Theta \rightarrow \Gamma$$

$$\varphi_{\text{add}} : (\gamma, \mathcal{G}) \mapsto \gamma',$$

wobei φ_{add} ein Gate-Netzwerk γ um das Quality Gate \mathcal{G} erweitert, falls $\mathcal{G} \in M_{\text{QG}}^{\gamma}$ ist. Im Falle eines Quality Gates in einer Iteration oder einem Inkrement wird die Kardinalität des Kontrollflusses auf 1 gesetzt. Dies heißt, dass das entsprechende Quality Gate bei jeder Iteration bzw. jedem Inkrement durchgeführt wird.

- **Lösch-Operation:**

$$\varphi_{\text{del}} : \Gamma \times \Theta \rightarrow \Gamma$$

$$\varphi_{\text{del}} : (\gamma, \mathcal{G}) \mapsto \gamma',$$

wobei φ_{del} das Quality Gate \mathcal{G} aus dem Gate-Netzwerk γ entfernt, falls γ das Quality Gate \mathcal{G} enthält.

- **Modifikations-Operation:**

$$\varphi_{\text{mod}} : \Gamma \times \Theta \times \mathbb{N} \rightarrow \Gamma$$

$$\varphi_{\text{mod}} : (\gamma, \mathcal{G}, n) \mapsto \gamma',$$

wobei φ_{mod} die Kardinalität des Kontrollflusses eines Quality Gates \mathcal{G} im Gate-Netzwerk γ auf n ändert. Dabei muss gelten, dass \mathcal{G} bereits im Gate-Netzwerk vorhanden und ein Quality Gate in einer Iteration oder einem Inkrement ist.

Hinzufüge-, Lösch- und Modifikations-Operation können hintereinander auf ein Gate-Netzwerk angewendet werden. Auf diese Weise erhält man einen Formalismus, der mächtig genug ist, um alle er-

denklichen (zum entsprechenden Entwicklungsprozess passenden) Gate-Netzwerke aus einem Basis-Gate-Netzwerk zu erzeugen. Erreichbar sind damit ebenfalls ein komplett leeres Gate-Netzwerk und folglich ein Entwicklungsprozess ohne Quality Gates, sowie auch ein Gate-Netzwerk mit maximaler Anzahl an Quality Gates.

Im Folgenden werden die drei Operationen kurz an einem Beispiel illustriert. Dazu wird das folgende, zum Entwicklungsprozess in Abbildung 52 passende, Basis-Gate-Netzwerk b gewählt:

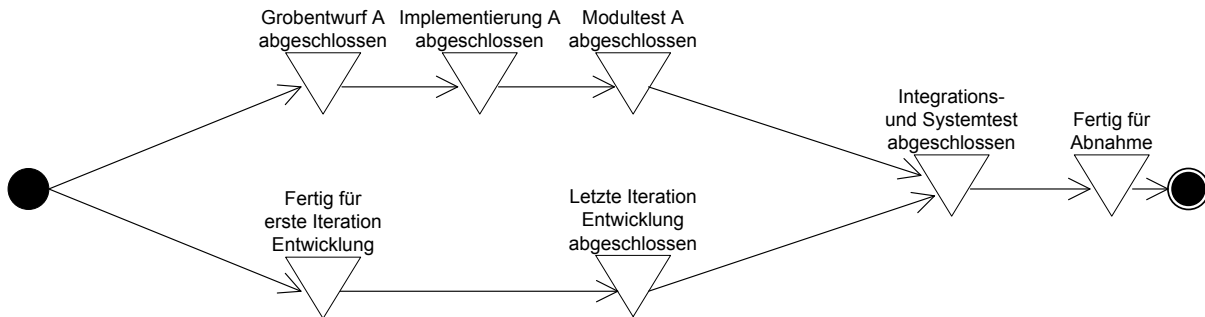


Abbildung 71: Basis-Gate-Netzwerk vor der Ausführung der Operationen

Die Menge M_{QG}^b ergibt sich für dieses Basis-Gate-Netzwerk zu:

$$M_{QG}^b = \overline{M}_{QG}^b \cup$$

{Anforderungen spezifizieren abgeschlossen, Fertig für nächste Iteration Entwicklung, Feinentwurf A abgeschlossen, Feinentwurf B abgeschlossen, Implementierung B abgeschlossen, Modultest B abgeschlossen}

wobei \overline{M}_{QG}^b die Menge aller bereits im Basis-Gate-Netzwerk b vorhandenen Quality Gates ist. Das Basis-Gate-Netzwerk b kann noch um die Quality Gates $M_{QG}^b - \overline{M}_{QG}^b$ erweitert werden.

Das Basis-Gate-Netzwerk wird nun für eine spezielle Projektsituation für ein wichtiges Projekt angepasst. Dazu soll nach jeder dritten Iteration „Entwicklung“ ein Quality Gate durchgeführt werden. Hingegen soll auf das Quality Gate „Letzte Iteration Entwicklung abgeschlossen“ verzichtet werden. Die Ausführung der Operationen in der folgenden Reihenfolge

$$\begin{aligned} b' &= \varphi_{\text{add}}(b, \text{Fertig für nächste Iteration Entwicklung}), \\ b'' &= \varphi_{\text{mod}}(b', \text{Fertig für nächste Iteration Entwicklung}, 3), \\ \text{und } \varphi_{\text{del}}(b'', \text{Letzte Iteration Entwicklung abgeschlossen}) \end{aligned}$$

führt auf das gewünschte Gate-Netzwerk (vgl. Abbildung 72).

Die Reihenfolge der Operationen kann nicht beliebig durchgeführt werden. Beispielsweise würde die Vertauschung der ersten beiden Operationen zu einem anderen Gate-Netzwerk führen, da die Modifikation der Kardinalität nicht vorgenommen werden kann, wenn das betroffene Quality Gate nicht im Gate-Netzwerk vorhanden ist. Um ein derartiges Problem zu vermeiden, müssen zunächst alle Hinzufüge-Operationen, dann alle Modifikations-Operationen und zuletzt alle Löschoptionen ausgeführt werden.

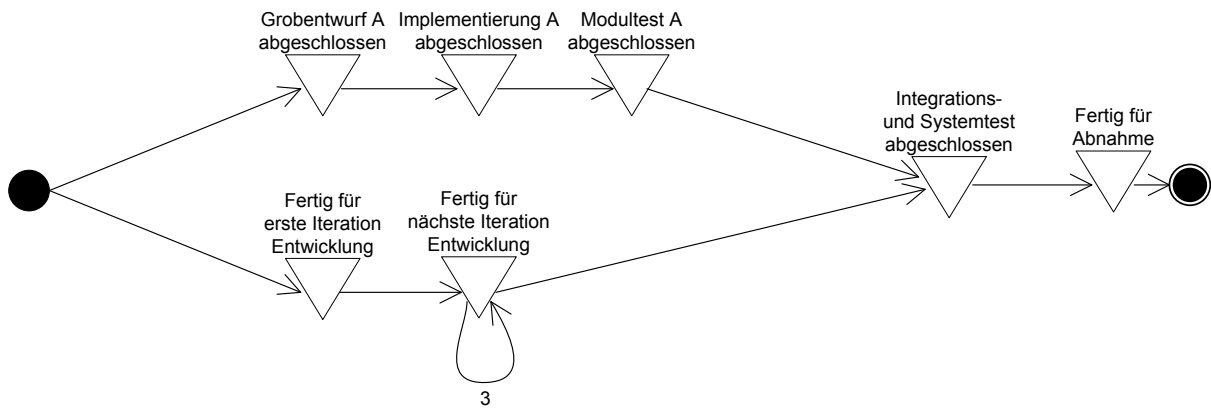


Abbildung 72: Basis-Gate-Netzwerk nach der Ausführung der Operationen

7 Ausgestaltung der inhaltlichen Konzepte

Die Ausgestaltung der inhaltlichen Konzepte beschäftigt sich damit, wie, wann und durch wen welche Kriterien für die Prüfung im Gate-Review erstellt werden. Das *Wie* und *Wann* wird durch die Ausgestaltung der Kriterienerstellung geregelt. Eine Ausgestaltung des Teilkonzepts *Ersteller* klärt die Frage nach dem *durch Wen*. Das *Was* wird durch die Teilkonzepte *Kriterien* und *Ergebnisse* festgelegt. Die nachfolgenden Abschnitte beschäftigen sich mit den Möglichkeiten zur Ausgestaltung dieser Teilkonzepte.

7.1 Kriterienerstellung

Die Kriterienerstellung ist das wichtigste Teilkonzept des Paketes *Inhalt*. Sie beschäftigt sich mit dem mehr oder weniger systematischen Vorgehen zur Erstellung der Kriterien. Erstellte Kriterien haben im Wesentlichen zwei Funktionen:

- **Bewertungsaspekt:** Kriterien dienen den Gutachtern als Bewertungsgrundlage für die geforderten Ergebnisse.
- **Steuerungsaspekt:** Sie zeigen den Lieferanten kompakt auf, welche Anforderungen an die Qualität der geforderten Ergebnisse gestellt werden. Ungeeignete Kriterien führen zur Lieferung der Ergebnisse in schlechter Qualität oder in unnötig hoher Qualität, wodurch unnötige Kosten verursacht werden.

Die Ausgestaltung der Kriterienerstellung ist stark abhängig von der gewählten grundlegenden Strategie. Die Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* verlangt es, jedes Projekt an den gleichen Kriterien zu messen. Diese Kriterien werden in einem Kriterienkatalog festgehalten. Hierbei ist es durchaus sinnvoll, bestimmte Kriterien auszulassen, weil sie nicht zu einer gegebenen Projektsituation passen. Ferner kann es sinnvoll sein, die Kriterien um weitere Kriterien in einem bestimmten Projekte zu ergänzen, weil beispielsweise ganz besondere qualitative Anforderungen an die Ergebnisse dieses Projektes gestellt werden.

Die Strategie *Quality Gates als flexible Qualitätsstrategie* erlaubt es, die Kriterien für alle Projekte gleich zu setzen oder aber auch individuell neu zu bestimmen. Bei Verfolgung dieser Strategie ist sogar eine Mischform beider Vorgehen möglich. So können beispielsweise für bekannte Projektsituation Kriterien aus einem etablierten festen Kriterienkatalog entnommen werden. Für neuartige Projektsituationen können die Kriterien hingegen neu und individuell bestimmt werden.

Die Wahl der grundlegenden Strategie beeinflusst damit also vor allem die *Individualität* der Kriterien. Es lassen sich noch zwei weitere Dimensionen unterscheiden, die weiter unten beschrieben werden. Zunächst folgen jedoch einige Definitionen, die wichtig für die weiteren Betrachtungen in diesem Abschnitt sind. Zunächst werden die Begriffe *Erstellung* und *Ergänzung* definiert.

Definition: Die *Erstellung* von Kriterien beschäftigt sich mit der einmaligen, dauerhaften und projektübergreifenden Erstellung eines Kriteriums im Falle des Vorgehens mit festem Kriterienkatalog. Die *Ergänzung* beschäftigt sich mit der temporären Hinzunahme von Kriterien für genau ein Projekt im Falle des Vorgehens mit festem Kriterienkatalog.

Im Fall von individuellen Kriterien (Vorgehen ohne Kriterienkatalog), beschäftigt sich die *Erstellung* mit der temporären Bildung von Kriterien für genau ein Projekt. Die so erstellten Kriterien müssen daher nicht notwendigerweise für einen größeren Kontext gelten.

Definition 15: Definition der Begriffe Erstellung und Ergänzung

Nachfolgend wird der Begriff *Auswahl* definiert.

Definition: Die *Auswahl* bei festem Kriterienkatalog umfasst die Möglichkeit, bestimmte Kriterien im Rahmen eines Projektes auszulassen.

Definition 16: Definition des Begriffes Auswahl

Zuletzt wird der Begriff Operationalisierung definiert.

Definition: Im Rahmen der *Operationalisierung* werden Kriterien (falls erforderlich) im Kontext eines Projektes interpretiert, so dass sie auf die Ergebnisse dieses Projektes anwendbar sind. Die Operationalisierung umfasst für jedes Kriterium die folgenden Interpretationsaufgaben:

- Das Maß wird konkretisiert, indem die Berechnungsvorschrift, der Definitionsbereich O oder der Wertebereich S angepasst werden.
- Das Messobjekt \bar{o} wird genau benannt.
- Der Sollwert \bar{s} wird konkretisiert, eventuell auch angepasst.
- Die Messung wird durch die Definition einer Messvorschrift ν konkretisiert.
- Der Indikator zur Visualisierung der Messwerte wird definiert oder angepasst.
- Das Kriterium wird eventuell aufgeteilt, um die Handhabung zu erleichtern.
- Das Kriterium wird gewichtet, um die Bedeutung des Kriteriums aufzuzeigen.

Definition 17: Definition der Operationalisierung von Kriterien

Die Kriterienerstellung umfasst die Teilaufgaben Erstellung, Ergänzung, Auswahl und Operationalisierung.

Mit den obigen Definitionen lassen sich nun die beiden weiteren Dimensionen *Zeitpunkt* und *Systematik* beschreiben:

- Der *Zeitpunkt* bestimmt, wann innerhalb eines Projektes Kriterien erstellt, ergänzt, ausgewählt oder operationalisiert werden. Mögliche Zeitpunkte sind hierbei Projektstart, Planungsphase und Durchführungsphase.
- Die *Systematik* bestimmt die Strukturiertheit des Vorgehens, das insbesondere für die Erstellung, aber auch für die Ergänzung, die Auswahl und die Operationalisierung befolgt werden muss. Die Strukturiertheit des Vorgehens lässt sich unterscheiden in zielorientierte, erfahrungsbasierte und in rein intuitive Vorgehen.

Die Dimensionen können nun miteinander verknüpft werden. Die folgende Tabelle zeigt die Verknüpfung der Dimensionen Zeitpunkt und Individualität und die jeweils resultierenden Vorgehen zur Kriterienerstellung.

		Zeitpunkt		
		Projektstart	Planungsphase	Durchführungsphase
Individualität	Fester Kriterienkatalog (mit Auswahl oder Ergänzung)	Kriterien werden aufgrund eines groben Projektmodells ausgewählt, ergänzt und operationalisiert. Sie müssen eventuell später noch weiter operationalisiert werden, da kaum konkrete Ergebnisse des Projektes bekannt sind.	Kriterien werden aufgrund des Projektstrukturplans oder während der Iterationsplanung ausgewählt, ergänzt und operationalisiert. Hierbei ist bereits eine Orientierung an den identifizierten Projektergebnissen möglich.	Kriterien werden im Vorfeld eines Quality Gates oder im Gate-Review (dann durch die Gutachter oder Gatekeeper) selbst ausgewählt, ergänzt und operationalisiert
	Kein Kriterienkatalog	Kriterien werden aufgrund eines groben Projektmodells erstellt. Sie müssen eventuell noch weiter operationalisiert werden, da noch keine konkreten Ergebnisse festliegen.	Kriterien werden aufgrund des Projektstrukturplans oder während der Iterationsplanung erstellt. Hierbei ist bereits eine Orientierung an den identifizierten Projektergebnissen möglich.	<i>Entweder werden</i> Kriterien im Vorfeld eines Quality Gates erstellt, für das sie gelten sollen <i>oder</i> sie werden im Extremfall erst im Gate-Review erstellt oder operationalisiert.

Tabelle 25: Verknüpfung der Dimensionen Individualität und Zeitpunkt

Zu jeder der sechs Kombinationen, die sich aus der Verknüpfung der Dimensionen Individualität und Zeitpunkt ergeben, kann ebenfalls die Systematik frei gewählt werden. Sie ergibt damit die dritte Dimension.

Beispielsweise lässt sich nun die Kriterienerstellung nach Pfeifer et al. [Pfeifer et al. '03] einordnen. In seiner Ausgestaltung wird kein Kriterienkatalog verwendet. Kriterien werden im Vorfeld eines Quality Gates (also in der Durchführungsphase) erstellt und operationalisiert. Hinsichtlich der Systematik wählt er ein erfahrungsbasiertes Vorgehen.

Die nachfolgenden drei Abschnitte gehen näher auf die drei Dimensionen Individualität, Zeitpunkt und Systematik ein.

7.1.1 Individualität der Kriterien

Kriterien, die verbindlich für jedes Projekt festliegen sollen, erfordern die Erstellung und Pflege eines Kriterienkataloges. Ein fester Kriterienkatalog ermöglicht es, den Fortschritt verschiedener Projekte miteinander vergleichen zu können, da alle Projekte an den gleichen Kriterien gemessen werden. Ein fester Kriterienkatalog (idealerweise ohne Möglichkeit zur Auswahl) ist daher zwingend bei Wahl der grundlegenden Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* erforderlich.

Der Kriterienkatalog enthält für jedes Quality Gate eine Checkliste von Kriterien. Die Kriterien spiegeln gesetztes Erfahrungswissen wider und wirken damit bekannten Risiken entgegen. Dieser Kriterienkatalog ist sorgfältig zu konzipieren, da er für alle Projektsituationen gelten muss. Die erforderlichen Maße sind entsprechend sorgfältig zu bestimmen (vgl. Anforderungen an ein gutes Maß, Abschnitt 3.3.3). Falls erforderlich, müssen einige Kriterien allgemeingültiger formuliert werden, so dass sie für alle Projektsituationen anwendbar sind. Ein Top-Down-Ansatz wie zum Beispiel GQM kann bei der Erstellung der Kriterien behilflich sein. Einen guten Kriterienkatalog zu erstellen erfordert langjährige Erfahrung mit den Projekten des betroffenen Unternehmens, sowie eine gewisse Gleichartigkeit der Projekte.

Insbesondere allgemeingültig formulierte Kriterien erfordern eine spätere Operationalisierung. Diese benötigt Ressourcen und lässt einen gewissen Interpretationsspielraum, der bei Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* eher unerwünscht ist. Deshalb sind die Kriterien des Kriterienkataloges möglichst so zu gestalten, dass eine spätere Operationalisierung nicht notwendig ist. Dieses Spannungsfeld zwischen allgemeingültiger Formulierung einerseits und zu vermeidender Operationalisierung andererseits ist bei Erstellung jedes einzelnen Kriteriums zu meistern. Viele Kriterien (jedoch nicht alle) müssen daher relativ einfach und klar formuliert sein.

Varianten des Vorgehens bei festem Kriterienkatalog lassen es zu, die verpflichtenden Kriterien um weitere Kriterien zu ergänzen oder nur eine (für die Projektsituation passende) Auswahl der verpflichtenden Kriterien zu prüfen. Im letzteren Fall kann es möglich sein, dass einige Kriterien auf jeden Fall nicht abwählbar sind. Durch die Auswahl werden der Nutzen und vermutlich auch die Akzeptanz der Kriterien erhöht, da ungeeignete Kriterien ausgelassen werden können. Hierbei ist jedoch streng darauf zu achten, dass nur solche Kriterien ausgelassen werden dürfen, die nicht zum Projekt passen. Nachteilig ist, dass der Fortschrittsvergleich zwischen verschiedenen Projekten erschwert wird, wenn die falschen Kriterien durch die Ersteller ausgelassen werden. Ergänzten Kriterien wirken sich hingegen nicht nachteilig auf die Möglichkeit zum Fortschrittsvergleich aus, da ein Projekt immer noch mindestens das gleiche Qualitätsniveau wie andere Projekte erreichen muss.

Ein vollständiger Verzicht auf einen Kriterienkatalog lässt viele Freiräume, die es ermöglichen, genau passende Checklisten für eine Projektsituation zu erstellen. Diese Kriterien können sehr gezielt auf die Qualitätsziele eingehen. Allerdings ist der Aufwand für die Erstellung höher, da die Kriterien für jedes Projekt (idealerweise mit hoher Systematik) neu erstellt werden müssen. Bei willkürlicher Festlegung besteht die Gefahr, dass Kriterien erstellt werden, deren Maße leicht anwendbar sind, jedoch nicht zielführend sind. Der Verzicht auf einen Kriterienkatalog stellt das ideale Vorgehen bei Wahl der grundlegenden Strategie *Quality Gates als flexible Qualitätsstrategie* dar. Jedoch sind auch alle anderen Varianten zulässig, die eine Anpassung der Checklisten erlauben. Daher ist auch das Vorgehen bei festem Kriterienkatalog mit Auswahl- oder Ergänzungsmöglichkeit erlaubt. Ferner ist es sinnvoll bei bekannten Standardprojekten auf einen festen Kriterienkatalog zurückgreifen zu können, da die Kriterien Erfahrungswissen widerspiegeln.

Tabelle 26 fasst Vor- und Nachteile des Vorgehens mit und ohne Kriterienkatalog zusammen.

+/-	Fester Kriterienkatalog	Kein fester Kriterienkatalog
Vorteile	<ul style="list-style-type: none"> - Qualitative Richtlinie für alle Projekte. - Erleichtert es, den Projektfortschritt mit anderen Projekten zu vergleichen. - Aufwand der Kriterienerstellung nur einmalig. Danach nur Operationalisierung. - Kann durch Auswahl oder Ergänzung in gewissen Rahmen flexibler gestaltet werden. 	<ul style="list-style-type: none"> - Kriterien sind ideal auf ein Projekt abgestimmt. Eine Operationalisierung ist daher meistens nicht nötig (außer bei Erstellung während des Projektstarts).
Nachteile	<ul style="list-style-type: none"> - Kriterien sind schwerer zu bestimmen, da sie für alle Projekte gelten müssen. - Akzeptanz ist vermutlich geringer trotz Operationalisierung. - Anwendbarkeit der Kriterien ohne Operationalisierung teilweise nicht möglich. 	<ul style="list-style-type: none"> - Aufwand für Erstellung erhöht sich, da sie für jedes Projekt erfolgen muss. - Vergleich des Projektfortschritts ist schwieriger oder sogar unmöglich, da Kriterien individuell sind.

Tabelle 26: Vor- und Nachteile des Vorgehens mit und ohne Kriterienkatalog

7.1.2 Zeitpunkt der Kriterienerstellung

Ein fester Kriterienkatalog muss vor dem Projektstart vorhanden sein, damit er genutzt werden kann. Aufgrund der Abstraktheit einiger Kriterien, ist es notwendig, diese Kriterien während des Projekt-

starts, der Planungsphase oder der Projektdurchführung zu operationalisieren, so dass sie auf die Ergebnisse eines Projektes anwendbar sind. Dies kann spätestens im Gate-Review passieren, wo die Kriterien dann zu bewerten sind. Dies ist jedoch zu spät, da dann die durch die Kriterien geforderte Qualität nicht rechtzeitig für die Lieferanten sichtbar ist. Gleichzeitig kann es jedoch vorteilhaft sein, die Operationalisierung spät durchzuführen, da dann genauere Informationen über die Ergebnisse eines Projektes vorliegen. Der optimale Weg dürfte daher darin bestehen, die Operationalisierung in der Planungsphase oder mit längerem Vorlauf vor dem betroffenen Quality Gate durchzuführen, da hier ausreichend viele Informationen zu den Ergebnissen vorliegen und die operationalisierten Kriterien noch rechtzeitig für die Lieferanten sichtbar sind. Die gleichen Überlegungen gelten auch für die Auswahl und Ergänzung von Kriterien.

Wird kein Kriterienkatalog verwendet, so können die Kriterien während des Projektstarts, während der Planungsphase oder der Projektdurchführung erstellt werden. Hinsichtlich der Sichtbarkeit und den optimalen Zeitpunkt gelten hierbei die gleichen Überlegungen wie beim Vorgehen bei festen Kriterienkatalog.

Tabelle 27 zeigt Vor- und Nachteile der frühen oder späten Erstellung, Operationalisierung, Ergänzung und Auswahl im Projekt.

+/-	Frühe Erstellung, Operationalisierung, Ergänzung und Auswahl	Spätere Erstellung, Operationalisierung, Ergänzung und Auswahl
Vorteile	<ul style="list-style-type: none"> - Kernteam des Projektes (und damit das Management) hat stärkeren Einfluss auf die Kriterien. - Kriterien haben stärkere Bedeutung, da sie durch das Kernteam des Projektes erstellt werden. - Kriterien sind länger sichtbar für Lieferanten. 	<ul style="list-style-type: none"> - Kriterien stimmen besser mit den Erfordernissen des Projektes überein.
Nachteile	<ul style="list-style-type: none"> - Stärkere Belastung für das Kernteam. - Kriterien sind unter Umständen weniger geeignet, da noch nicht genug Informationen über das Projekt vorliegen. 	<ul style="list-style-type: none"> - Die Kriterien werden eventuell weniger stark beachtet, da sie nicht unbedingt vom Kernteam eines Projektes erstellt werden. - Kriterien sind unter Umständen nicht lange genug sichtbar.

Tabelle 27: Vor- und Nachteile des Vorgehens der frühen oder späten Kriterienerstellung

7.1.3 Systematik des Vorgehens bei der Kriterienerstellung

Die Systematik bestimmt, wie vorgegangen werden muss, um Kriterien zu erstellen, zu ergänzen, auszuwählen und zu operationalisieren. Eine zielorientierte Herleitung von Maßen und Messplänen mittels der GQM-Methode oder über Qualitätsmodelle stellt das Optimum dar, ist allerdings aufwändig. Durch das zielorientierte Vorgehen wird allerdings am ehesten eine breite Abdeckung der Qualitätsziele durch Kriterien sichergestellt.

Bei einem festen Kriterienkatalog ohne Möglichkeit zur Ergänzung ist der Aufwand für ein zielorientiertes Vorgehen vertretbar, da die Kriterien nur einmalig erstellt werden. Bei der Erstellung ist darauf zu achten, dass die Kriterien so erstellt werden, dass sie für viele Projekte gelten. Kriterien sind daher gegebenenfalls etwas abstrakter zu formulieren und im Rahmen eines Projektes zu operationalisieren.

Werden Kriterien für jedes Projekt individuell erstellt oder besteht die Möglichkeit zur Ergänzung um Kriterien, so ist ein zielorientiertes Vorgehen vermutlich zu aufwändig. Diese Überlegung gilt ebenfalls für die Operationalisierung von Kriterien.

Weniger aufwändig gestaltet sich das erfahrungsbasierte Vorgehen. Hierbei werden durch Experten nur auf Grundlage ihres Erfahrungswissens Kriterien erstellt, ergänzt, ausgewählt oder operationali-

siert. Dieses Vorgehen liefert gute Kriterien, allerdings auch eine schlechtere Abdeckung der Qualitätsziele und ungeeignete Kriterien (vgl. Abschnitt 13.2.2).

Am wenigsten aufwändig gestaltet sich das intuitive Vorgehen. Hierbei werden die Kriterien durch Personen erstellt, ergänzt, ausgewählt oder operationalisiert, die nur über wenig Erfahrungswissen verfügen. Ungeeignete Kriterien und eine schlechte Abdeckung der der Qualitätsziele sind hierbei wahrscheinlich.

Die Kriterienerstellung kann risikobasiert (unabhängig von der Wahl der Systematik) erfolgen. Das heißt, dass Risiken zuerst bestimmt werden und daraus Kriterien mit Forderungen entwickelt werden, die diesen Risiken präventiv entgegenwirken. Dies bedeutet gleichzeitig, dass nur die wichtigsten Risiken durch Kriterien abgedeckt werden können, da nur begrenzt Ressourcen vorhanden sind. Wird ein solches Kriterium nicht erfüllt bedeutet dies, dass das Risiko wahrscheinlich eintritt. Nachteilig ist, dass eine risikobasierte Kriterienerstellung mehr Ressourcen benötigt.

Nach Pfeifer [Pfeifer et al. '03] sollte immer ein strukturierter Ablauf zur Kriterienerstellung befolgt werden, gleich welches der drei Vorgehen gewählt wird:

1. **Beteiligte Ersteller identifizieren:** Typische Ersteller sind interne oder externe Kunden und Lieferanten der Ergebnisse, sowie Rollen der Projekt- und Qualitätsmanagement-Hierarchie (insbesondere Projektleiter und Qualitätsbeauftragte).
2. **Identifizierte Ersteller einladen:** Hierbei lädt der Gate-Verantwortliche die identifizierten Ersteller zu einer gemeinsamen Sitzung. Idealerweise sollten die Ersteller im Vorfeld über den Ablauf der Kriterienerstellung informiert werden.
3. **Kriterien konkretisieren:** Hierbei werden durch die Ersteller (idealerweise unter Leitung des Gate-Verantwortlichen) Kriterien erstellt, operationalisiert, ergänzt und ausgewählt.
4. **Kriterien operationalisieren:** Werden Kriterien sehr früh erstellt, so sind die Kriterien in der Planungs- oder Durchführungsphase durch die Ersteller oder Gutachter zu operationalisieren.

Tabelle 28 fasst die Vor- und Nachteile des systematischen, intuitiven und unstrukturierten Vorgehens bei der Kriterienerstellung zusammen.

+/-	Zielorientiertes Vorgehen	Erfahrungsbasiertes Vorgehen	Intuitives Vorgehen
Vorteile	- Liefert gute Abdeckung der Qualitätsziele.	- Weniger zeitlich aufwändig. - Liefert gute Kriterien.	- Am wenigsten aufwändig.
Nachteile	- Zeitlich und personell aufwändig.	- Abdeckung der Qualitätsziele schlechter als beim zielorientierten Vorgehen. - Benötigt sehr viel Erfahrung. - Liefert auch triviale oder unnötige Kriterien. - Personell aufwändig.	- Orientierung der Kriterien an dem, was leicht zu messen ist, aber wahrscheinlich ungeeignet ist. - Vermutlich schlechte Abdeckung der Qualitätsziele.

Tabelle 28: Vor- und Nachteile der systematischen, intuitiven und unstrukturierten Vorgehens

7.2 Ersteller

Die Ersteller der Kriterien sind für die Erstellung, Ergänzung, Auswahl und Operationalisierung von Kriterien verantwortlich. Ein fester Kriterienkatalog, dessen Kriterien für viele Projektsituationen gelten sollen, erfordert eine sorgfältige Konzipierung. Im Rahmen der Ausgestaltung wird diese Erstel-

lung vom Prozessmanagement übernommen. Die Pflege des ausgestalteten Kriterienkataloges fällt später in den Aufgabenbereich des Gate-Managements.

Die Aufgabe der Operationalisierung, Ergänzung und Auswahl und bei fehlendem Kriterienkatalog auch die Erstellung ist für jedes Projekt individuell. Sie sollte abhängig vom Zeitpunkt von verschiedenen Rollen durchgeführt werden (vgl. Tabelle 29). Idealerweise sind Rollen der Projekt- und Qualitätsorganisation anwesend, durch die das richtige Maß an Kriterien ausgehandelt werden kann. Fachliche und inhaltliche Unterstützung kann dabei durch Lieferanten und Kunden erfolgen. Die Identifikation von Kunden und Lieferanten ist jedoch erst in oder mit Abschluss der Planungsphase möglich, da erst dann die Ergebnisse auf Basis der Arbeitspakete feststehen und somit auch Kunden und Lieferanten benannt werden können.

Kunden und Lieferanten können entweder intern oder extern sein, je nachdem ob sie zum Projekt selbst gehören oder außerhalb des Projektes stehen. Kunden arbeiten mit den Produkten weiter. Zum Beispiel ist ein Programmierer ein interner Kunde, der auf den erstellten Entwurf angewiesen ist. Ein Software-Architekt ist diesem Fall der interne Lieferant des Entwurfs. Externe Lieferanten und Kunden sind keine Projektmitglieder. Sie können entweder Teil eines anderen Projektes im gleichen Unternehmen sein oder zu einem anderen Unternehmen gehören. Interne Kunden sind deutlich eher gewillt an der Kriterienerstellung mitzuwirken als externe Kunden, da interne Kunden viel stärker in ein Projekt involviert sind und das Management die Teilnahme an der Kriterienerstellung anordnen kann.

Im Rahmen der Ausgestaltung ist generell zu entscheiden, ob externe Kunden aktiv an der Kriterienerstellung mitwirken sollten, da sie schließlich dafür bezahlen, dass ein Unternehmen die Qualitätssicherung selbst organisiert. Daher kann der Willen zur Bereitschaft an einer aktiven Teilnahme seitens des Kunden gering ausfallen. In diesem Fall werden jedoch meistens durch den Kunden abstrakte Qualitätsziele vorgegeben.

Die Identifikation der notwendigen Rollen sollte durch das Projektkernteam, Planungsteam oder später durch den Projektleiter und Qualitätsbeauftragten erfolgen. Die identifizierten Rollen können jedoch auch durch die Ausgestaltung fest vorgegeben werden. So kann beispielsweise festgelegt werden, dass stets der leitende Programmierer bei der Operationalisierung der Kriterien für das Quality Gate „Fertig für Implementierung“ als interner Kunde beteiligt sein muss. Bei der Identifikation der notwendigen Rollen sollten die folgenden Regeln beachtet werden:

- Die Gruppe der Ersteller sollte nur wenige Personen umfassen. Jede Rolle ist möglichst durch wenige Personen zu besetzen. Dies erhöht die Effizienz der Kriterienerstellung. Zentrale Rollen, wie zum Beispiel leitende Softwarearchitekten und leitende Programmierer im Projekt und Rollen des Projekt- und Qualitätsmanagements sind bevorzugt zu wählen.
- Das Profil identifizierter Rollen sollte zu den Erfordernissen des jeweiligen Quality Gates passen, für das die Kriterien erstellt, operationalisiert, ergänzt oder ausgewählt werden sollen. Ein leitender Tester als Ersteller im Quality Gate „Anforderungen spezifizieren abgeschlossen“ kann sinnvoll sein, wenn die Anforderungen hinsichtlich ihrer Testbarkeit zu prüfen sind. Falls dies nicht der Fall ist, so ist die Rolle wahrscheinlich überflüssig.
- Die Beteiligung des Gate-Verantwortlichen an der Kriterienstellung unterstützt die Auswahl der richtigen Kunden und Lieferanten und den Ablauf der Kriterienerstellung. Die Kehrseite hierbei ist, dass die Beteiligung einer weiteren Person weitere Kosten verursacht.
- Im Gate-Review selbst kann die Erstellung, Ergänzung, Auswahl und Operationalisierung von Kriterien nur durch Gutachter oder Gatekeeper erfolgen.

Tabelle 29 fasst die ideale Besetzung der Ersteller je nach Zeitpunkt im Projekt zusammen.

Ausgestaltung (fester Kriterien- katalog)	Projektstart	Planungsphase	Durchführungs- phase (im Vorfeld eines Quality Gates)	Durchführungs- phase (im Gate-Review)
Gate- Management mit Fachexperten	Projektkernteam mit Fachexperten	Planungsteam mit identifizierten Kunden, Liefe- ranten und Fach- experten	Projektleiter, Qualitätsbeauf- tragter und identi- fizierte Kunden, Lieferanten und Fachexperten	Gutachter oder Gatekeeper

Tabelle 29: Ideale Besetzung der Ersteller nach Zeitpunkten innerhalb eines Projektes

7.3 Kriterien

Qualitätsbezogenen Kriterien, die auf Produkte (z.B. auf die Anforderungsspezifikation) oder Kennzahlwerte (z.B. auf die Testabdeckung) angewendet werden und aus definierten Qualitätszielen abgeleitet worden sind, stehen im Fokus eines Quality Gates. Sie bilden die primären Kriterien. Diese Kriterien lassen sich in zwei Gruppen unterteilen:

- Kriterien, die das Vorhandensein eines Produktes zum Quality Gate fordern.
- Kriterien, die bestimmte Sollwerte an die Ergebnisse stellen.

Da im Rahmen eines Quality Gates der qualitative Status eines Projektes dem Management gegenüber sichtbar gemacht wird, ist es auch sinnvoll, gleichzeitig weitere Information über das Projekt sichtbar zu machen und durch weitere (nicht qualitätsbezogene) Kriterien zu prüfen – diese Kriterien zählen zu den sekundären Kriterien. Durch Prüfung dieser Kriterien verliert das Quality Gate jedoch seinen qualitätsorientierten Fokus und wird zum generischen Entscheidungspunkt. Hierbei besteht die Gefahr, dass die qualitätsorientierte Ausrichtung vollständig in den Hintergrund tritt.

Sekundäre Kriterien betreffen nach Hawlitzky [Hawlitzky '02] und Cooper [Cooper '01] die folgenden Aspekte:

- **Projektfortschritt:** Wie sieht die Kosten- und Termsituation eines Projektes aus? Zur Bewertung bietet sich zum Beispiel das EVA-Kennzahlensystem an [Hindel et al. '06].
- **Marktattraktivität:** Sind die Vermarktungsmöglichkeiten (weiterhin) positiv? Ist der Markt wachsend und zukunftssträftig?
- **Wert für den Kunden:** Sind die Vorzüge der Software klar herausgearbeitet worden? Setzt sich die Software von ähnlichen Produkten ab? Wird der Kunde die Software kaufen?
- **Technische Risiken:** Ist die Software technisch realisierbar? Kann das Risiko durch den Einkauf von Fertigungskomponenten abgeschwächt werden (Make-or-Buy-Entscheidung)?
- **Risiken des Marktes:** Können die Vorteile der Software vermittelt werden?
- **Return on Investment:** Ist der voraussichtliche Gewinn hoch genug?
- **Nichttechnische Risiken:** Welche weiteren Risiken bestehen (z.B. rechtliche Aspekte)?
- **Übereinstimmung mit der Unternehmensstrategie:** Inwieweit passt die Software in das Portfolio?

Die Prüfung dieser Kriterien erscheint sinnvoll, wenn die dazu notwendigen Kennzahlwerte ohnehin ermittelt werden. In diesem Fall ist der Mehraufwand der Prüfung im Gate-Review eher gering, wobei jedoch entsprechende Gutachter und Gatekeeper einzuladen sind und an der Kriterienerstellung weitere Fachexperten beteiligt werden müssen. Müssen diese Kennzahlwerte jedoch ermittelt werden, obwohl sie sonst nicht ermittelt und benötigt werden, so sollte eine Prüfung dieser sekundären Kriterien überdacht werden. Werden Projekte anhand primärer und sekundärer Kriterien gleichzeitig bewertet, so können komplexe Bewertungsverfahren wie die beispielsweise das Balanced-Scorecard-Konzept

[Kaplan et al. '92] vorteilhaft eingesetzt werden. Derartige Verfahren betrachten und bewerten eine Organisation oder ein Projekt aus verschiedenen Perspektiven (vergleiche obige Aufzählung). Da in dieser Arbeit ausschließlich die qualitätsorientierte Ausrichtung des Quality-Gate-Konzeptes betrachtet wird, werden derartige Verfahren jedoch nicht weiter betrachtet.

Die Unterscheidung nach Muss- und Sollkriterien (vgl. Abschnitt 4.6.1.4) ist nur dann sinnvoll, wenn Projekte hinsichtlich ihrer Attraktivität miteinander verglichen werden sollen. Sie spielen im Rahmen dieser Arbeit keine Rolle. Im Fokus dieser Arbeit liegt lediglich eine Betrachtung der qualitätsbezogenen Kriterien. Abbildung 73 zeigt primäre Maße, sekundäre Maße und nicht geeignete Maße, die für Kriterien genutzt werden können. Zulässig sind nur solche Kriterien, die sich auf primäre und sekundäre Maße beziehen.

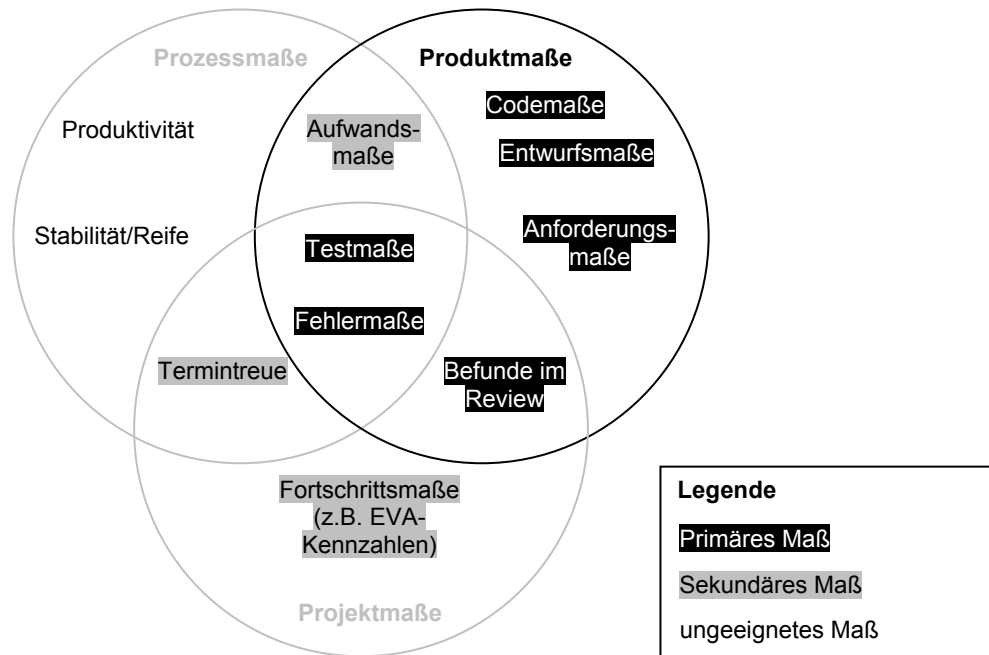


Abbildung 73: Eignung verschiedener Maße für die Bildung von Kriterien

Operationalisierung von Kriterien

Das Ziel der Operationalisierung als Teil der Kriterienerstellung ist es, Kriterien soweit wie möglich anwendbar zu gestalten. Dazu gehört unter anderem, die Berechnungsvorschrift des Maßes, den Sollwert und das Messobjekt möglichst genau zu benennen. Ein „schwammig“ formulierter Sollwert kann beispielsweise zu einer willkürlichen Bewertung im Gate-Review führen. Ideale Kriterien sind eindeutig zu formulieren und lassen möglichst keinen Spielraum zur Interpretation.

Weiterhin ist es wichtig, eine Messvorschrift zu definieren, die angibt wie, wann und durch wen das jeweilige Maß auf das Messobjekt anzuwenden ist. Auf diese Weise wird die Messung operationalisiert. Findet die Messung (also die Prüfung) in der Sitzung im Gate-Review selbst statt, so ist idealerweise im Vorfeld der Sitzung die notwendige (IT-)Infrastruktur aufzusetzen, die eine effiziente und effektive Messung innerhalb der Sitzung ermöglicht (vgl. Abschnitt 13.2.3).

Die Gewichtung eines Kriteriums klärt, inwieweit die Nichterfüllung des jeweiligen Kriteriums als kritisch angesehen werden kann oder einfacher ausgedrückt, wie wichtig das Kriterium für ein Projekt ist. Die Gewichtung wird ferner für eine eventuell vorhandene Entscheidungsunterstützung benötigt (vgl. Abschnitt 9.3) und erleichtert damit die Entscheidungsfindung im Gate-Review.

Bei der Gewichtung sind verschiedene Strategien möglich:

- **Gleichgewichtung:** Jedes Kriterium wird gleich gewichtet. Wenn eine definierte Anzahl oder ein definierter Prozentsatz von Kriterien nicht erfüllt wurde, besteht Handlungsbedarf.
- **Gewichtung auf ordinaler Skala (Ampelgewichtung):** Jedes Kriterium wird auf einer ordinalen Skala bewertet. Zum Beispiel kann ein Kriterium als „rot“, „orange“ oder „gelb“ bewert-

tet werden, wobei eine rote Gewichtung deutlich stärkeren Handlungsbedarf signalisiert, als eine orange oder gelbe Gewichtung.

- **Relative Gewichtung:** Die Gewichtung aller Kriterien ergibt zusammen 100%. Wichtigere Kriterien haben eine höhere Gewichtung. Ist ein gewisser Prozentsatz (z.B. 25%) nicht erfüllt, so ist dies ein Signal für ein zu hohes Risiko. Es sind dann auf jeden Fall Maßnahmen zu treffen.

Die Gewichtung stellt einen Teil des Risikomanagements dar, wenn die Gewichtung der Kriterien risikobasiert erfolgt. Wichtigere Kriterien implizieren bei Nichterfüllung schwerwiegendere Risiken.

Für die Ermittlung der Gewichtung bietet sich der folgende systematische Prozess an, der im Rahmen dieser Arbeit entwickelt wurde:



Abbildung 74: Teilprozess zur Gewichtung von Kriterien

Im Folgenden werden die einzelnen Aktivitäten zur Gewichtung der Kriterien beschrieben:

1. **Identifikation der Risiken:** Hierbei werden die Risiken ermittelt, die bei Nichterfüllung der Kriterien bestehen. Für jedes Kriterium k_i wird hierzu die Menge von möglichen Risiken $M_{\text{risk},i}$ bestimmt. Idealerweise ist die Kriterienerstellung schon risikobasiert erfolgt, so dass zumindest ein Teil der Risiken bereits bekannt ist. Weitere Risiken können sich aus den Besonderheiten des Projektes ergeben.
2. **Bewertung der Risiken:** Für jedes Kriterium k_i wird jedes Risiko $r \in M_{\text{risk},i}$ anhand eines Risikoportfolios oder über die Risikokennzahl hinsichtlich der Schadenshöhe und der Eintretenswahrscheinlichkeit bewertet.
3. **Ableitung der Gewichtung:** Aus jeder Menge von bewerteten Risiken $M_{\text{risk},i}$ wird eine Gewichtung des Kriteriums k_i ermittelt. Dabei gilt, dass Risiken mit einer höheren Risikokennzahl oder kritischeren Bewertung im Risikoportfolio auch zu einer stärkeren Gewichtung der zugehörigen Kriterien führen. Idealerweise existieren Richtlinien, die die Ableitung der Gewichtung eines Kriteriums aus der Bewertung der Risiken erleichtern.

Dieses systematische Vorgehen zur Gewichtung von Kriterien gestaltet sich vergleichsweise aufwändig. Es sollte daher nur in wichtigen Projekten sehr strukturiert durchgeführt werden. In weniger wichtigen Projekten sollten die Kriterien gleich gewichtet werden oder auf einer Ampelskala intuitiv durch die Ersteller gewichtet werden.

Im Folgenden wird das systematische Vorgehen zur risikobasierten Gewichtung anhand eines Beispiels illustriert.

Beispiel: Im Quality Gate *Fertig für Entwurfsphase* werden unter anderem die folgenden fünf Kriterien geprüft:

1. „Die Anforderungen in der Anforderungsspezifikation müssen vom Kunden priorisiert sein.“
2. „Die Anforderungsspezifikation muss eine Versionsnummer und ein Erstellungsdatum besitzen.“
3. „Angaben zum Ablauf der Anforderungsanalyse (Herkunft, wahrgenommene Kundentermine, etc.) müssen in der Anforderungsspezifikation vorhanden sein.“
4. „Ein Glossar muss vorhanden sein, das wichtige Fachbegriffe und Abkürzungen erklärt.“
5. „Das Layout der Anforderungsspezifikation muss konsistent sein.“

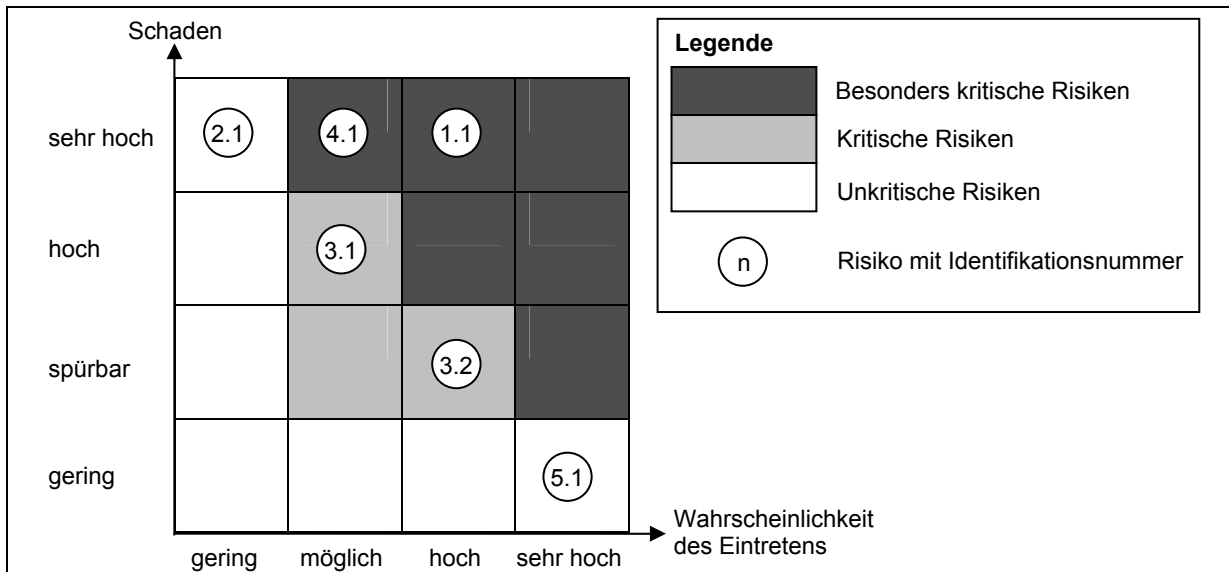
Zunächst werden für jedes Kriterium die Risiken identifiziert, die bei Nichterfüllung jeweils bestehen. Dazu werden die Mengen der Risiken $M_{\text{risk},i}$ für jedes Kriterium k_i bestimmt:

Kriterium	Menge von Risiken
k_1	1.1 Wenn Anforderungen unterschiedlich wichtig sind, dann erhält der Kunde ein Produkt, das nicht seinen Wünschen entspricht, weil wichtige Funktionen aufgrund von Zeitmangel fehlen.
k_2	2.1 Wenn eine falsche/ältere Version der Anforderungsspezifikation genutzt wird, dann fehlt Funktionalität oder ist überflüssig.
k_3	3.1 Wenn Stakeholder nicht berücksichtigt wurden, dann fehlen Anforderungen. 3.2 Wenn Anforderungen mit unklarer Herkunft in die Spezifikation fließen, dann können Anforderungen nicht zurückverfolgt und eventuell hinterfragt werden.
k_4	4.1 Wenn Begriffe inkonsistent verwendet werden, dann kommt es zu Missverständnissen und zu falsch umgesetzter Funktionalität.
k_5	5.1 Wenn das Layout nicht konsistent ist, dann ist die Anforderungsspezifikation schwerer zu lesen.

Im nächsten Schritt werden die Risiken mittels eines Risikoportfolios bewertet, indem sie durch die Ersteller hinsichtlich ihrer Eintretenswahrscheinlichkeit und ihrer potentiellen Schadenshöhe eingeordnet werden. Dabei ergeben sich drei Risikobereiche:

- Besonders kritische Risiken (dunkelgrauer Bereich) sind besonders zu beachten, da sie höchstwahrscheinlich eintreten und dann erheblichen Schaden verursachen.
- Kritische Risiken (hellgrauer Bereich) sind zu beachten, da sie wahrscheinlich eintreten und Schaden verursachen.
- Unkritische Risiken (weißer Bereich) sind weniger zu beachten, da sie entweder wahrscheinlich nicht eintreten oder nur geringen Schaden verursachen.

Die folgende Abbildung zeigt ein Portfolio mit den bewerteten Risiken.



Es seien nun die folgenden Richtlinien zur Ableitung der Gewichtung angenommen: Falls ein Kriterium mit mindestens einem Risiko im dunkelgrauen Bereich verbunden ist, so ist das Kriterium mit *rot* zu gewichten. Ist das Kriterium mit mindestens einem Risiko im hellgrauen Bereich verbunden, aber mit keinem im dunkelgrauen Bereich, so ist das Kriterium mit *orange* zu gewichten. Entsprechend wird ein Kriterium mit *gelb* gewichtet werden, wenn es mit mindestens einem Risiko im weißen Bereich verbunden ist, aber mit keinem Risiko aus dem hell- oder dunkelgrauen Bereich. Auf diese Weise kann eine Ampelgewichtungs-Strategie umgesetzt werden.

Alternativ kann auch über die Risikokennzahl gewichtet werden. Als Ausgangspunkt werden dabei die möglichen Eintretenswahrscheinlichkeiten und Schadenshöhen genutzt, die in Tabelle 4 und Tabelle 5 definiert werden. Für jedes Risiko müssen die Schadenshöhe und die Eintretenswahrscheinlichkeit genau bestimmt werden. Durch die Bildung der Risikokennzahlen kann die Strategie „Relative Gewichtung“ umgesetzt werden, wenn die Summe aller Risikokennzahlen als 100% angesehen wird.

Kriterium	Gewichtung über Farben	Gewichtung über Risikokennzahl
1	rot	$r = p \cdot d = 0,7 \cdot 9 = 6,3$
2	gelb	$r = p \cdot d = 0,1 \cdot 10 = 1$
3	orange	$r = p \cdot d = 0,4 \cdot 6 = 2,4$
4	rot	$r = p \cdot d = 0,5 \cdot 7 = 3,5$
5	gelb	$r = p \cdot d = 0,95 \cdot 1 = 0,95$

Beispiel 2: Beispiel für die Gewichtung von Kriterien

Ein weiterer Aspekt, der durch die Operationalisierung konkretisiert wird, ist der Indikator eines Kriteriums. Ein Indikator visualisiert ein Maß durch die Verdichtung von Rohdaten [Assmann et al. '02]. Indikatoren dienen dazu, Messwerte so darzustellen, dass eine spezielle Rolle (oder auch Person) leicht notwendige Informationen gewinnen kann. Im Sinne der Modelltheorie ist ein Indikator ebenfalls ein Modell. Er ist daher so zu bilden, dass er für jemanden nutzbringend ist. Abbildung 75 zeigt ein Beispiel für einen Indikator, der verschiedene Messwerte der Testüberdeckung visualisiert.

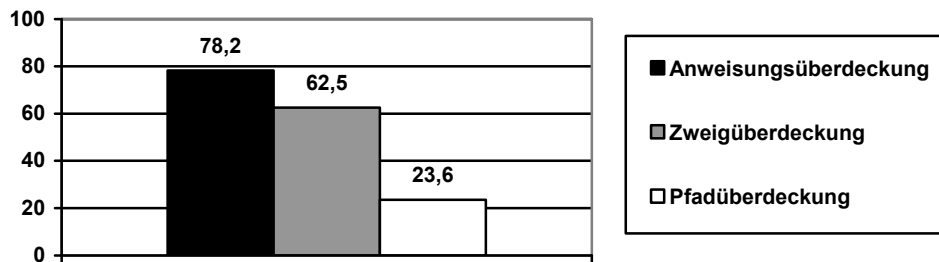


Abbildung 75: Beispiel für einen Indikator für verschiedene Testüberdeckungen

Ein derartiger Indikator erleichtert die Bewertung der Kriterien für die Gutachter, ist jedoch für die Gatekeeper wahrscheinlich schon zu technisch und nicht verdichtet genug. Häufig reicht ihnen die Darstellung als „Ampel-Indikator“, der den Status eines Quality Gates oder die Erfüllung eines Kriteriums mit den Zuständen „rot“, „gelb“ oder „grün“ visualisiert. Es sind aber auch mehr Ampel-Zustände mit feineren Abstufungen denkbar. Im Falle eines dreistufigen Ampel-Indikators signalisiert die Farbe „rot“ unbedingten Handlungsbedarf, „gelb“ legt die Einleitung von Maßnahmen nahe, „grün“ signalisiert, dass ein Projekt oder Kriterium keine größeren Probleme aufweist.

7.4 Ergebnisse

Nach dem abstrahierenden Domänenmodell, kann ein Ergebnis entweder ein Produkt oder ein Kennzahlwert sein. Kennzahlwerte sind dabei nichts anderes als Messwerte, die bereits im Vorfeld eines Quality Gates ermittelt und im entsprechenden Gate-Review gegen die Kriterien geprüft werden. Produkte können auch im Gate-Review vermessen werden, um so die notwendigen Kennzahlwerte zu bestimmen. Grundsätzlich kann jeder Kennzahlwert entweder im Gate-Review oder im Vorfeld eines Quality Gates bestimmt werden.

Kandidaten für Kennzahlwerte, die im Vorfeld des Quality Gates ermittelt werden sollten, sind die folgenden:

- Kennzahlwerte, die aufwändig zu bestimmen sind. Insbesondere also Prozess- und Projektkennzahlen.
- Kennzahlwerte, die nicht nur für ein Quality Gate benötigt werden.

Kandidaten für Kennzahlwerte, die im Gate-Review berechnet werden sollten, sind die folgenden:

- Kennzahlwerte, die nur für das betroffene Quality Gate benötigt werden.
- Kennzahlwerte, die sich auf Grundlage eines geforderten Produktes unaufwändig berechnen lassen.

7.5 Beispiel für eine Ausgestaltung der inhaltlichen Konzepte

In diesem Abschnitt wird anhand eines Beispiels gezeigt, wie die Kriterienerstellung in einem Unternehmen erfolgen könnte.

Ein mittelständiges Unternehmen möchte die Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* umsetzen, um ein einheitliches Mindestniveau an Qualität in allen Projekten zu sichern und einen Fortschrittsvergleich zwischen Projekten zu ermöglichen. Das Prozessmanagement des Unternehmens entscheidet sich dafür, einen festen Kriterienkatalog erstellen. Von der Möglichkeit zur Auswahl von Kriterien wird abgesehen, da die Bewertung des Projektfortschritts wichtiger ist, als auf die besonderen Bedürfnisse eines Projektes eingehen zu können. Von der Möglichkeit zur Ergänzung um Kriterien wird ebenfalls abgesehen, da hierfür der Aufwand zu hoch wäre.

Die erste Aufgabe des Prozessmanagements besteht darin, einen Kriterienkatalog zu erstellen. Da dies eine einmalige Aufgabe ist und im späteren Verlauf der Kriterienkatalog erfahrungsbasiert gepflegt werden soll, wählt das Prozessmanagement hierfür das zielorientierte Vorgehen. Das zielorientierte Vorgehen ist zwar mit einem erhöhten Aufwand verbunden, es lässt sich jedoch bei einmaliger Durchführung (wie in diesem Fall) rechtfertigen. Eine spätere Operationalisierung, die für jedes Projekt notwendig ist, findet hingegen erfahrungsbasiert und nicht beispielsweise über Qualitätsmodelle statt. Auf diese Weise kann der Aufwand verringert werden.

Vor der Ausgestaltung des Kriterienkataloges befindet sich das Unternehmen in der folgenden Ausgangssituation:

Bei der Wartung entwickelter Software kommt es häufig zu Schwierigkeiten. Entwickler stellen immer wieder fest, dass die Behebung von Fehlern einen erheblichen Teil der Ressourcen verlangt, da die Entdeckung der Fehler sich zumeist als sehr aufwändig gestaltet. Es erscheint daher wünschenswert, die Wartbarkeitseigenschaften in einem Quality Gate „Fertig für Abnahme“ vor der Abnahme analysieren zu können.

Aufgrund dieser Ausgangslage entscheidet sich das Prozessmanagement dafür, die Wartbarkeitseigenschaften einer Software im Quality Gate „Fertig für Abnahme“ zu bewerten. Dazu ist die Software jeweils aus der Perspektive eines Entwicklers oder Designers zu analysieren, der die Software warten soll. Der zielorientierten Vorgehensweise folgend, wendet das Prozessmanagement die GQM-Methode an. Das Prozessmanagement beginnt – wie bei der GQM-Methode vorgesehen – auf der Ebene der Ziele und orientiert sich dabei an einem üblichen Standard für Qualitätsaspekte (in diesem Fall den Qualitätsbaum nach Boehm, vgl. Abbildung 24). Das Prozessmanagement erhält als Ergebnis den folgenden Zielbaum:

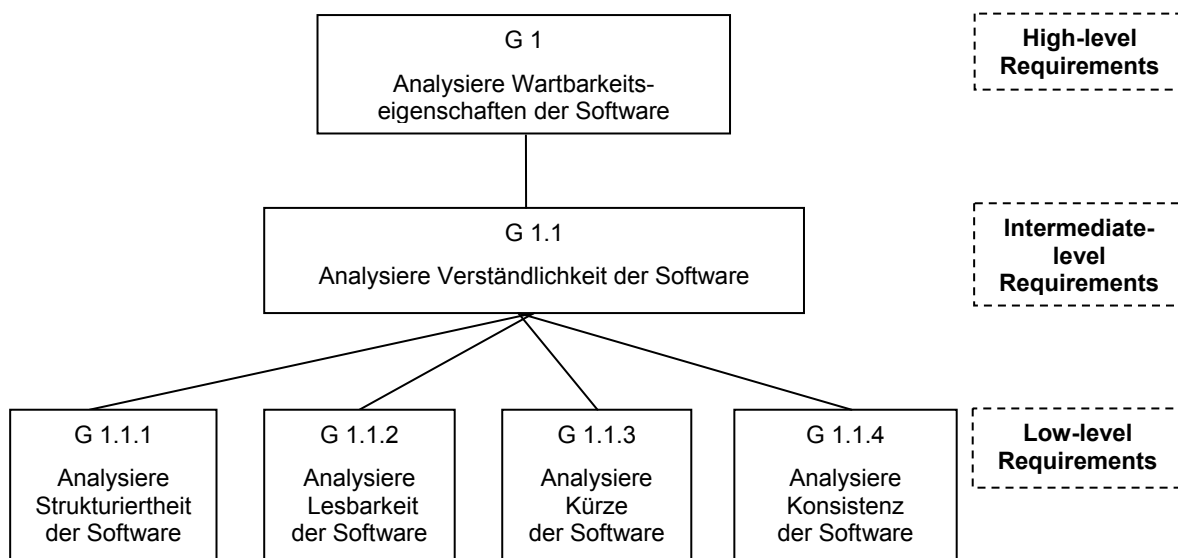


Abbildung 76: Zielbaum für den Qualitätsaspekt Wartung

Im nächsten Schritt leitet das Prozessmanagement Messziele über Facettenbeschreibungen her, da dies die Herleitung von Fragen und letztendlich von Maßen erleichtert. Dazu sind der Zweck, der Qualitätsaspekt, der Betrachtungsgegenstand, die Perspektive und der Kontext für jeden verfeinerten Qualitätsaspekt („Low-level Requirements“) des Zielbaumes zu beschreiben. Das Prozessmanagement erhält die folgenden Facettenbeschreibungen:

Ziel	Zweck	Qualitätsaspekt	Betrachtungsgegenstand	Perspektive	Kontext
1.1.1	Analysiere	Strukturiertheit	Quelltext, Entwurf	Designer und Programmierer, die SW warten sollen	QG „Fertig für Abnahme“
1.1.2	Analysiere	Lesbarkeit	Quelltext, Entwurf	Designer und Programmierer, die SW warten sollen	QG „Fertig für Abnahme“
1.1.3	Analysiere	Kürze	Quelltext	Programmierer, der SW warten soll	QG „Fertig für Abnahme“
1.1.4	Analysiere	Konsistenz	Quelltext	Programmierer, der SW warten soll	QG „Fertig für Abnahme“

Tabelle 30: Facettenbeschreibungen für verschiedene Qualitätsaspekte

Das Prozessmanagement entscheidet sich zunächst dafür, für den Qualitätsaspekt Lesbarkeit geeignete Fragen herzuleiten. Dies erfolgt auf Basis eines so genannten *Abstraction Sheets*. Neben dem Prozessmanagement wirken an der Erstellung des Abstraction Sheets noch ein leitender Software-Designer und leitender Programmierer mit.

Ziel G 1.1.2	Ausgefüllt von: Fritz Meier (Qualitätsmanager)			Datum: 13.8.2006
Zweck	Qualitätsaspekt	Betrachtungsgegenstand	Perspektive	Kontext
Analysiere	Lesbarkeit	Quelltext	Designer und Programmierer, die SW warten sollen	QG „Fertig für Abnahme“
Qualitätsfaktoren (Welche Faktoren definieren den Qualitätsaspekt?) <ol style="list-style-type: none"> Vorhandensein eines Entwurfsdokuments Vorhandensein einer Schnittstellenbeschreibung Kommentierung des Quelltextes Formatierung des Quelltextes Benennung von Bezeichnern im Quelltext 			Einflussfaktoren (Was hat Einfluss auf den Qualitätsaspekt?) <ul style="list-style-type: none"> Werkzeuge zur Erstellung der Schnittstellenbeschreibung Richtlinien Schulungsmaßnahmen Struktur- und Verhaltensdiagramme 	
Ausgangshypothese (Momentane Erwartung bzgl. des Qualitätsaspektes) <ol style="list-style-type: none"> Ein Entwurfsdokument in guter Qualität liegt in allen Projekten vor. Eine Schnittstellenbeschreibung wird nicht in allen Projekten erstellt. Dies erschwert den Überblick über den Quelltext. Der Kommentierungsgrad liegt unter 10%. Die Formatierung der Quelltexte orientiert sich an den unternehmenseigenen Richtlinien. Sie werden fast immer befolgt. Es gibt keine Richtlinien zur Benennung der Bezeichner. Es kommt häufig zu „unglücklichen“ Namen für Bezeichner (keine selbstsprechenden Bezeichner). 			Einflusshypothese (Was beeinflusst wie den Qualitätsaspekt?) <ul style="list-style-type: none"> Für einige Programmiersprachen sind keine geeigneten Werkzeuge verfügbar, die das Erstellen einer Schnittstellenbeschreibung erleichtern. Richtlinien oder Schulungsmaßnahmen zur Kommentierung, Formatierung und Benennung beeinflussen c), d) und e) positiv. Struktur- und Verhaltensdiagramme im Entwurfsdokument erleichtern das Verständnis für den Quelltext. 	

Abbildung 77: Abstraction Sheet für den Qualitätsaspekt Lesbarkeit

Das Prozessmanagement bildet nun Fragen für die Qualitätsaspekte, die in negativer Weise im Feld *Ausgangshypothese* des Abstraktion Sheets aufgefallen sind und verknüpft sie gegebenenfalls mit den Einflussfaktoren:

- **Q1:** Wie gut ist der Quelltext kommentiert?
- **Q2:** Ist eine Schnittstellenbeschreibung vorhanden?
- **Q3:** Wurden Schulungsmaßnahmen hinsichtlich der Kommentierung, Formatierung und zur Benennung von Bezeichnern durchgeführt?
- **Q4:** Sind Richtlinien zur Kommentierung, Formatierung und zur Benennung von Bezeichnern vorhanden?

Für die Frage Q1 erstellt das Prozessmanagement den folgenden Indikator und visualisiert auf diese Weise gleichzeitig die Ausgangshypothese:

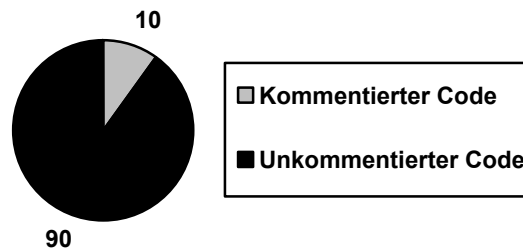


Abbildung 78: Indikator zur Visualisierung der Kommentierung des Quelltextes

Aus den vier Fragen leitet das Prozessmanagement schließlich die folgenden Maße ab:

- **M1:** Kommentierungsgrad des Quelltextes
- **M2:** Vorhandensein einer Schnittstellenbeschreibung
- **M3:** Durchführung von Schulungsmaßnahmen der Programmierer hinsichtlich der Kommentierung, Formatierung und zur Benennung von Bezeichnern
- **M4:** Vorhandensein von Richtlinien zur Kommentierung, Formatierung und zur Benennung von Bezeichnern.

Die letzte Aufgabe des Prozessmanagements besteht nun darin, aus den Maßen Kriterien zu bilden, die für die Analyse der Wartbarkeitseigenschaften im Quality Gate „Fertig für Abnahme“ geeignet sind. Die notwendigen Sollwerte werden durch den beteiligten leitenden Software-Designer und durch den leitenden Programmierer vorgegeben. Das Prozessmanagement erhält die folgenden vier Kriterien:

- **K1:** „Der Quelltext ist ausreichend kommentiert?“
- **K2:** „Eine Schnittstellenbeschreibung ist vorhanden?“
- **K3:** „Alle Mitarbeiter haben an einer Schulung mit dem Thema *Kommentierung, Formatierung und Benennung von Bezeichnern* teilgenommen?“
- **K4:** „Es sind Richtlinien zur Kommentierung, Formatierung und zur Benennung von Bezeichnern vorhanden?“

Die soeben erstellten Kriterien sollen nun in einem konkreten Projekt angewendet werden. Im Rahmen des Projektes „MyWebSchedule“ soll eine Web-Applikation zur Terminabsprache für eine andere Fachabteilung des Unternehmens in der Programmiersprache PHP entwickelt werden.

Eine Operationalisierung einiger Kriterien ist notwendig, da die Kriterien noch nicht konkret genug formuliert sind und daher Spielraum zur Interpretation lassen. Dies bedeutet jedoch nicht, dass eine Operationalisierung für jedes Kriterium notwendig ist. Beispielsweise sind die Kriterien K2 bis K4 so konkret, dass eine Operationalisierung nicht notwendig ist.

Wie bereits oben beschrieben, hat sich das Prozessmanagement darauf festgelegt, die Kriterien erfahrungsbasiert zu operationalisieren. Hierfür ist weniger Zeit einzuplanen, als bei einem zielorientierten Vorgehen. Dennoch ist ein erfahrungsbasiertes Vorgehen vergleichsweise teuer, da Fachleute an der Operationalisierung beteiligt werden müssen. Weiterhin werden die Kriterien bereits während der Planungsphase operationalisiert, um sie rechtzeitig für Lieferanten sichtbar zu machen.

An der Operationalisierung für das Quality Gate „Fertig für Abnahme“ wirken der Qualitätsbeauftragte (als Gate-Verantwortlicher), ein leitender Designer und ein leitender Programmierer des Projektes „MyWebSchedule“ mit. Sie bilden zusammen die Kriterienersteller für das Projekt. Im Rahmen der Operationalisierung muss das Kriterium

„Der Quelltext ist ausreichend kommentiert?“

operationalisiert werden.

Die Kriterienersteller sehen vor allem den Kommentierungsgrad der Methoden und Klassen als entscheidend für eine Bewertung der Wartbarkeitseigenschaften an. Kommentierungen müssen aufgrund der gewählten Programmiersprache in der PHPDoc-Syntax erfolgen. Die Kriterienersteller einigen sich auf die folgenden beiden operationalisierten Kriterien:

- Jede Methode einer Klasse in „MyWebSchedule“ muss einen Methodenkommentar haben, der die Aufgabe, die Funktionsweise, Ein- und Ausgabeparameter beschreibt und in der PHPDoc-Syntax formuliert ist.
- Jede Klasse in „MyWebSchedule“ muss einen Kommentar haben, der die Bedeutung der Klasse beschreibt und in der PHPDoc-Syntax formuliert ist.

Weiterhin legen die Kriterienersteller fest, dass die Gutachter die Messung mit dem Softwarewerkzeug „CommentCounter 2.0“ im Gate-Review durchführen müssen. Hierdurch haben die Kriterienersteller die Messvorschrift festgelegt.

Für beide Kriterien wird jeweils der Indikator angepasst und der Sollwert visualisiert:

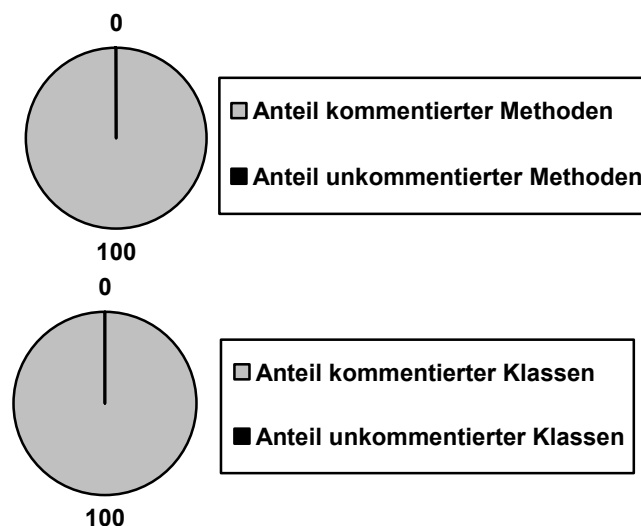


Abbildung 79: Operationalisierte Indikatoren für „Der Quelltext ist ausreichend kommentiert?“

Die Kriterienersteller haben im Rahmen der Operationalisierung des Kriteriums „Der Quelltext ist ausreichend kommentiert?“ folgende Interpretationsaufgaben durchgeführt:

1. Das Kriterium wurde in zwei Kriterien aufgeteilt, um die Handhabung zu erleichtern.
2. Es wurde für jedes der zwei Maße die Berechnungsvorschrift und der Definitions- und Wertebereich angepasst.
3. Die Sollwerte und Messobjekte wurden jeweils festgelegt.
4. Die Messvorschrift wurde jeweils fixiert.
5. Indikatoren wurden operationalisiert.

8 Ausgestaltung der Reviewkonzepte

Reviewkonzepte beschäftigen sich vorrangig mit den Prüfungsaktivitäten, die zu einem Quality Gate stattfinden. Die Ausgestaltung beschäftigt sich damit, wie ein Gate-Review durchgeführt kann und wie die Rollen Gate-Verantwortlicher, Gutachter, Projektvertreter und Protokollant ausgestaltet werden können. Weiterhin ist zu definieren, wie das Teilkonzept *Protokoll* ausgestaltet werden kann. Die nachfolgenden Abschnitte beschäftigen sich mit den Richtlinien und dem Spektrum zur Ausgestaltung dieser Teilkonzepte.

8.1 Gate-Review

Ein Gate-Review definiert einen mehr oder weniger geordneten Prüfungsablauf. Wesentliche Bestandteile dieses Prüfungsablaufes sind eine Prüfung von Ergebnissen gegen Kriterien (die innerhalb der Kriterienerstellung erstellt wurden) und das Fällen einer Entscheidung über die Fortsetzung des Projektes.

Dunham [Dunham '99] stellt verschiedene Anforderungen an ein Gate-Review:

- Das Hauptziel eines Gate-Reviews ist das rasche Treffen einer unmissverständlichen Entscheidung über die Fortsetzung des Projektes.
- Ein Gate-Review sollte das Projekt nicht ausbremsen. Es ist daher möglichst schnell durchzuführen.
- Neben der Prüfung der Kriterien, sollte eine Bewertung der Risiken stattfinden.

Cornish [Cornish '05] stellt weitere Anforderungen an ein Gate-Review:

- Ein Gate-Review sollte möglichst bald nach dem Eintreten eines Quality Gates (auch bei unvollständigen Ergebnissen) durchgeführt werden, um ein Projekt nicht auszubremsen und um zu verhindern, dass Ressourcen fehlgeleitet werden (für den Fall, dass das Projekt größere Schwierigkeiten hat).
- Die Prüfungsaktivitäten orientieren sich ausschließlich an den vereinbarten Kriterien. Die Prüfung ist formal und objektiv. Kein Projekt darf in irgendeiner Weise bevorzugt werden.
- Gutachter und Gatekeeper müssen vorbereitet sein. Insbesondere müssen daher aufwändige Messungen außerhalb des Gate-Reviews stattfinden oder gut vorbereitet sein, so dass sie im Gate-Review leicht durchzuführen sind.

Gate-Reviews sind weit weniger untersucht als technische Reviews im Allgemeinen und die formale Inspektion im Speziellen. Nur wenige Quality-Gate-Referenzprozesse definieren ein systematisches Gate-Review. Die wenigen Quality-Gate-Referenzprozesse, die ein systematisches Vorgehen für das Gate-Review definieren [Cornish '05, '06], gehen im Wesentlichen davon aus, dass sich ein Gate-Review aus Organisationsaktivitäten, Prüfaktivitäten und Abschlussaktivitäten ähnlich dem technischen Review zusammensetzt (vgl. auch Abbildung 47). Organisationsaktivitäten dienen dazu, den reibungslosen Ablauf des Gate-Reviews zu ermöglichen. Prüfaktivitäten dienen der Bewertung der Kriterien und der Entscheidungsfindung. Abschlussaktivitäten dienen dazu, getroffene Maßnahmen zu überwachen und Leistungsdaten zum Gate-Review zu erheben. Die grobe Aufteilung und Abfolge der Unteraktivitäten orientiert sich dabei an den Aktivitäten der formalen Inspektion (vgl. Abbildung 25).

Zu den Organisationsaktivitäten zählen:

- Das Einreichen der Ergebnisse und eine formale Eingangsprüfung. Die formale Eingangsprüfung prüft, ob eine festgelegte Vorbedingung (zum Beispiel, dass zumindest alle Dokumente vorliegen) erfüllt ist. Falls ja, so kann mit den folgenden Aktivitäten fortgefahren werden.

- Eine Vorbereitung der Ergebnisse, die die spätere Prüfung erleichtert. Beispielsweise kann dies eine Aufbereitung in ein leichter zugängliches Format sein.
- Das Aufstellen einer Agenda, die den zeitlichen und inhaltlichen Ablauf und die Personen nennt, die beim Gate-Review anwesend sein müssen. Basierend hierauf erfolgt die Einladung aller beteiligten Personen.
- Die Kriterienerstellung, falls diese nicht schon vorher abgeschlossen wurde. Die Kriterienerstellung ist jedoch im eigentlichen Sinne keine Aktivität eines Gate-Reviews.
- Eine inhaltliche Einführung, in dem der Ablauf, die Prüfung und die Ziele des Quality Gates für die Gutachter erklärt werden.
- Die Verteilung der zu prüfenden Ergebnisse und Checklisten an die Gutachter.

Zu den Prüfaktivitäten zählen:

- Eine individuelle Vorbereitung. Hierbei werden geforderte Ergebnisse auf Grundlage der Kriterien durch die Gutachter im Vorfeld der Sitzung geprüft. Dabei sind im Wesentlichen durch die Durchführung von Messungen die Istwerte und ihre Abweichung zu den Sollwerten zu bestimmen. Die verschiedenen Messungen sind idealerweise auf verschiedene Gutachter aufgeteilt, um die Messung zu beschleunigen. Ferner ist es möglich, die gleiche Messung von verschiedenen Gutachtern durchführen zu lassen, um die Wahrscheinlichkeit von Falschmessungen zu verringern.
- Durchführung einer Sitzung mit dem Ziel der Entscheidungsfindung. Innerhalb der Sitzung kann auch die Prüfung der Ergebnisse stattfinden, wenn zuvor keine individuelle Vorbereitung stattgefunden hat.
- Gegebenenfalls eine Eskalation zum oberen Management, falls eine weitreichende Entscheidung nicht in der Sitzung getroffen werden kann.

Zu den Abschlussaktivitäten zählen:

- Die Verteilung und Archivierung des Protokolls.
- Nachverfolgung von nicht erfüllten Kriterien im Falle der bedingten Projektfortsetzung.
- Erhebung von Leistungsdaten zur Bewertung der Effizienz und Effektivität des durchgeführten Gate-Reviews. Dies umfasst insbesondere die Zeit zur Durchführung der Messungen.

Im Gegensatz zum technischen Review liegt der Hauptaufwand beim Gate-Review nicht bei der Suche nach Befunden und der Diskussion dieser Befunde, sondern vielmehr bei der Durchführung der Messungen und bei der Entscheidungsfindung. Dies liegt vorrangig daran, dass die Messung anhand der (operationalisieren) Kriterien vergleichsweise diskussionslos vollzogen werden kann – zumindest für den Fall, dass die Kriterien bereits vor dem Gate-Review operationalisiert wurden. Das Fällen der Entscheidung erfordert umso mehr Aufwand, da im Falle einer falschen Entscheidung, beachtliche Ressourcen verloren gingen.

Ein systematischer Ablauf, das heißt die Durchführung aller beschriebenen Aktivitäten, ist daher unabdingbar, jedoch nicht in jeder Situation passend. Kleinere Unternehmen oder Unternehmen mit vielen gleichzeitigen Projekten haben ein Interesse daran, den personellen Aufwand so gering wie möglich zu halten und die Entscheidungsträger nur dann mit in das Gate-Review einzubinden, wenn das Projekt besonders wichtig ist oder größere Schwierigkeiten aufweist.

Obwohl ein technisches Review und ein Gate-Review sich hinsichtlich ihrer Zielsetzung unterscheiden, können in Anlehnung an das technische Review (vgl. Tabelle 9) ähnliche Varianten gebildet werden. Für ein technisches Review existieren im Wesentlichen die Varianten: Schreibtischtest, Walkthrough und Inspektion. Der Schreibtischtest ist die günstigste Variante, da keine Sitzung durchzuführen ist. Gleichzeitig ist aber auch die Anzahl der Befunde häufig niedriger. Walkthrough und Inspektion sind aufwändiger, sie erfordern einen längeren terminlichen Vorlauf mit eventueller Vorbereitung und die Teilnahme vieler Personen. Im Gegensatz zum Walkthrough erfordert die formale Inspektion nicht unbedingt eine teure Sitzung [Sapsomboon '99]. Wie bereits in Abschnitt 3.3.4 bemerkt,

ist die Wirkung der Sitzung auf die Anzahl und Qualität der Befunde umstritten. Die Sitzung ist in zweierlei Hinsicht teuer:

- Sie erfordert eine terminliche und räumliche Koordinierung und verzögert dadurch nachfolgende Prozessschritte und somit auch das Projekt.
- Sie involviert Personen, deren eigentliche Arbeit durch die Sitzung unterbrochen wird.

Für das Gate-Review existieren keine Untersuchungen hinsichtlich der Notwendigkeit einer Sitzung. Es kann jedoch davon ausgegangen werden, dass die terminliche Koordinierung sich ebenso verzögernd auf das Projekt auswirkt, wie beim technischen Review. Ferner sollte die Anzahl der Teilnehmer an einer Sitzung möglichst gering gehalten werden.

Grundsätzlich lassen sich die drei Varianten des Reviews auf das Gate-Review übertragen. Weiterhin lässt sich für das Walkthrough und die Inspektion jeweils festlegen, ob die Gruppe der anwesenden Gatekeeper ebenfalls Entscheidungsträger umfasst, die weitreichende Entscheidungen treffen können. Eine aufwändigere Variante ist nicht nur teurer, sondern liefert bei komplexen Projekten auch verlässlichere Ergebnisse, weil zum Beispiel Kriterien von mehr als einem Gutachter geprüft werden können. Ferner kann eine Entscheidung von mehr als einem Gatekeeper getroffen werden und somit die Wahrscheinlichkeit einer Fehlentscheidung verringert werden.

Variante des Gate-Reviews	Beschreibung	Aufwand
Formale Inspektion (mit oder ohne Teilnahme von Gatekeepern mit weitreichenden Befugnissen)	Stark formalisiertes Vorgehen bestehend aus verschiedenen Aktivitäten (vgl. obige Aktivitätsliste). Es sind Varianten ohne und mit individueller vorbereitender Prüfung möglich. Bei individueller Vorbereitung kann eine Sitzung durchgeführt werden oder darauf verzichtet werden. Innerhalb der Sitzung sind Gatekeeper für weniger weitreichende Entscheidungen auf jeden Fall anwesend (vgl. Abbildung 80, Abbildung 81 und Abbildung 82).	Hoch
	Formale Inspektionen ohne individuelle Vorbereitung durch die Gutachter im Vorfeld der Sitzung leiden unter weniger gründlich durchgeführten Messungen, da in der Sitzung selbst nur relativ oberflächlich gemessen werden kann.	Mittel (entweder ohne Sitzung oder ohne individuelle Vorbereitung)
Walkthrough (mit oder ohne Teilnahme von Gatekeepern mit weitreichenden Befugnissen)	Projektvertreter erläutern ihre Ergebnisse auf Grundlage der Kriterien innerhalb einer Sitzung. Gutachter bewerten gegen Kriterien. Gatekeeper für weniger weitreichende Entscheidungen sind auf jeden Fall anwesend. Im Gegensatz zur formalen Inspektion gibt es keine vorbereitende Prüfung. Das Walkthrough eignet sich besonders dann, wenn sehr komplizierte Sachverhalte nur durch die Projektvertreter selbst erläutert werden können.	Mittel
Schreibtischtest	Einzelner Gutachter prüft Ergebnisse gegen Kriterien und erstellt Protokoll. Gegebenfalls fragt er bei den Projektvertretern nach. Eine Sitzung findet nicht statt. Idealerweise kann der Gutachter bereits weniger weitreichende Entscheidungen selber treffen und ist damit gleichzeitig ein Gatekeeper. Die Eskalation zu Gatekeepern mit höherer Entscheidungsbefugnis ist ggf. notwendig.	Niedrig

Tabelle 31: Varianten des Gate-Reviews

Falls keine Gatekeeper anwesend sind, die weitreichende Entscheidungen treffen können, so ist auf jeden Fall festzuhalten, bei welchen Entscheidungen zum höheren Management zu eskalieren ist. Im ungünstigsten Fall muss bei allen Entscheidungen eskaliert werden. Im Standardfall ist hingegen nur beim Abbruch oder der Zurückstellung des Projektes zu eskalieren. Im günstigsten Fall ist dies hinge-

gen nie notwendig, weil die Entscheidung schon von den anwesenden Gatekeepern getroffen werden kann (vgl. auch Abschnitt 9.2).

Nachfolgend ist für die Variante „Formale Inspektion“ des Gate-Reviews jeweils das systematische Vorgehen für die Organisations-, Prüfungs- und Abschlussaktivitäten zu sehen.

Abbildung 80 zeigt den Ablauf der Organisationsaktivitäten. Je nachdem ob die Prüfung innerhalb der Sitzung erfolgt oder bereits im Vorfeld der Sitzung, sind jeweils Gutachter in die Prüfaktivitäten einzuführen und Ergebnisse und Checklisten zu verteilen.

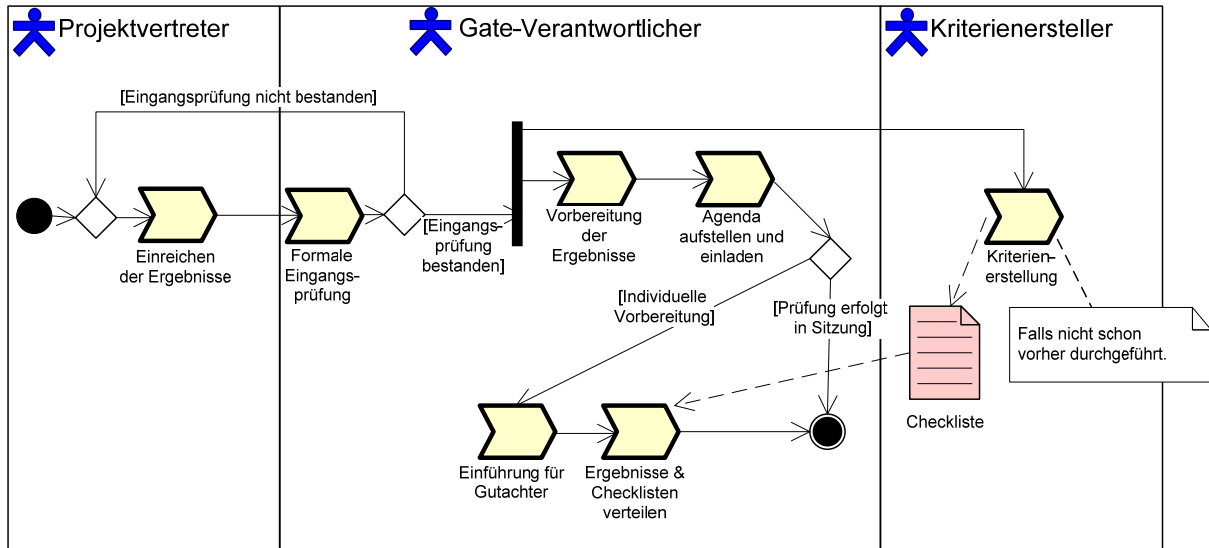


Abbildung 80: Organisationsaktivitäten und Verantwortlichkeiten

Abbildung 81 gibt einen Überblick über die Prüfungsaktivitäten. Hierbei kann Grundsätzlich zwischen einer formalen Inspektion mit und ohne Sitzung unterschieden werden. Die Sitzung ist auf jeden Fall durchzuführen, wenn keine vorherige individuelle Vorbereitung stattfindet, da die Prüfung dann innerhalb der Sitzung selbst stattfinden muss. Bei vorhandener individueller Vorbereitung kann die Sitzung hingegen entfallen. Nachteilig ist dann allerdings, dass eine zentrale Plattform zur Diskussion über die Bewertung der Kriterien und über die Entscheidung fehlt.

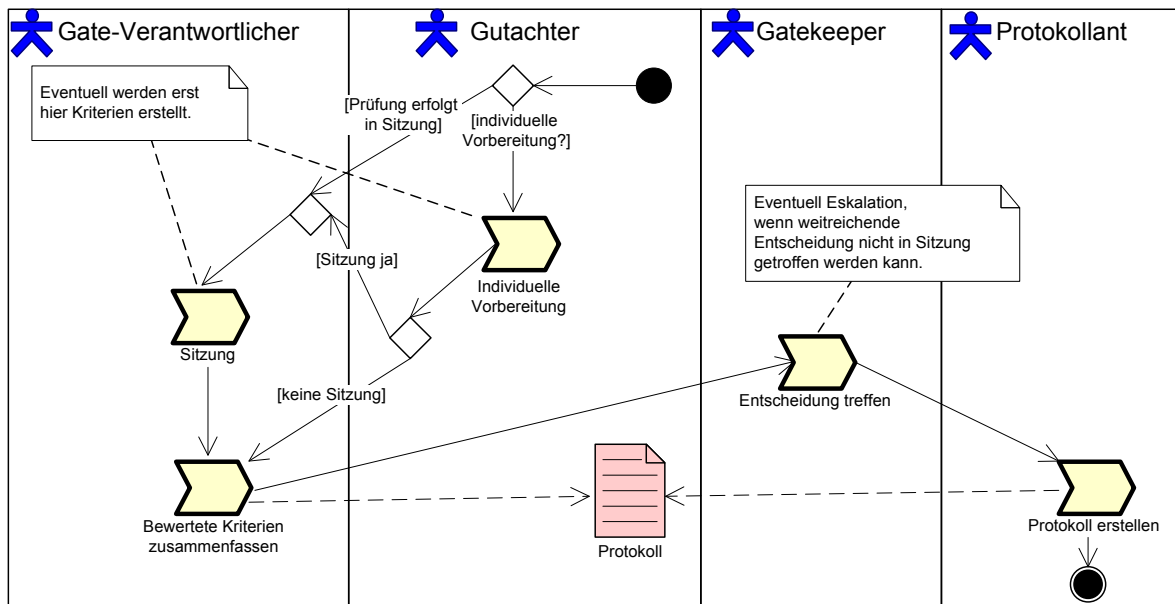


Abbildung 81: Prüfungsaktivitäten und Verantwortlichkeiten

Abbildung 82 zeigt die Abschlussaktivitäten eines Gate-Reviews, das der Variante der formalen Inspektion folgt. Die Durchführung der Aktivitäten liegt in der Verantwortung des Gate-Verantwortlichen.

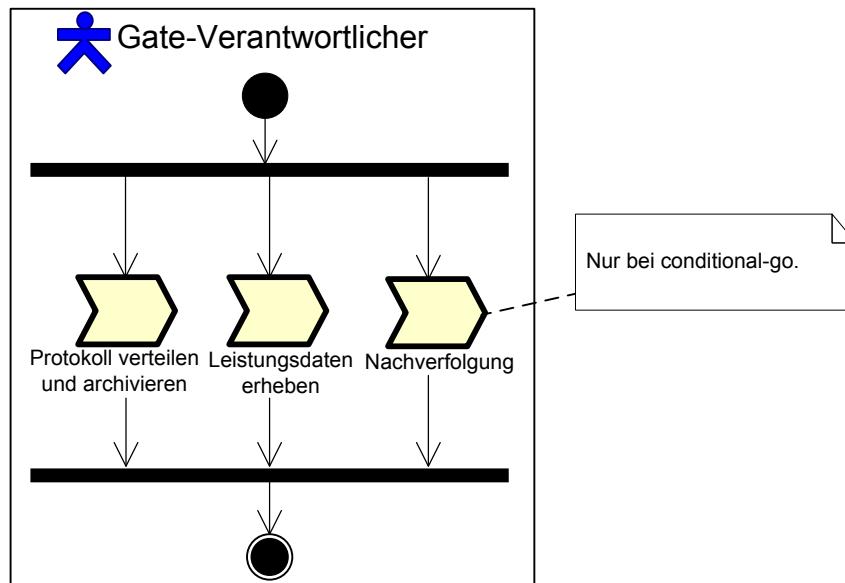


Abbildung 82: Abschlussaktivitäten und Verantwortlichkeiten

Wie beim technischen Review verfügt das Walkthrough als Variante des Gate-Review über keine individuelle Vorbereitung. Eine Einführung der Gutachter entfällt daher. Der einzige Unterschied zur Variante ohne individuelle Vorbereitung besteht darin, dass die Ergebnisse von den Projektvertretern vorgestellt werden.

8.1.1 Risikomanagement im Gate-Review

Nach Abschnitt 3.2 umfasst das Risikomanagement die folgenden Aktivitäten:

- Risikoermittlung (Identifikation, Analyse, Bewertung und Priorisierung von Risiken).
- Die Konzipierung und Einleitung von Gegenmaßnahmen.
- Die Verfolgung von Risiken im Projektverlauf.

Die Risikoermittlung wurde bereits mit der Erstellung der Kriterien und deren Operationalisierung durch die Gewichtung der Kriterien abgeschlossen, falls die Gewichtung risikobasiert durchgeführt wurde. Sie findet damit spätestens im Gate-Review selbst statt (vgl. Tabelle 25).

Die Konzipierung und Einleitung von Gegenmaßnahmen ist eine wesentliche Aktivität eines Gate-Reviews, zumindest für den Fall, dass ein Projekt nicht zurückgestellt oder abgebrochen wird. Die zu treffenden Gegenmaßnahmen wirken dabei vorrangig den Risiken entgegen, die durch nicht erfüllte Kriterien entstehen.

Wie bereits festgestellt, lassen sich Gegenmaßnahmen des Risikomanagements in präventive Maßnahmen und in Notfallmaßnahmen unterscheiden. Zu den präventiven Maßnahmen gehört die schnellstmögliche Erfüllung der Kriterien. Falls dies nicht möglich sein sollte, so sind entsprechende Gegenmaßnahmen zu konzipieren. Die Konzipierung der Gegenmaßnahmen ist ein kreativer Akt, sie hängt vom Risiko, seiner Bewertung und den möglichen Alternativen ab.

Die Wirkung der Gegenmaßnahmen ist zu verfolgen. Diese Aufgabe kann nur bedingt durch Quality Gates erfüllt werden, da die Abstände zwischen den einzelnen Quality Gates zu groß sind, als das eine Überwachung im nächsten Quality Gate noch sinnvoll wäre. Vielmehr sind die Kriterien idealerweise gerade so konzipiert worden, dass sie die Risiken der Folgephase bei Erfüllung möglichst gering halten (Steuerungsaspekt der Kriterien). Beim Erreichen des nächsten Quality Gates wären Risiken mit der Eintretenswahrscheinlichkeit daher längst zu Problemen geworden. Der Status der Risiken, die durch nicht erfüllte Kriterien bestehen, sind daher (je nach Wichtigkeit) in der internen Fortschrittsbesprechung, der formalen Fortschrittsbesprechung oder im Meilenstein-Review zu verfolgen (vgl. Tabelle 3), da sie zwischen den Quality Gates stattfinden.

8.1.2 Überlegungen zum Aufwand eines Gate-Reviews

Der Gesamtaufwand eines Gate-Reviews ergibt sich aus der Summe der Aufwände der Organisations-, Prüfungs- und Abschlussaktivitäten. Es können im Wesentlichen zwei Aufwände unterschieden werden: der zeitliche und monetäre Aufwand. Der monetäre Aufwand kann aus dem zeitlichen Aufwand über die verschiedenen Stundensätze der beteiligten Personen errechnet werden. Im Folgenden wird nur der zeitliche Aufwand betrachtet.

Der Gesamtaufwand ergibt sich aus den Organisations-, Prüfungs- und Abschlussaktivitäten also zu:

$$e_{\text{gesamt}} = e_o + e_p + e_a,$$

wobei e_o der Aufwand für die Organisationsaktivitäten, e_p der Aufwand für Prüfungsaktivitäten und e_a der Aufwand für die Abschlussaktivitäten ist.

Die Aktivitäten sind in den drei verschiedenen Varianten unterschiedlich aufwändig. Weiterhin ist bei der Variante formale Inspektion relevant, ob eine individuelle Vorbereitung oder Sitzung stattfindet. Eine Sitzung oder Einführung für Gutachter ist insbesondere auch deshalb teuer, da hierfür eventuell Personen anreisen müssen. Mitunter kann die An- und Abreisezeit im Mittel der Sitzungs- oder Einführungszeit entsprechen. Gleichzeitig ist zu bedenken, dass die Sitzungszeit bei fehlender individueller Vorbereitung höher ist, da die Messungen nunmehr in der Sitzung stattfinden müssen. Da die Messungen jedoch in diesem Fall wahrscheinlich oberflächlicher verlaufen, wird dieser Effekt vermutlich relativiert.

Umgekehrt muss die Entscheidung ohne Sitzung auf anderem Wege zustande kommen, was den Aufwand ebenfalls erhöht. Die Zeit zur Entscheidungsfindung kann geschätzt im Bereich von 10% der Sitzungszeit liegen.

Ferner ist relevant, durch wie viele Personen die Rollen Gatekeeper, Gutachter und Projektvertreter jeweils besetzt werden. Hingegen kann bei den Rollen Gate-Verantwortlicher und Protokollant davon ausgegangen werden, dass sie mit höchstens einer Person besetzt werden. Beim Schreibtischtest werden alle Rollen (bis auf die des Projektvertreter) von einer Person wahrgenommen.

Die folgende Tabelle zeigt unter den gegebenen Voraussetzungen den geschätzten zeitlichen Aufwand für die verschiedenen Varianten des Gate-Reviews. Die Schätzungen werden dabei anteilig zum Aufwand der jeweiligen Aktivitäten der formalen Inspektion mit individueller Vorbereitung und Sitzung angegeben.

Variante des Gate-Reviews	Organisationsaktivitäten	Prüfungsaktivitäten	Abschlussaktivitäten
Form. Inspektion (Sitzung + ind. Vorbereitung)	$e_o \approx m_g \cdot e_k + e_{org}$	$e_p \approx m_g \cdot e_v + m_s \cdot \frac{6}{10} \cdot e_s$	e_a
Form. Inspektion (nur ind. Vorbereitung)	$e_o \approx m_g \cdot e_k + \frac{1}{2} \cdot e_{org}$	$e_p \approx m_g \cdot e_v + m_s \cdot \frac{2}{10} \cdot e_s$	e_a
Form. Inspektion (nur Sitzung)	$e_o \approx \frac{1}{2} \cdot e_{org}$	$e_p \approx m_s \cdot e_s$	e_a
Walkthrough	$e_o \approx \frac{1}{2} \cdot e_{org}$	$e_p \approx m_s \cdot \frac{3}{2} \cdot e_s$	e_a
Schreibtischtest	$e_o \approx 0$	$e_p \approx \frac{3}{2} \cdot e_v + \frac{1}{10} \cdot e_s$	e_a

Legende

e_k Aufwand für die Einführung der Gutachter (inkl. An- und Abreisezeit)
 e_s Aufwand für Sitzung (inkl. An- und Abreisezeit)
 e_v Aufwand für individuelle Vorbereitung
 e_{org} Aufwand für weitere Organisationsaktivitäten (exkl. Einführung für Gutachter)
 m_g Anzahl der Gutachter + Gate-Verantwortlicher
 m_s Anzahl der Teilnehmer an Sitzung

Tabelle 32: Geschätzter zeitlicher Aufwand für verschiedene Varianten des Gate-Reviews

Die folgende Tabelle zeigt den zeitlichen Aufwand in Personenstunden anhand eines Zahlenbeispiels für verschiedene Varianten des Gate-Reviews.

Variante des Gate-Reviews	Organisationsaktivitäten	Prüfungsaktivitäten	Abschlussaktivitäten	Gesamtaufwand
Form. Inspektion (Sitzung + ind. V.)	6,5	16,2	1	23,7
Form. Inspektion (ind. Vorbereitung)	6,3	11,4	1	18,7
Form. Inspektion (nur Sitzung)	0,3	12	1	13,3
Walkthrough	0,3	18	1	19,3
Schreibtischtest	0	4,7	1	5,7

Zugrunde liegende Zahlen
 $e_k = 2$ h, $e_s = 2$ h, $e_v = 3$ h, $e_{org} = 0,5$ h, $e_a = 1$ h,
 $m_g = 3$ und $m_s = 6$ (2 Gutachter, 2 Gatekeeper, 2 Projektvertreter, Protokoll wird durch Gutachter geschrieben und ein Gatekeeper übernimmt Rolle des Gate-Verantwortlichen)
Angaben in der Tabelle in Personenstunden und auf eine Stelle hinter dem Komma gerundet.

Tabelle 33: Zahlenbeispiel für den zeitlichen Aufwand in Personenstunden

8.2 Gate-Verantwortlicher

Ein Gate-Verantwortlicher ist für die systematische und zügige Durchführung des Gate-Reviews zuständig. Idealerweise sollte er auch die Kriterienerstellung leiten. Da der Gate-Verantwortliche mit den Entscheidungsträgern kommuniziert und unter Umständen auch eskalieren muss, ist der Gate-Verantwortliche durch eine Rolle zu besetzen, die über Entscheidungskompetenz und Autorität verfügt. Das V-Modell XT besetzt die Rolle des Gate-Verantwortlichen beispielsweise mit der Rolle des Projektmanagers, der zumindest eine eingeschränkte Entscheidungsbefugnis über alle Projekte eines Unternehmensbereiches besitzt (vgl. Abbildung 10). Diese Ausgestaltung ist in zweierlei Hinsicht nachteilig:

- Ein Projektmanager ist kein Mitglied der Qualitätsmanagement-Hierarchie. Er hat folglich weniger Interesse an der Qualitätssicherung.
- Die Teilnahme an allen Kriterienstellungen und Gate-Reviews könnte einen Projektmanager überfordern.

Die Besetzung der Rolle des Gate-Verantwortlichen durch die Rolle eines Qualitätsbeauftragten ist mit Hinblick auf die qualitätsbezogene Ausrichtung eines Quality-Gate-Referenzprozesses die bessere Besetzung. Ein Qualitätsbeauftragter besitzt als Qualitätsexperte ein besseres Verständnis für die Kriterienerstellung, da er die Verfahren (namentlich GQM und Qualitätsmodelle) und Begriffe im Bereich des Qualitätsmanagements kennt und über einen eigenen Berichtsweg verfügt, über den eine Eskalation leichter möglich ist (vgl. Abschnitt 3.3.2).

8.3 Gutachter

Die Hauptaufgabe der Gutachter besteht darin festzustellen, welche Kriterien erfüllt sind. Die dazu notwendigen Messungen zur Bestimmung der Istwerte erfordert Fachkenntnisse. Idealerweise besitzen die Gutachter Entscheidungsbefugnisse, wodurch sie ebenfalls die Aufgabe eines Gatekeepers übernehmen können. Auf diese Weise wird die Anzahl der beteiligten Rollen im Gate-Review reduziert. Leider fallen jedoch Entscheidungsbefugnis und tiefgreifende fachliche Kenntnisse eher selten zusammen, so dass die Rollen Gutachter und Gatekeeper in der Regel von verschiedenen Personen wahrgenommen werden müssen.

Ein weiterer Grund, warum die Beurteilung und die Entscheidung von verschiedenen Personen vorzunehmen ist, liegt in einer Entlastung der Entscheidungsträger. Die Messung erfordert Zeit, die nur bedingt durch das Management aufgebracht werden kann.

Idealerweise wird die Rolle des Gutachters durch mehr als eine Person besetzt, da

- dann Messungen von anderen Gutachtern verifiziert werden können
- und die Gutachter entlastet werden.

8.4 Projektvertreter

Die Aufgabe der Projektvertreter besteht darin, für Fragen zur Verfügung zu stehen und gegebenenfalls das Projekt zu verteidigen, falls ein Kriterium „irrtümlicherweise“ als nicht erfüllt bewertet wurde. In der Gate-Review-Variante „Walkthrough“ werden die Ergebnisse durch die Projektvertreter präsentiert.

Da im Gate-Review eine Entscheidung über den Fortgang des Projektes getroffen wird, sollten nur Rollen mit Leitungs- und Entscheidungsbefugnis das Projekt im Gate-Review vertreten. Falls notwendig, können die Projektvertreter durch weitere Personen mit entsprechenden Fachkenntnissen begleitet werden. Im Gegensatz zur technischen Inspektion, sind also die Projektvertreter nicht notwendigerweise identisch mit den Lieferanten (Autoren) der Ergebnisse. Sinnvolle Vertreter sind daher der Projektleiter oder der Qualitätsbeauftragte des Projektes. Idealerweise sind beide Rollen im Gate-Review vertreten. Der Projektleiter sollte vertreten sein, da eine Entscheidung über den Fortgang des Projektes getroffen wird. Der Qualitätsbeauftragte sollte vertreten sein, da das Quality Gate eine qualitätsorien-

tierte Ausrichtung hat (er kann dann auch gleichzeitig die Rolle des Gate-Verantwortlichen besetzen). Ferner sorgt gerade der Qualitätsbeauftragte dafür, dass die Kriterien eines Quality Gates eingehalten werden können. Betrifft ein Quality Gate mehrere Teilprojekte, so sind die Projektvertreter dieser Teilprojekte einzuladen.

Projektvertreter sollten beim Walkthrough und der formalen Inspektion in der Sitzung anwesend sein. Beim Schreibtischtest stehen sie hingegen im Kontakt mit dem Gutachter bzw. Gatekeeper.

8.5 Protokollant

Die Aufgabe des Protokollanten liegt bei der Erstellung eines Protokolls. Sie kann nur in den Gate-Review-Varianten Walkthrough und formale Inspektion durch eine Person wahrgenommen werden, die sonst keine andere Rolle wahrnimmt. Bei der Variante Schreibtischtest wird sie direkt vom Gutachter bzw. Gatekeeper und damit von ein und derselben Person wahrgenommen. Innerhalb der Sitzung kann sie mit jeder anderen Rolle kombiniert werden, da die Rolle keine besondere Qualifikation erfordert.

8.6 Protokoll

Im Protokoll werden im Wesentlichen das Ergebnis eines Gate-Reviews (erfüllte und nichterfüllte Kriterien), die Entscheidung und zu treffende Maßnahmen verfestigt. Weitere Informationen sind hingegen nützlich. Zusammengefasst sollte das Protokoll für ein Quality Gate die folgenden Angaben enthalten:

- Nennung des Kontextes (Quality Gate, Projekt und Datum des Gate-Reviews), für das das Protokoll gilt.
- Angaben zur Art und Durchführung des Gate-Reviews und zu den Teilnehmern.
- Die Checkliste mit den Kriterien ist dem Protokoll beizufügen. Für jedes Kriterium ist jeweils vermerkt, ob es erfüllt ist oder nicht (Erfüllungsgrad). Bei Nichterfüllung sollte der Istwert oder die Abweichung zum Sollwert mindestens als Begründung des Erfüllungsgrades notiert werden. Falls ein Kriterium sich als nicht anwendbar herausstellt, sollte dies ebenfalls dokumentiert werden. Gegebenenfalls ist ebenfalls zu dokumentieren, wenn die Messung anders als vorgesehen durchgeführt oder der Sollwert angepasst wurde.
- Die Entscheidung ist zu dokumentieren. Gegebenenfalls ist sie zu begründen, insbesondere wenn eine weitreichende Entscheidung getroffen wurde oder sie vom Ergebnis der Entscheidungsunterstützung abweicht. Im Falle der Wiederholung des Quality Gates, ist der Termin des Quality Gates festzuhalten.
- Die Maßnahmen sind zu dokumentieren. Hierzu gehört auch der Inhalt des Freigabeplans.
- Dokumentation von Leistungsdaten des Gate-Reviews.
- Angaben über Inhalte und Termine beliebig vieler nachfolgender Quality Gates.

Abbildung 83 zeigt eine Übersicht der Bestandteile eines Protokolls. Verpflichtende Bestandteile sind mit einem dicken schwarzen Rahmen versehen. Die weiteren Bestandteile sind hingegen optional.

9 Ausgestaltung der Steuerungskonzepte

Steuerungskonzepte beschäftigen sich damit, welche Entscheidungen in einem Quality Gate generell getroffen werden können. Dabei ist wichtig, durch wen die Rolle des Gatekeepers besetzt werden kann und wie die Gatekeeper durch eine Entscheidungsunterstützung unterstützt werden können. Die nachfolgenden Abschnitte beschäftigen sich mit den Richtlinien, Empfehlungen und dem Spektrum zur bzw. der Ausgestaltung dieser Teilkonzepte.

9.1 Mögliche Entscheidungen

Im Rahmen der Ausgestaltung eines Quality-Gate-Referenzprozesses ist zu klären, welche Entscheidungen generell durch die Gatekeeper in einem Quality Gate getroffen werden können. Dabei muss es generell möglich sein, ein Projekt uneingeschränkt fortsetzen zu können, sowie die Fortsetzung in irgendeiner Weise zu beeinflussen, falls nicht alle Kriterien erfüllt sein sollten. Folglich ergibt sich die Menge der möglichen Entscheidungen, die im Gate-Review getroffen werden können, formal gesehen zu

$$\{\text{go}\} \cup E_{neg},$$

wobei $E_{neg} \in P(\{\text{conditional-go, repeat-gate, hold, kill}\})$ mit $|E_{neg}| \geq 1$ gilt.

Nachfolgend werden die einzelnen Entscheidungen näher erläutert.

- **Uneingeschränkte Projektfortsetzung („go“):** Die uneingeschränkte Projektfortsetzung ist immer dann zu wählen, wenn alle Kriterien des jeweiligen Quality Gates erfüllt sind oder nur wenige (nicht stark gewichtete) Kriterien nicht erfüllt worden sind und die Gatekeeper sich dafür entscheiden, dass die Erfüllung dieser Kriterien nicht mehr notwendig ist. Die uneingeschränkte Projektfortsetzung impliziert die uneingeschränkte Freigabe aller notwendigen Ressourcen für den nachfolgenden Prozessschritt. Maßnahmen sind in diesem Fall nicht zu treffen.
- **Bedingte Projektfortsetzung („conditional-go“):** Die bedingte Projektfortsetzung ist dann zu wählen, wenn das Projekt einige leicht gewichtete Kriterien und eventuell nur sehr wenige stärker gewichtete Kriterien nicht erfüllen kann. Zusätzlich sollte es sich dabei um Kriterien handeln, die ohne größeren Ressourcen-Aufwand nebenläufig zum nächsten Prozessschritt erfüllt werden können. Die Kontrolle der offenen Kriterien kann dann entweder innerhalb der formalen Fortschrittsüberwachung oder im Meilenstein-Review durch das Management erfolgen (vgl. Tabelle 3). In der Regel sollte ein Termin gesetzt werden, zu dem die Kriterien spätestens erfüllt sein müssen. Sollten die Kriterien dann nicht erfüllt sein, so sollte das Management Maßnahmen ergreifen oder die Wiederholung des jeweiligen Quality Gates anordnen. Die Gefahr der bedingten Projektfortsetzung ist, dass die Grenze zwischen bestandenen und nicht bestandenen Quality Gate schwimmt – es handelt quasi um eine „schwammige“ Projektfortsetzung. Das Quality-Gate-Konzept eignet sich nur eingeschränkt als Fortschrittsmaß, wenn die bedingte Projektfortsetzung zulässig ist. Die bedingte Projektfortsetzung ist eine Möglichkeit, das Fuzzy-Konzept nach Cooper umzusetzen (vgl. Abschnitt 4.6.1.2) und verursacht eine geringere Verzögerung als eine Wiederholung des Gate-Reviews.
- **Wiederholung des Quality Gates („repeat-gate“):** Die Wiederholung des Quality Gates ist dann zu wählen, wenn das Projekt einige wichtige Kriterien nicht erfüllen kann, die jedoch noch mit vertretbarem Aufwand zeitnah erfüllt werden können. Dabei spielt die psychologische Wirkung des weiteren Quality Gates eine wichtige Rolle. Es signalisiert, dass die Erfüllung der Kriterien wichtig ist und dass bei Nichterfüllung weitreichende Maßnahmen getroffen werden können. Der Nachteil der Wiederholung ist, dass erneut ein Gate-Review durchgeführt werden muss und hierfür Ressourcen aufgebracht werden müssen.

- **Zurückstellung des Projektes („hold“):** Bei der Zurückstellung handelt es sich um eine schwächere Form des Projektabbruchs. Es werden Ressourcen vom Projekt abgezogen, wobei aber damit zu rechnen ist, dass das Projekt zukünftig fortgesetzt wird. In diesem Fall ist absehbar, dass die Kriterien bei Wiederaufnahme auch erfüllt werden können. Die Möglichkeit zur Zurückstellung eines Projektes ist besonders sinnvoll, wenn der Faktor Zeit bei Entwicklung weniger relevant ist. Das Projekt kann fortgesetzt werden, wenn freie Ressourcen zur Verfügung stehen.
- **Projektabbruch („kill“):** Der Projektabbruch ist dann zu wählen, wenn eine Vielzahl sehr wichtiger Kriterien nicht erfüllt werden konnten und nicht absehbar ist, dass sie noch erfüllt werden können. Dabei muss es sich nicht notwendigerweise um die Nichterfüllung von qualitätsbezogenen Kriterien handeln. Vielmehr sind es sogar nicht qualitätsbezogene Kriterien, die zu einem Abbruch des Projektes führen, zum Beispiel weil das Softwareprodukt nicht mehr in das Produktportfolio des Unternehmens passt oder weil eine Änderung der Marktsituation eingetreten ist (vgl. Abschnitt 7.3). Andererseits kann eine hohe Gewinnaussicht auch dafür sorgen, dass Qualitätsdefizite ignoriert werden und das Projekt nicht abgebrochen wird, obwohl dies eigentlich notwendig wäre. Können Abbruch-Entscheidungen in Quality Gates getroffen werden, sind wahrscheinlich nicht qualitätsorientierte Kriterien für die Entscheidungsfindung notwendig.

Ein Unternehmen kann sich nun dazu entscheiden, grundsätzlich keine bedingte Projektfortsetzung zu erlauben, sondern nur die Möglichkeit zur Wiederholung eines Quality Gates einzuräumen, um die besondere Bedeutung eines Quality Gates herauszuheben. Dies ist besonders wichtig, wenn Quality Gates als Fortschrittsmaß genutzt werden sollen.

Es kann zwischen weitreichenden Entscheidungen und Fortsetzungs-Entscheidungen unterschieden werden. Zu den weitreichenden Entscheidungen zählen die Zurückstellung und der Abbruch des Projektes. Zu den Fortsetzungsentscheidungen zählen die uneingeschränkte und die bedingte Projektfortsetzung. Die Wiederholung des Quality Gates kann je nach Aufwand des Gate-Reviews beiden Gruppen von Entscheidungen hinzugezählt werden. Durch die Ausgestaltung sollte geregelt werden, welche Gatekeeper Rollen welche Entscheidungen treffen können.

Das Prozessmanagement eines Unternehmens wird typischerweise eine der folgenden Kombinationen möglicher Entscheidungen wählen:

- Alle Entscheidungen sind möglich.
- Alle Entscheidungen außer der Zurückstellung des Projektes sind möglich.
- Alle Entscheidungen außer der Zurückstellung des Projektes und der bedingten Projektfortsetzung sind möglich.

9.2 Gatekeeper

Die Aufgabe der Gatekeeper besteht darin, eine Entscheidung über die Fortsetzung des Projektes im Gate-Review zu treffen. Gatekeeper sind folglich immer durch Personen mit Entscheidungsbefugnis und idealerweise auch mit tiefgehender Fachkenntnis zu besetzen. Sind nicht ausschließlich qualitätsorientierte Kriterien zu bewerten, so sind Rollen mit weiteren Fachkenntnissen und Befugnissen in das Gate-Review einzubeziehen (beispielsweise Produktmanager). Gatekeeper stehen außerhalb des Projektes bzw. „über“ dem Projekt (vgl. Abschnitt 3.1.1).

Gatekeeper können hinsichtlich ihrer Entscheidungsbefugnis unterschieden werden. Einige Gatekeeper können keine weitreichenden Entscheidungen treffen, dennoch aber über die uneingeschränkte, bedingte Projektfortsetzung und meistens auch über die Wiederholung des Quality Gates entscheiden. Weitreichende Entscheidungen können durch den Qualitätsmanager in Absprache mit dem Projektmanager getroffen werden. Bei wichtigen Projekten (wenn das Projekt z.B. mehrere Unternehmensbereiche involviert) sind die Geschäftsleitung und die Stabstelle des Qualitätsleiters einzuschalten. Ebenfalls kann ein externer Kunde zu den Gatekeepern bei Quality Gates mit externer Wirkung zählen, da

er über die Freigabe der Ressourcen entscheiden kann und die Produkte für ihn relevant bzw. nutzbar sind.

Cooper geht davon aus (vgl. Abschnitt 4.6.3), dass Gatekeeper innerhalb eines Projektes möglichst immer mit den gleichen Personen besetzt werden sollten. Dies ist zum Teil durch die obigen Richtlinien gewährleistet, wenn zum Beispiel immer der Projektmanager und der Qualitätsmanager als Gatekeeper an der Entscheidung beteiligt werden.

Es ist sinnvoll, wenn an der Sitzung im Gate-Review nur Gatekeeper teilnehmen, die über keine weitreichende Entscheidungsbefugnis verfügen (vgl. Abschnitt 8.1). Dies ist ausreichend, da die Fortsetzungs-Entscheidungen im Gate-Review den Normalfall darstellen und nur im Ausnahmefall weitreichende Entscheidungen zu treffen sind. Im letzteren Falle ist zu den Gatekeepern mit weitreichender Entscheidungsbefugnis zu eskalieren. Diese Vorgehensweise kann in einer Eskalationsstrategie festgehalten werden (vgl. Abbildung 85). Im Rahmen der Ausgestaltung ist festzulegen, welche Rolle welche Entscheidung treffen darf. Hierdurch wird auch gleichzeitig die Eskalationsstrategie verbindlich festgelegt.

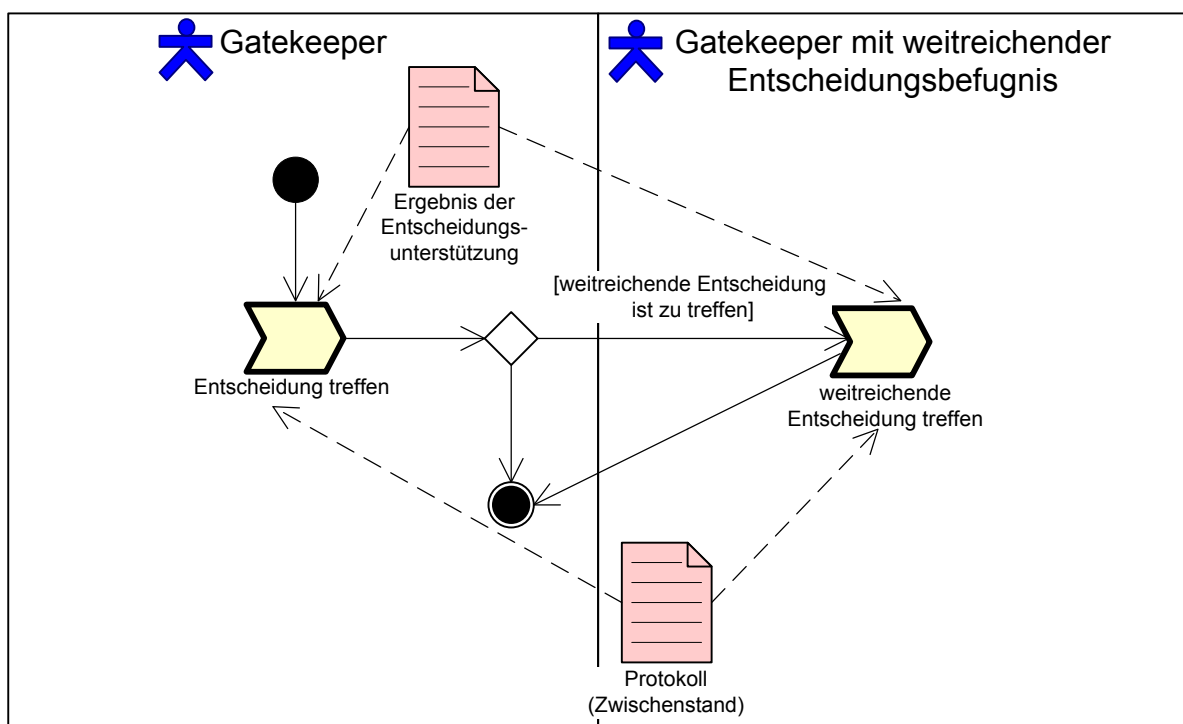


Abbildung 85: Eskalationsstrategie

9.3 Entscheidungsunterstützung

Eine Entscheidungsunterstützung unterstützt die Gatekeeper beim Treffen einer Entscheidung im Gate-Review. Eine durchdachte Entscheidungsunterstützung unterstützt die Entscheidungsfindung in zweierlei Hinsicht:

- Sie liefert eine Empfehlung darüber, welche Entscheidung aufgrund der Erfüllung oder Nichterfüllung von Kriterien im aktuellen Quality Gate zu treffen ist.
- Sie hilft dabei, Entscheidungen über die Grenzen des Projekts hinweg, auch in anderen Projekten gleich treffen zu können und erleichtert damit die Umsetzung der grundlegenden Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie*.

Eine Entscheidungsunterstützung kann wie folgt formalisiert werden:

Definition: Sei g eine Funktion, die für ein beliebiges n -Tupel

$$k = (k_1, \dots, k_n) \text{ mit } k_i \neq k_j \quad \forall i, j \in \{1, \dots, n\} \text{ mit } i \neq j$$

von operationalisierten Kriterien eines Quality Gates die Gewichtung

$$(g_1, \dots, g_n)$$

dieser Kriterien liefert, wobei g_i das Gewicht des Kriteriums k_i für alle $i \in \{1, \dots, n\}$ ist.

Sei $b = (b_1, \dots, b_n) \in B^n$ ein n -Tupel, das für jedes Kriterium in k angibt, ob es erfüllt ist oder nicht. Dabei gilt

$$\forall i \in \{1, \dots, n\} \text{ gilt: } k_i \text{ ist erfüllt} \Leftrightarrow b_i = \text{true}.$$

b ist das Ergebnis der Bewertung der Kriterien durch die Gutachter im Gate-Review.

Ferner sei E eine Ordinalskala, die die möglichen Empfehlungen enthält. Für je zwei Elemente e_1 und e_2 aus E mit $e_1 \leq e_2$ gilt, dass e_2 eine „bessere“ (oder zumindest gleich gute) Empfehlung für ein Projekt ist als e_1 .

Eine Entscheidungsunterstützung λ ist eine Funktion, die Gewichtung und Bewertung der Kriterien auf eine Empfehlung abbildet:

$$\lambda(g(k), b) = e.$$

λ hat die folgende Eigenschaft (monoton steigend):

$\forall b_1, b_2 \in B^n$ mit $b_1 \leq b_2$ und gegebenes k gilt:

$$\lambda(g(k), b_1) \leq \lambda(g(k), b_2).$$

Wobei die Relation \leq auf B^n wie folgt definiert ist:

Sind $b_1, b_2 \in B^n$ dann gilt $b_1 \leq b_2$ genau dann, wenn $b_{1,i} \leq b_{2,i} \quad \forall i \in \{1, \dots, n\}$, wobei für die boolschen Werte *true* und *false* gilt: *false* < *true*.

Definition 18: Definition Entscheidungsunterstützung

Umgangssprachlich lässt sich die untenstehende Eigenschaft einer Entscheidungsunterstützung λ folgendermaßen interpretieren:

Bewerten zwei Gutachter die Kriterien genau gleich und entscheidet sich ein Gutachter später dazu, ein weiteres Kriterium als erfüllt zu bewerten, dann erhält das Projekt im Quality Gate eine bessere Empfehlung als zuvor (oder zumindest eine gleich gute Empfehlung).

Die Menge der Empfehlungen muss nicht zwangsläufig unmittelbar die Entscheidungen *go*, *conditional-go*, *repeat-gate*, *hold* und *kill* enthalten. Falls dies dennoch der Fall ist, gilt:

$$\text{go} > \text{conditional-go} > \text{repeat-gate} > \text{hold} > \text{kill}.$$

Das klassische Beispiel für eine Menge von Empfehlungen ist die dreifarbige Ampel bestehend aus den Farben rot, gelb und grün. Ein Projekt enthält dann als Empfehlung eine der drei Ampelfarben. Die Gatekeeper haben dann die Aufgabe, aus der Farbe der Ampel die eigentliche Entscheidung abzuleiten.

Im Folgenden werden zwei Beispiele für Entscheidungsunterstützungen gegeben:

Beispiel: Es werden die gleichen Kriterien mit Gewichtungen wie in Beispiel 2 angenommen. Die Gutachter kommen im Gate-Review zum Resultat, dass die Kriterien $k = (k_1, \dots, k_5)$ wie folgt zu bewerten sind:

$$b = (\text{true}, \text{false}, \text{true}, \text{false}, \text{true}).$$

Für die farbliche Gewichtung g_{farb} der Kriterien (vgl. Beispiel 2) gilt:

$$g_{\text{farb}}(k) = (\text{rot}, \text{gelb}, \text{orange}, \text{rot}, \text{gelb}).$$

Die Menge der Empfehlungen E sei (rot, gelb, grün).

Eine Entscheidungsunterstützung λ_1 , die sich auf farbliche Gewichtung bezieht, sei wie folgt definiert:

- λ_1 liefert *rot* zurück, wenn mindestens ein rot gewichtetes Kriterium nicht erfüllt ist.
- λ_1 liefert *gelb* zurück, wenn mindestens zwei gelb gewichtete Kriterien nicht erfüllt sind, aber alle rot gewichteten Kriterien erfüllt sind.
- λ_1 liefert *grün* zurück, wenn höchstens ein gelb gewichtetes Kriterium nicht erfüllt ist, aber alle rot gewichteten erfüllt sind.

λ_1 liefert *rot* zurück, da das rot gewichtete Kriterium 4 nicht erfüllt ist.

Eine zweite Entscheidungsunterstützung λ_2 , die sich auf die Risikokennzahl bezieht, sei wie folgt definiert:

$$\lambda_2 : (g(k), b) \mapsto \begin{cases} \text{rot,} & \text{falls } f(g(k), b) < 0,5 \\ \text{gelb,} & \text{falls } 0,5 \leq f(g(k), b) < 0,85 \\ \text{grün,} & \text{falls } f(g(k), b) \geq 0,85, \end{cases}$$

wobei die Funktion $f(g(k), b)$ wie folgt definiert ist:

$$f : (g(k), b) \mapsto \frac{\sum_{i=1}^n \chi(b_i) \cdot g_i}{\sum_{i=1}^n g_i} \quad \text{mit } \chi_i : b \mapsto \begin{cases} 1, & \text{falls } b = \text{true} \\ 0, & \text{falls } b = \text{false.} \end{cases}$$

Nach Beispiel 2 ergibt sich folgende Gewichtung g_{risk} der Kriterien über die Risikokennzahl:

$$g_{\text{risk}}(k) = (6.3, 1, 2.4, 3.5, 0.95).$$

Damit ergibt sich $f(g_{\text{risk}}(k), b) \approx 0,68$. λ_2 liefert also in diesem Fall *gelb* zurück.

Beispiel 3: Beispiele für Entscheidungsunterstützungen

Die Definition einer guten und geeigneten Entscheidungsunterstützung erfordert viel Erfahrung, meistens wird man sich dabei auf Intuition und Erfahrung stützen.

Informelle Entscheidungsunterstützungen, wie

„Wenn zu viele wichtige Kriterien nicht erfüllt sind, dann sollte entweder das Projekt abgebrochen werden oder zumindest ein weiteres Quality Gate durchgeführt werden.“

oder

„Wenn nur einige grün gewichtete und ganz wenige gelb gewichtete Kriterien verletzt sind, dann kann das Projekt bedingt fortgesetzt werden.“

lassen sich mit einer Entscheidungsunterstützung ähnlich zu λ_1 (vgl. Beispiel 3) relativ leicht und intuitiv formalisieren. Die mathematischere Formalisierung einer Entscheidungsunterstützung (vgl. λ_2 in Beispiel 3) ist hingegen eher schwieriger umzusetzen, da zum einen die Gewichte zumindest Elemente einer Rationalskala sein müssen und zum anderen die Berechnungsvorschrift schwer zu bestimmen ist.

Zusammengefasst lassen sich Entscheidungsunterstützungen unterscheiden, die nur informell und vage definiert sind und solche Entscheidungsunterstützungen, denen eine klare Berechnungsvorschrift zu Grunde liegt. Erstere folgen stark der menschlichen Intuition und Erfahrung, sind aber ungenau. Die informelle Definition erschwert es, Entscheidungen in ähnlichen Kontexten gleich treffen zu können. Ein leicht anderer Kontext ist bereits gegeben, wenn die Gruppe der Gatekeeper personell anders besetzt ist.

Formalisierte Entscheidungsunterstützungen mit einer Berechnungsvorschrift sind wiederholt einsetzbar. Sie setzen allerdings eine durchdachte Berechnungsvorschrift und eine systematisch durchgeführte Gewichtung der Kriterien voraus.

Eine weitere Dimension neben der Formalität einer Entscheidungsunterstützung, ist die Verbindlichkeit einer Entscheidungsunterstützung, die jedoch nur im Falle einer formalisierten Entscheidungsunterstützung sinnvoll ist. Eine informelle Entscheidungsunterstützung ist hingegen immer unverbindlich. Im Falle einer verbindlichen Entscheidungsunterstützung, ist das Ergebnis einer Entscheidungsunterstützung für die Gatekeeper bindend. Dies gilt selbst für den Fall, dass das Ergebnis einer Entscheidungsunterstützung noch auf eine der fünf möglichen Entscheidungen abgebildet werden muss. In diesem Fall darf es beispielsweise nicht möglich sein, das Projekt fortzuführen, obwohl die Entscheidungsunterstützung „rot“ zurückliefert. Eine verbindliche Entscheidungsunterstützung erleichtert es, projektübergreifend die gleichen Entscheidungen bei gleicher Ausgangslage treffen zu können.

Unverbindliche Entscheidungsunterstützungen geben lediglich einen Ratschlag, welche Entscheidung durch die Gatekeeper getroffen werden sollte. Das Ergebnis einer Entscheidungsunterstützung ist dann interpretierbar. Unverbindliche Entscheidungsunterstützungen erschweren es, gleiche Entscheidungen bei gleicher Ausgangslage treffen zu können. Der Vorteil ist allerdings, dass Gatekeeper flexibler reagieren können, falls sich zum Beispiel die Gewichtung eines Kriteriums als falsch herausstellt oder sich ein wichtiges, jedoch nicht erfülltes, Kriterium durch leichte Nacharbeit parallel zum fortlaufenden Projekt erfüllen lässt. Entscheidungsunterstützungen, die formalisiert und unverbindlich ausgestaltet sind, erleichtern es also, gleiche Entscheidungen treffen zu können, wenn sich Projekte lediglich in Details unterscheiden.

In der Literatur sind nur selten ausgestaltete Entscheidungsunterstützungen zu finden. Hawlitzky [Hawlitzky '02] definiert in seinen Beispielen eine formale und verbindliche Entscheidungsunterstützung basierend auf den Ampelfarben Gewichtungen *rot*, *gelb* und *grün*. Sie ist wie folgt definiert:

- Wenn mindestens ein *rot* gewichtetes Kriterium nicht erfüllt ist, so muss das jeweilige Quality Gate wiederholt werden. Die Entscheidungsunterstützung liefert *rot*.
- Wenn mindestens ein *gelb* gewichtetes Kriterium nicht erfüllt ist, aber alle *rot* gewichteten Kriterien erfüllt sind, so ist das Projekt bedingt fortzusetzen. Die Entscheidungsunterstützung liefert *gelb*.
- Wenn alle Kriterien erfüllt sind, so kann das Projekt uneingeschränkt fortgesetzt werden. Die Entscheidungsunterstützung liefert *grün*.

Cooper [Cooper '01] gewichtet nach Muss- und Soll-Kriterien und gestaltet seine verbindliche und teilweise formalisierte Entscheidungsunterstützung so aus, dass nicht erfüllte Muss-Kriterien sofort zum Projektabbruch führen. Auf Grundlage der Erfüllung der Soll-Kriterien werden verschiedene Projekte untereinander verglichen (vgl. Abbildung 41). Zu viele nicht erfüllte Soll-Kriterien führen zunächst zu einer Zurückstellung des Projektes, wobei dieser Fall nur vage durch Cooper beschrieben wird, um die notwendige Flexibilität für den Vergleich zu wahren.

Mit einer formalisierten und verbindlichen Entscheidungsunterstützung kann auch das Fuzzy-Konzept nach Cooper (vgl. Abschnitt 4.6.1.2) realisiert werden, da mit der Definition einer Entscheidungsunterstützung festgelegt werden kann, unter welchen Bedingungen das Projekt bedingt fortgesetzt werden kann, obwohl noch nicht alle Kriterien erfüllt sind.

Tabelle 34 fasst die Vor- und Nachteile einer informellen oder formalisierten Entscheidungsunterstützung zusammen.

+/-	Informell	Formalisiert
Vorteile	- Vage Formulierung ist näher an der menschlichen Intuition und Erfahrung.	- Entscheidung kann bei gleicher Ausgangslage gleich getroffen werden.
Nachteile	- Entscheidung wird trotz gleicher Ausgangslage wahrscheinlich nicht gleich getroffen.	- Formalisierung eher schwierig, da sie Erfahrung erfordert.

Tabelle 34: Vor- und Nachteile einer informellen oder formalisierten Entscheidungsunterstützung

Tabelle 35 zeigt die Vor- und Nachteile einer formalisierten verbindlichen oder formalisierten unverbindlichen Entscheidungsunterstützung zusammen. Die Verbindlichkeit ist nur bei einer formalisierten Entscheidungsunterstützung relevant.

+/-	Unverbindlich	Verbindlich
Vorteile	- Es kann eine andere Entscheidung getroffen werden, wenn die Gatekeeper den Projektstatus anders beurteilen und die Entscheidung lediglich von Details abhängt.	- Entscheidung kann bei gleicher Ausgangslage gleich getroffen werden.
Nachteile	- Es besteht eine gewisse (jedoch vermutlich geringe) Wahrscheinlichkeit, dass die Entscheidung nicht einheitlich getroffen wird.	- Kann zur Fehlbeurteilung eines Projektes führen, wenn ein Kriterium falsch gewichtet wurde, aber Nacharbeit leicht ist. - Setzt hohes Vertrauen in Entscheidungsunterstützung voraus.

Tabelle 35: Vor- und Nachteile einer ver- oder unverbindlichen Entscheidungsunterstützung

10 Konzepte zur Anpassung von Quality-Gate-Referenzprozessen

Anpassungskonzepte beschäftigen sich damit, wie und durch wen ein Quality-Gate-Referenzprozess für eine gegebene Projektsituation angepasst werden kann. Die Grundannahme dabei ist, dass nicht für jedes Projekt der gleiche Quality-Gate-Prozess verwendet werden kann. Dies kann beispielsweise der Fall sein, wenn dieser Quality-Gate-Prozess zu aufwändig in Relation zur Wichtigkeit eines Projektes ist oder weil bestimmte Kriterien nicht in jedem Projekt sinnvoll sind.

Darüber hinaus ist zu klären, welche Teilkonzepte angepasst werden können. Weiterhin wird im Rahmen einer Ausgestaltung der Anpassungskonzepte geklärt, *wer* für die Pflege des Quality-Gate-Referenzprozesses verantwortlich ist. Das *Wie* der Anpassung wird durch die Ausgestaltung des Teilkonzeptes *Tailoring-Verfahren* geklärt. Ein Tailoring-Verfahren nutzt ein Regelwerk bestehend aus einzelnen Regeln. Regeln und Tailoring-Verfahren stützen sich auf ein Projektmodell. Die Durchführung des Tailorings kann durch Tailoring-Softwarewerkzeuge unterstützt werden. Abbildung 86 fasst das Umfeld eines Tailoring-Verfahrens zusammen.

Der Prozess-Tailorer ist für die Anpassung eines Quality-Gate-Referenzprozess verantwortlich, während das Gate-Management für die Pflege verantwortlich ist. Das *Was* wird letztlich durch die Klärung der auswählbaren Elemente bestimmt (vergleiche Abbildung 49). Die nachfolgenden Abschnitte beschäftigen sich mit den Richtlinien und dem Spektrum zur Ausgestaltung dieser Teilkonzepte.

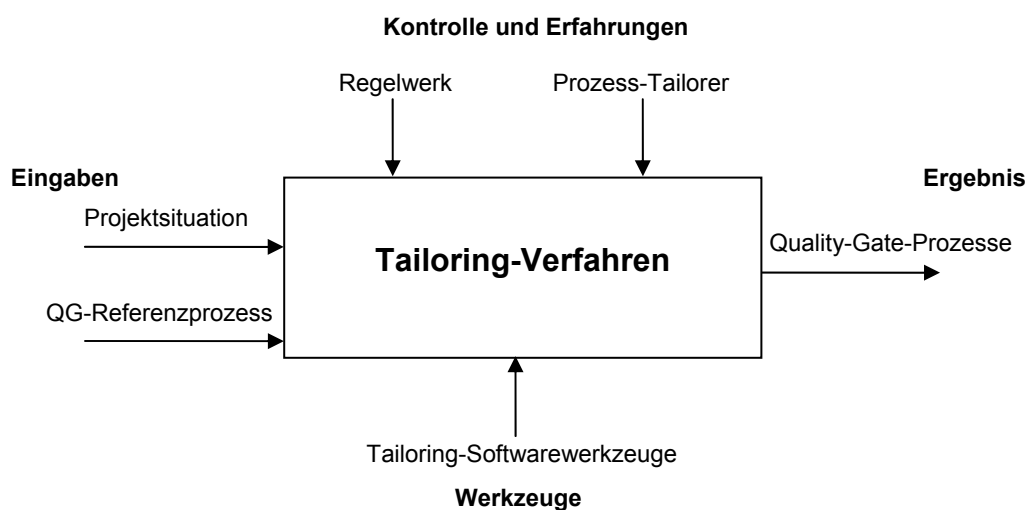


Abbildung 86: Umfeld eines Tailoring-Verfahrens in IDEF0-Notation

10.1 Gate-Management

Die Aufgabe des Gate-Managements besteht in der Pflege eines bereits vorhandenen Quality-Gate-Referenzprozesses. Das Prozessmanagement kann jedoch schon sehr früh ein Gate-Management bestimmen, das auch die erstmalige Ausgestaltung übernimmt. Das Gate-Management kann dabei mit dem Prozessmanagement eines Unternehmens übereinstimmen, das den Quality-Gate-Referenzprozess ausgestaltet hat.

Dauerhaft kann das Gate-Management auch als Teil des Prozessmanagements etabliert werden. Nicht jedes Unternehmen besitzt die Ressourcen, um ein dediziertes Gate-Management oder ein Prozessmanagement zu etablieren. Größe des Unternehmens und fehlende fachliche Organisation beeinflussen die Ausgestaltungsmöglichkeiten. Ein Quality-Gate-Referenzprozess, der nicht durch ein Gate-Management gepflegt wird, kann nicht verbessert und auf neue Herausforderungen hin optimiert wer-

den. Dies führt unweigerlich über die Zeit zu einem inadäquaten Quality-Gate-Referenzprozess (vgl. Abschnitt 11.4).

Es existieren verschiedene Ausbaustufen des Gate-Managements, die auch kleineren Unternehmen die Etablierung eines Gate-Managements ermöglichen (vgl. Tabelle 36). Sie unterscheiden sich im benötigten Ressourcenbedarf und dem Aufwand, der für die Pflege des Quality-Gate-Referenzprozess aufgebracht wird.

Keines	Minimal	Mittel	Maximal
Unternehmen besitzt kein Gate-Management.	Übernommen durch Geschäftsleitung oder Delegation an Stabsstelle, da Unternehmen kein Prozessmanagement besitzt.	Prozessmanagement übernimmt Gate-Management. Gate-Management wird nicht als eine Hauptaufgabe betrachtet.	Dediziertes Gate-Management als Teil des Prozessmanagements. Gate-Management wird zu einer querschnittlichen Hauptaufgabe.

Tabelle 36: Ausbaustufen des Gate-Managements

10.2 Prozess-Tailorer

Da die Anpassung eines Quality-Gate-Referenzprozesses in der Phase Projektstart stattfindet, ist das Projektkernteam für das Tailoring verantwortlich. Da der Fokus des Quality-Gate-Referenzprozess qualitätsorientiert, sollte das Projektkernteam idealerweise mit einem Qualitätsbeauftragten und einem Qualitätsmanager besetzt werden, da ansonsten die Qualitätssicherung innerhalb des Projektes vernachlässigt werden könnte und ein Quality-Gate-Prozess weniger aufwändig ausfallen könnte, als er eigentlich sein müsste.

Ebenso sind Projektleiter und Projektmanager anwesend, um die richtige Maß an Qualitätssicherung zu garantieren. Kleinere Unternehmen, ohne ausgeprägte Qualitätsorganisation, müssen notgedrungen das Tailoring durch Projektleiter und Projektmanager vornehmen lassen.

10.3 Tailoring-Verfahren

Das Tailoring-Verfahren eines Quality-Gate-Referenzprozesses bewegt sich in einem Spektrum zweier Extreme:

- Dem informellen Tailoring-Verfahren, das kein festes Projektmodell, keinen formalisierten Algorithmus und keine Klärung der auswählbaren Elemente benötigt, um einen geeigneten Quality-Gate-Prozess zu bestimmen. Ein solches Tailoring-Verfahren basiert auf Intuition und dem Erfahrungswissen der Prozess-Tailorer.
- Dem formalisierten Tailoring-Verfahren, das ein festes Projektmodell, einen festen Algorithmus mit Regelwerk und eine Klärung der auswählbaren Elemente benötigt (vgl. V-Modell XT).

Weiterhin kann das Tailoring-Verfahren hinsichtlich seiner Verbindlichkeit in verbindlich und unverbindlich unterschieden werden. Dies ist jedoch nur beim formalisierten Tailoring-Verfahren sinnvoll. Es weist damit die gleichen Dimensionen wie die Entscheidungsunterstützung auf. Entsprechend ergeben sich ähnliche Vor- und Nachteile wie bei der Entscheidungsunterstützung (vgl. Tabelle 34 und Tabelle 35). Die Verbindlichkeit ist im Falle eines formalisierten Tailoring-Verfahrens relevant.

Die folgende Tabelle 37 fasst die Vor- und Nachteile eines informellen oder formalisierten Tailoring-Verfahrens zusammen.

+/-	Informell	Formalisiert
Vorteile	<ul style="list-style-type: none"> - Vage Formulierung ist näher an der menschlichen Intuition und Erfahrung. - Ergebnis des Tailoring-Verfahrens wird stärker akzeptiert. 	<ul style="list-style-type: none"> - Gleiche Projektsituation liefert die gleichen möglichen Quality-Gate-Prozesse.
Nachteile	<ul style="list-style-type: none"> - Quality-Gate-Prozess wird bei gleicher Ausgangslage wahrscheinlich nicht gleich gewählt. 	<ul style="list-style-type: none"> - Formalisierung schwierig und langwierig, da sie Erfahrung und Feedbackschleifen erfordert.

Tabelle 37: Vor- und Nachteile eines informellen oder formalisierten Tailoring-Verfahrens

Tabelle 38 fasst Vor- und Nachteile eines verbindlichen oder unverbindlichen Tailoring-Verfahrens zusammen.

+/-	Unverbindlich	Verbindlich
Vorteile	<ul style="list-style-type: none"> - Der Prozess-Tailorer kann das Erfahrungswissen nutzen, kann aber anhand seiner Intuition und Erfahrung weiter anpassen. 	<ul style="list-style-type: none"> - Auswahl von gleichen Quality-Gate-Prozessen für gleiche Projektsituationen wird erleichtert.
Nachteile	<ul style="list-style-type: none"> - Quality-Gate-Prozess wird bei gleicher Ausgangslage wahrscheinlich nicht gleich gewählt. 	<ul style="list-style-type: none"> - Quality-Gate-Prozess wird unter Umständen nicht akzeptiert werden, wenn er den Erwartungen widerspricht.

Tabelle 38: Vor- und Nachteile eines verbindlichen oder unverbindlichen Tailoring-Verfahrens

Wie aus Abbildung 86 ersichtlich ist, kann ein Tailoring-Verfahren mehrere mögliche Quality-Gate-Prozesse liefern, die für eine Projektsituation geeignet sind. Für das Projekt muss dann genau einer dieser Quality-Gate-Prozesse gewählt werden. Bei der Auswahl des geeigneten Quality-Gate-Prozesses wird der Prozess-Tailorer sich dabei auf Wissen über das Projekt stützen, welches nicht über das Projektmodell und Regeln ausgedrückt werden kann. Dies ist durchaus gewünscht, da eine Projektsituation als Modell eines Projektes eventuell nicht alle zur Anpassung notwendigen Merkmale erfasst. Jedoch ist das Projektmodell so auszugestalten und zu pflegen, dass alle typischen Projektsituationen damit erfasst werden können. Die nicht abgebildeten (also präterierten Merkmale im Sinne der Modellbildung) aller Projekte sind dann als Ausnahmen zu betrachten. Sie sind jedoch für die Auswahl des passenden Quality-Gate-Prozesses wichtig, falls ein Tailoring-Verfahren mehrere mögliche Quality-Gate-Prozesse liefert.

Aus den bisherigen Betrachtungen ergeben sich zwei Möglichkeiten, die menschliche Intuition und Denkweise in das Ergebnis eines formalisierten Tailoring-Verfahrens einfließen zu lassen, um die Akzeptanz des Ergebnisses zu verbessern:

- Das Ergebnis des Tailoring-Verfahrens ist nicht verbindlich.
- Das Tailoring-Verfahren ist so gestaltet, dass es die menschliche Denkweise formalisiert.

Die Fuzzy-Logik bietet die Möglichkeit, die menschliche Intuition zu formalisieren und in ein Regelwerk und einen Algorithmus abzubilden, der das Tailoring erleichtert. In den folgenden Abschnitten wird ein regelbasiertes Expertensystem zur Anpassung von Quality-Gate-Referenzprozessen vorgestellt, das auf Fuzzy-Logik basiert. Es lehnt sich dabei an die Grundkonzepte der Ergebnisse der Doktorarbeit von Ittner [Ittner '05] an, nutzt jedoch ein anderes Verfahren – ein so genanntes Fuzzy-Inferenz-System – als Grundlage.

10.3.1 Grundlagen der Fuzzy-Logik

Ein Mensch bewertet Objekte der realen Welt nicht nur in der booleschen Logik mit ihren logischen Werten wahr und falsch. Vielmehr beurteilt ein Mensch verschiedene Objekte mittels unscharfer Werte wie zum Beispiel „fast“, „ein wenig“ oder „häufig“. Solche unscharfen Werte können mit Hilfe der Fuzzy-Logik ausgedrückt werden. Auf Basis der Fuzzy-Logik können dann beispielsweise vage Implikationen der folgenden Form

„Wenn das Projekt nicht so groß ist, dann sind bestimmte Kriterien nicht zu stark zu gewichten.“

oder

„Wenn das Projekt sehr wichtig ist, dann sollen an allen Gate-Review-Sitzungen Gatekeeper mit weitreichenden Entscheidungsbefugnissen teilnehmen.“

durch Regeln formalisiert werden. Mit Unterstützung eines so genannten Fuzzy-Inferenz-Systems ist es dann möglich, die Regeln durch ein Computersystem mit dem Ziel zu verarbeiten, geeignete Quality-Gate-Prozesse für eine gegebene Projektsituation zu bestimmen. Fuzzy-Inferenz-Systeme finden vor allem in der Regelungstechnik, zum Beispiel bei der Ansteuerung von Waschmaschinen, Anwendung. Ihre Grundidee lässt sich jedoch auch ohne Probleme für die Realisierung eines Tailoring-Verfahrens verwenden.

Innerhalb der Fuzzy-Logik existieren nicht nur die Werte *wahr* ($= 1$) und *falsch* ($= 0$), sondern auch beliebige Zwischenwerte (so genannte Fuzzy-Werte) des Einheitsintervalls $[0;1]$. Auf diese Weise ist es möglich, die oben beschriebenen unscharfen Werte „fast“, „ein wenig“ und „häufig“ auszudrücken. Ein Wert x des Intervalls $[0;1]$ lässt sich nun folgendermaßen interpretieren:

- Falls $x = 0$ ist, so ist die Aussage falsch.
- Falls $x \in (0;1)$ ist, so ist die Aussage zum Teil wahr. Der Wahrheitsgehalt nimmt zu, wenn der Abstand $|1 - x|$ sich verringert.
- Falls $x = 1$ ist, so ist die Aussage wahr.

Mit Hilfe der Fuzzy-Logik kann nun beispielsweise bewertet werden, wann ein Projekt in welchem Maße als groß bezeichnet werden kann. Dabei handelt es sich in diesem Zusammenhang bei „groß“, um einen so genannten linguistischen Wert. Er ist ein möglicher Wert der linguistischen Variablen „Projektgröße“. Weitere mögliche linguistische Werte der linguistischen Variable „Projektgröße“ können die Werte „klein“ und „mittel“ sein.

Allgemein lassen sich eine linguistische Variable und ein linguistischer Wert wie folgt definieren:

Definition: Eine linguistische Variable A ist ein Konstrukt des Sprachgebrauchs, das verschiedene unscharfe Werte annehmen kann. Diese Werte sind die so genannten linguistischen Werte der linguistischen Variablen A .

Definition 19: Linguistische Variable und linguistischer Wert

Die Bewertung der Zugehörigkeit eines bestimmten Wertes einer anderen Menge zu einem linguistischen Wert kann über eine so genannte Zugehörigkeitsfunktion vorgenommen werden. Eine Zugehörigkeitsfunktion ist wie folgt definiert:

Definition: Sei α ein linguistischer Wert einer linguistischen Variable A. Eine Zugehörigkeitsfunktion (auch Fuzzy-Funktion) für α ist eine Abbildung μ_α von einer Menge X in das Intervall $[0;1]$:

$$\mu_\alpha : X \rightarrow [0;1].$$

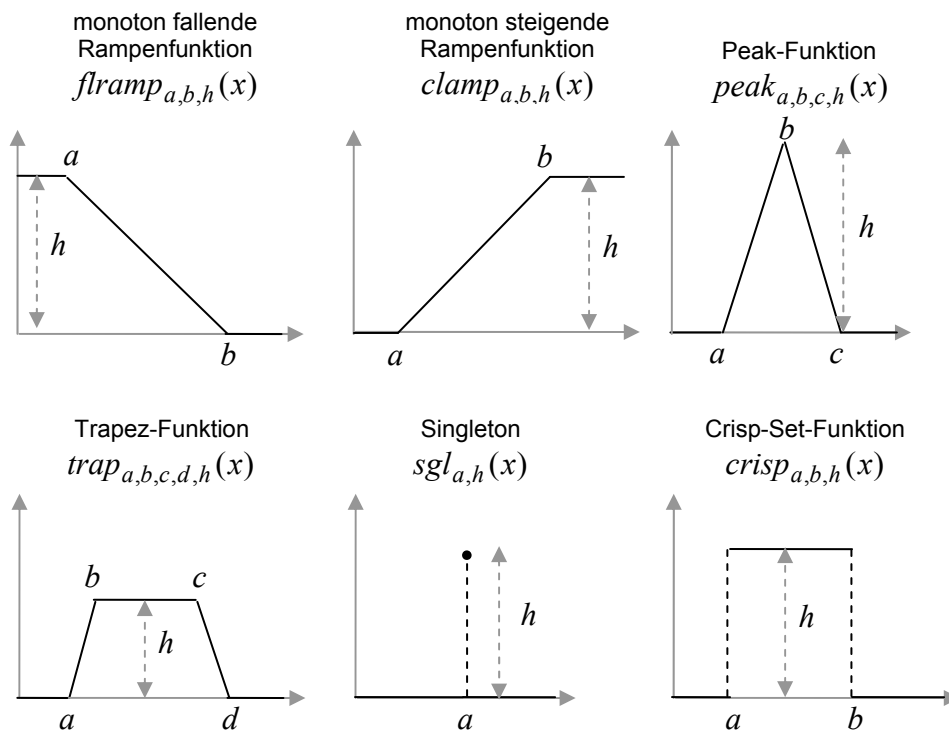
Die Menge X wird auch als Referenzmenge bezeichnet.

Definition 20: Zugehörigkeitsfunktion

Grundsätzlich kann eine Zugehörigkeitsfunktion beliebig gewählt werden, solange sie in das Einheitsintervall $[0;1]$ abbildet. Bestimmte Typen von Funktionen werden jedoch in der Fuzzy-Logik bevorzugt gewählt, weil sie intuitiv, einfach zu berechnen und gleichzeitig mächtig genug sind [Ittner '05] (vgl. Abbildung 87 und mathematische Definition im Anhang B). Im weiteren Verlauf werden nur Funktionen dieser Art betrachtet.

Inwieweit ein bestimmter Wert $x \in X$ eine linguistische Variable α „erfüllt“, kann nun mittels der Zugehörigkeitsfunktion beantwortet werden:

x erfüllt zum Grad $\mu_\alpha(x)$ den linguistischen Wert α .



Für alle Funktionen gilt $a < b < c < d$ mit $a, b, c, d \in X$ und $h \in [0;1]$.

Abbildung 87: Typische Zugehörigkeitsfunktionen

Singleton- und Crisp-Set-Funktionen mit Höhe 1 sind keine Fuzzy-Funktionen im eigentlichen Sinne, da sich mit ihnen nur die booleschen Werte wahr oder falsch modellieren lassen. Sie werden im weiteren Verlauf der Arbeit als uneigentliche Fuzzy-Funktionen betrachtet.

Im Folgenden wird ein Anwendungsbeispiel für Zugehörigkeitsfunktionen gegeben.

Beispiel: Die Projektgröße kann in Codezeilen (Lines of Code) gemessen werden. Anhand der Codezeilen ist nun festzustellen, wie „klein“, „mittel“ oder „groß“ ein Projekt ist. Die Referenzmenge X ergibt sich daher zu

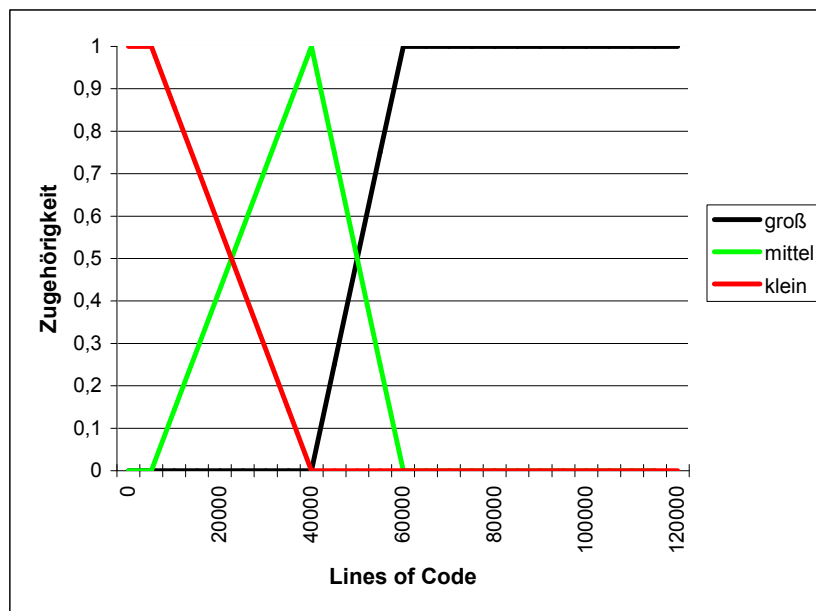
$$X = N^0.$$

In diesem Beispiel ist ein Projekt dann größer, wenn mehr Codezeilen zu schreiben sind. Für die linguistischen Werte „klein“, „mittel“ und „groß“ der linguistischen Variablen Projektgröße seien die Zugehörigkeitsfunktionen nun wie folgt definiert:

$$\begin{aligned} \mu_{\text{klein}} : N^0 &\rightarrow [0;1] & \mu_{\text{mittel}} : N^0 &\rightarrow [0;1] \\ \mu_{\text{klein}}(x) &= \text{flramp}_{5000,40000,1}(x) & \mu_{\text{mittel}}(x) &= \text{peak}_{5000,40000,60000,1}(x) \end{aligned}$$

$$\begin{aligned} \mu_{\text{groß}} : N^0 &\rightarrow [0;1] \\ \mu_{\text{groß}}(x) &= \text{clramp}_{40000,60000,1}(x). \end{aligned}$$

Die Graphen der Zugehörigkeitsfunktionen stellen sich wie folgt dar:



Folglich kann ein Projekt mit 30000 Codezeilen zu $\mu_{\text{klein}}(30000) = 0,29$ als klein, zu $\mu_{\text{mittel}}(30000) = 0,71$ als mittel und zu $\mu_{\text{groß}}(30000) = 0$ als groß bewertet werden.

Beispiel 4: Beispiele für Zugehörigkeitsfunktionen

10.3.2 Das Fuzzy-Inferenz-System

In der Fuzzy-Logik lässt sich ebenfalls die Implikation definieren. Es existieren verschiedene Definitionen der Implikation in der Fuzzy-Logik. Im Folgenden wird die sehr geläufige Definition nach Mamdani verwendet:

Definition: Seien α und β linguistische Werte und μ_α und μ_β jeweils Zugehörigkeitsfunktionen. Dann gilt:

$$(\mu_\alpha(x) \Rightarrow \mu_\beta(y)) = \min(\mu_\alpha(x), \mu_\beta(y)).$$

Definition 21: Implikation in der Fuzzy-Logik

Weiterhin lassen sich die Konjunktion, Disjunktion und Negation wie folgt definieren.

Definition: Seien α und β linguistische Werte und μ_α und μ_β jeweils Zugehörigkeitsfunktionen. Dann gilt:

$$\mu_\alpha(x) \wedge \mu_\beta(y) = \min(\mu_\alpha(x), \mu_\beta(y)) \quad \text{Fuzzy-Konjunktion}$$

$$\mu_\alpha(x) \vee \mu_\beta(y) = \max(\mu_\alpha(x), \mu_\beta(y)) \quad \text{Fuzzy-Disjunktion}$$

$$\neg\mu_\alpha(x) = 1 - \mu_\alpha(x) \quad \text{Fuzzy-Negation}$$

Definition 22: Logische Operatoren in der Fuzzy-Logik

Seien α und β linguistische Werte der linguistischen Variablen A und B. Weiterhin seien μ_α und μ_β jeweils Zugehörigkeitsfunktionen.

Mit Hilfe der Fuzzy-Logik können nun umgangssprachliche Regeln der Form

$$\text{Wenn } A = \alpha \text{ Dann } B = \beta$$

über

$$A = \alpha \Rightarrow B = \beta$$

in Fuzzy-Regeln übersetzt werden zu

$$\mu_\alpha(x) \Rightarrow \mu_\beta(y).$$

A wird hierbei als Eingangsgröße und B als Stellgröße innerhalb eines Fuzzy-Inferenz-Systemes bezeichnet.

Die Prämisse kann selbstverständlich aus beliebig vielen linguistischen Werten verschiedener Eingangsgrößen zusammengesetzt werden, die über logische Operatoren (vgl. Definition 22) miteinander verknüpft werden können. Dabei können alle Gesetze und Axiome übertragen werden, die auch in der booleschen Logik gelten. Im Folgenden werden nur Implikationen betrachtet, bei denen die Prämisse ausschließlich aus linguistischen Werten der Eingangsgrößen zusammengesetzt ist. Für die Konklusion wird angenommen, dass sie nur aus einem einzelnen linguistischen Wert einer Stellgröße besteht. Zum Beispiel wird eine Regel der Form

$$\text{Wenn } (A_1 = \alpha_1 \text{ und nicht } A_2 = \alpha_2) \text{ oder } A_3 = \alpha_3 \text{ Dann } B = \beta$$

zu

$$(\mu_{\alpha_1}(x_1) \wedge \neg\mu_{\alpha_2}(x_2)) \vee \mu_{\alpha_3}(x_3) \Rightarrow \mu_\beta(y).$$

Die Prämisse kann also als Abbildung beliebig vieler Referenzmengen in das Einheitsintervall $[0;1]$ verstanden werden:

$$\varphi : X_1 \times \dots \times X_n \rightarrow [0;1]$$

$$\varphi : (x_1, \dots, x_n) \mapsto \varphi(x_1, \dots, x_n),$$

wobei n die Anzahl der Stellgrößen ist.

Die Darstellung der Implikation vereinfacht sich damit zu:

$$\varphi(x_1, \dots, x_n) \Rightarrow \mu_\beta(y).$$

Regeln dieser Form bilden die Grundlage eines Fuzzy-Inferenz-Systems. Ein Fuzzy-Inferenz-System besteht aus drei Komponenten (Abbildung 88):

1. Der Fuzzifizierung, die über geeignete Zugehörigkeitsfunktionen einen scharfen Eingangswert $(x_1, \dots, x_n) \in X_1 \times \dots \times X_n$ in Fuzzy-Werte überführt.
2. Der Fuzzy-Inferenz, die aufgrund von Fuzzy-Regeln Schlüsse zieht.
3. Der Defuzzifikation, die einen nach außen hin repräsentativen und scharfen Ausgangswert liefert.

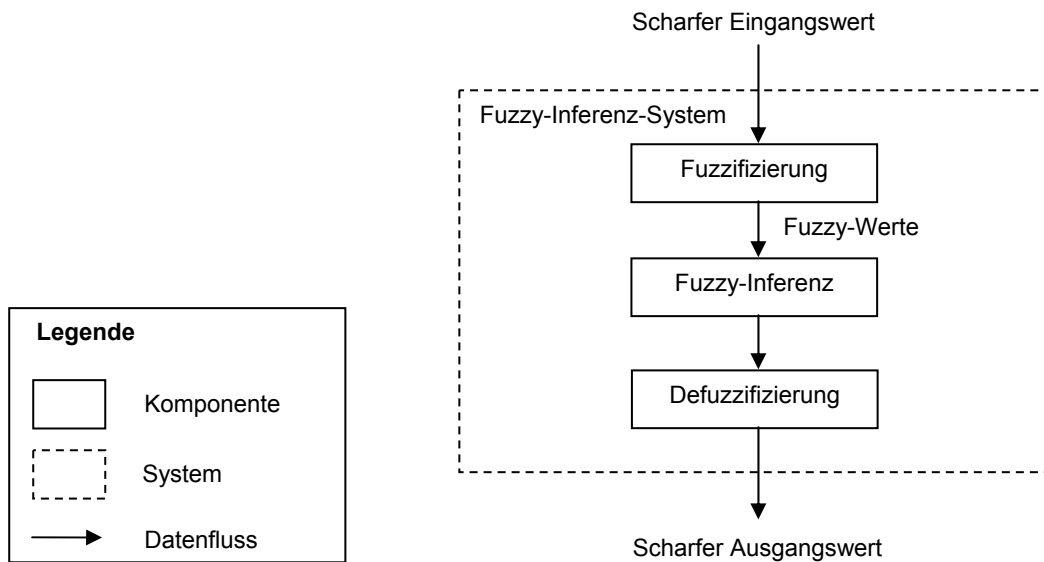


Abbildung 88: Aufbau eines Fuzzy-Inferenz-Systems

Nachfolgend werden die drei Komponenten eines Fuzzy-Inferenz-Systems beschrieben.

Fuzzifizierung

Seien A_1, \dots, A_n Eingangsgrößen und B eine Stellgröße. Jede der Eingangsgrößen A_i besitzt verschiedene linguistische Werte $\alpha_{i,j}$ mit $i \in \{1, \dots, n\}$ und $j \in \{1, \dots, k_i\}$. Ebenso besitzt B linguistische Werte β_l mit $l \in \{1, \dots, m\}$.

Für jedes $\alpha_{i,j}$ existieren Zugehörigkeitsfunktionen der Form

$$\mu_{\alpha_{i,j}} : X_i \rightarrow [0;1]$$

und für β_l der Form

$$\mu_{\beta_l} : Y \rightarrow [0;1].$$

Weiterhin existieren Implikationen (Fuzzy-Regeln) der folgenden Form:

$$\varphi_g(x_1, \dots, x_n) \Rightarrow \mu_{\beta_l}(y)$$

mit $\varphi_g : X_1 \times \dots \times X_n \rightarrow [0;1]$ und $g \in \{1, \dots, q\}$, wobei jedes μ_{β_l} höchstens einmal als Konklusion einer Regel auftaucht. Es gilt also $q \leq l$. Die Prämissen sind aus einer beliebigen Kombination der Zugehörigkeitsfunktionen $\mu_{\alpha_{i,j}}$ zusammengesetzt.

Die linguistischen Werte der linguistische Variable B im Kontext des Tailorings von Quality-Gate-Referenzprozessen bewerten stets den Grad der Notwendigkeit eines auswählbaren Elementes. Die Referenzmenge Y kann daher im weiteren Verlauf stets als Einheitsintervall $[0;1]$ angenommen werden.

Das Ergebnis der Fuzzifizierung ist die Auswertung der Prämissen für den einen konkreten Eingangswert $(\bar{x}_1, \dots, \bar{x}_n)$:

$$\varphi_1(\bar{x}_1, \dots, \bar{x}_n), \dots, \varphi_q(\bar{x}_1, \dots, \bar{x}_n).$$

Fuzzy-Inferenz

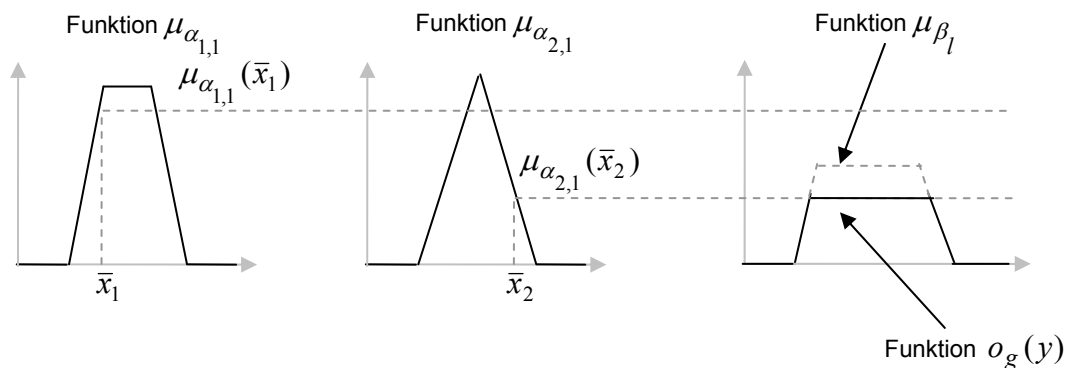
Innerhalb der Fuzzy-Inferenz werden mit Hilfe der ausgewerteten Prämissen aus der Fuzzifizierung und den Fuzzy-Regeln Schlüsse gezogen. Für jede Implikation

$$\varphi_g(x_1, \dots, x_n) \Rightarrow \mu_{\beta_l}(y)$$

liefert die Fuzzy-Inferenz keinen Fuzzy-Wert zurück, sondern zunächst eine so genannte Outputfunktion o_g der Form

$$o_g(y) = \min(\varphi_g(x_1, \dots, x_n), \mu_{\beta_l}(y)).$$

Grafisch lässt sich dies als Abschneiden der Zugehörigkeitsfunktion μ_{β_l} in Höhe $\varphi_g(x_1, \dots, x_n)$ interpretieren (vgl. beispielhafte Illustration in Abbildung 89).



$$\text{Prämisse } \varphi_g(x_1, \dots, x_n) = \mu_{\alpha_{1,1}}(x_1) \wedge \mu_{\alpha_{2,1}}(x_2)$$

$$\text{Ausgewertete Prämisse } \varphi_g(\bar{x}_1, \dots, \bar{x}_n) = \mu_{\alpha_{1,1}}(\bar{x}_1) \wedge \mu_{\alpha_{2,1}}(\bar{x}_2) = \mu_{\alpha_{2,1}}(\bar{x}_2)$$

Abbildung 89: Beispiel für das „Abschneiden“ einer Zugehörigkeitsfunktion

Defuzzifizierung

Die Defuzzifizierung liefert auf Grundlage der Outputfunktionen einen repräsentativen und scharfen Ausgangswert. Es existieren verschiedene Methoden zur Defuzzifizierung. Die hier vorgestellte Methode liefert gute Ergebnisse und ist leicht zu implementieren. Sie basiert darauf, dass so genannte „Mean of Maximum“ für jede Outputfunktion o_g zu bestimmen und zusammen mit den ausgewerteten Prämissen $\varphi_g(x_1, \dots, x_n)$ ein gewichtetes arithmetisches Mittel zu bilden – das so genannte „Center of Maximum“.

Die Outputfunktionen o_g verlaufen stets trapezförmig oder können zumindest als spezielle Trapeze angesehen werden, so dass die Berechnung des Mean of Maximum einfach ist (vgl. Abbildung 90). Das Mean of Maximum ist dabei grafisch gesehen der horizontale Mittelpunkt der Plateaufläche bzw. des Peaks der Outputfunktion.

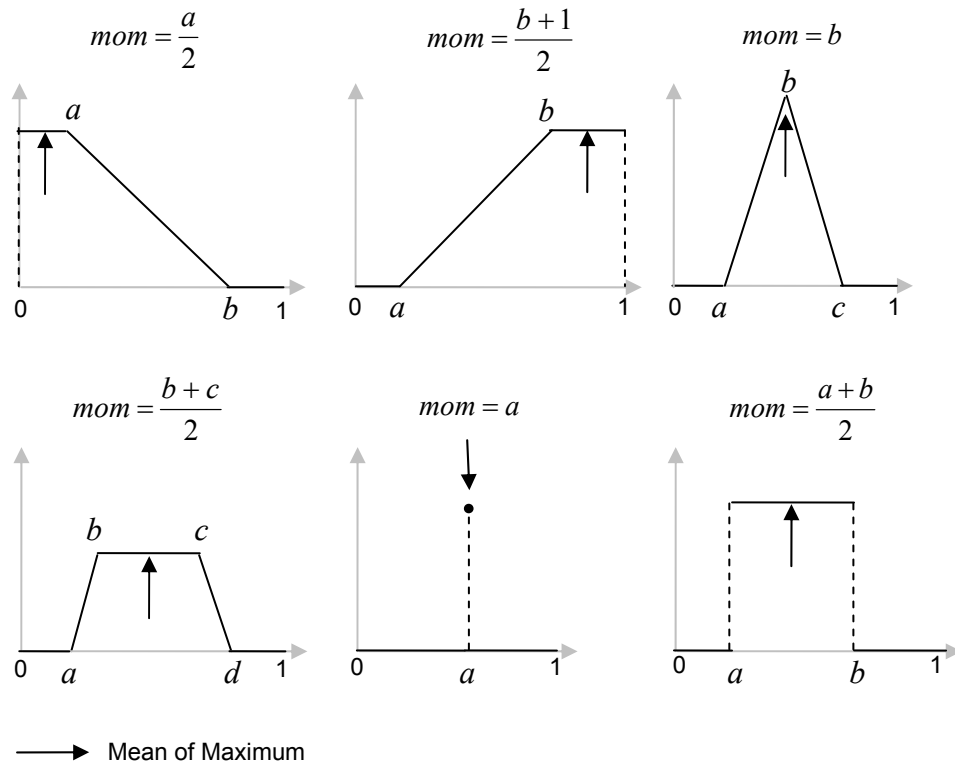


Abbildung 90: Mean of Maximum verschiedener Outputfunktionen

Das Center of Maximum bildet dann das Resultat der Defuzzifizierung und kann durch die folgende Formel berechnet werden:

$$com_B = \frac{\sum_{g=1}^q \varphi_g(\bar{x}_1, \dots, \bar{x}_n) \cdot mom(o_g)}{\sum_{g=1}^q \varphi_g(\bar{x}_1, \dots, \bar{x}_n)}$$

Verarbeitung mehrerer Stellgrößen

Die Verarbeitung mehrerer Stellgrößen gestaltet sich nun einfach. Seien B_1, \dots, B_n verschiedene Stellgrößen. Da nur jeweils ein linguistischer Wert einer Stellgröße als Konklusion jeder Implikation vorkommt, kann die Berechnung der Center of Maximum für jede Stellgröße einzeln erfolgen. Der Ausgangswert des Fuzzy-Inferenz-Systems ist dann ein n-Tupel der Form:

$$(com_{B_1}, \dots, com_{B_n}).$$

Filtern des Ausgangswertes

Das Center of Maximum ist ein Wert des Einheitsintervalls $[0;1]$. Ein Prozess-Tailorer braucht diese Genauigkeit jedoch nicht, so dass es sinnvoll ist, den Ausgangswert auf eine viel kleinere ordinale Skala abzubilden, deren Werte die letztendliche Auswahl der auswählbaren Elemente erleichtert. Diese Skala besteht aus Werten, die leichter zu interpretieren sind als das Center of Maximum selbst. Sinnvolle Skalen bestehen aus drei bis sieben Werten. Die Abbildung des Centers of Maximum kann durch eine Filterfunktion geschehen, die allgemein wie folgt definiert ist:

Definition: Sei S_{ordinal} eine ordinale Skala. Eine Filterfunktion σ ist eine monoton steigende Funktion folgender Form:

$$\begin{aligned}\sigma &: [0;1] \rightarrow S_{\text{ordinal}} \\ \sigma &: x \mapsto \sigma(x).\end{aligned}$$

Definition 23: Filterfunktion

Im Falle mehrerer Stellgrößen wird die Filterfunktionen auf jedes Center of Maximum angewandt:

$$(\sigma(\text{com}_{B_1}), \dots, \sigma(\text{com}_{B_n})).$$

Verhältnis zu auswählbaren Elementen und Projektmodell

Das Ziel eines Tailoring-Verfahrens ist es, geeignete Quality-Gate-Prozesse auf Basis einer Projektsituation zu bestimmen. Die Projektsituation ist dabei ein Element des Projektmodells (vgl. Definition 1). Für jedes auswählbare Element ist zu bestimmen, ob es Bestandteil des Quality-Gate-Prozesses ist oder nicht. Ein Fuzzy-Inferenz-System selbst arbeitet mit Eingangs- und Stellgrößen, daher ist das Verhältnis zum Projektmodell und den auswählbaren Elementen zu klären.

Die Abbildung gestaltet sich dabei relativ einfach:

- Jede Eingangsgröße A entspricht genau einem Projektmerkmal m des Projektmodells. A besitzt verschiedene linguistische Werte α_i und m eine nichtleere Menge M von Merkmalswerten. Jeder linguistische Wert α_i besitzt eine Zugehörigkeitsfunktion μ_{α_i} mit folgender Eigenschaft:

$$\mu_{\alpha_i} : M \rightarrow [0;1].$$

Das heißt, dass die Menge von Merkmalswerten von m zur Referenzmenge der Zugehörigkeitsfunktion μ_{α_i} wird.

- Jede Stellgröße entspricht genau einem auswählbaren Element.

10.3.3 Beispiel für die Unterstützung des Tailoring mit einem Fuzzy-Inferenz-System

Ein Unternehmen möchte das Tailoring mittels eines Fuzzy-Inferenz-System unterstützen. Aus Erfahrung ist bekannt, dass folgende Regeln sinnvoll sind:

1. „Wenn die Projektgröße groß ist, dann ist die Anwesenheit von Gatekeepern mit weitreichender Entscheidungsbefugnis in der Gate-Review-Sitzung unverzichtbar.“
2. „Wenn die Projektgröße mittel oder der Auftraggeber extern ist, dann ist die Anwesenheit von Gatekeepern mit weitreichender Entscheidungsbefugnis in der Gate-Review-Sitzung empfohlen.“
3. „Wenn die Projektgröße klein oder der Auftraggeber firmenintern ist, dann ist die Anwesenheit von Gatekeepern mit weitreichender Entscheidungsbefugnis in der Gate-Review-Sitzung verzichtbar.“

Der erste Schritt besteht im Aufbau des Fuzzy-Inferenz-Systems.

Aus den Regeln sind zunächst die linguistischen Variablen und ihre linguistischen Werte zu bestimmen. Die linguistischen Variablen ergeben sich in diesem Beispiel zu

$$A_1 = \text{Projektgröße}, A_2 = \text{Auftraggeber}$$

und $B =$ Notwendigkeit der Anwesenheit von Gatekeepern in der Gate-Review-Sitzung.

Die linguistischen Werte von A_1 , A_2 und B sind wie folgt definiert:

$\alpha_{1,1}$ = groß, $\alpha_{1,2}$ = mittel und $\alpha_{1,3}$ = klein

$\alpha_{2,1}$ = intern und $\alpha_{2,2}$ = extern

β_1 = unverzichtbar, β_2 = empfohlen und β_3 = verzichtbar

Für jeden der obigen linguistischen Werte ist nun eine Zugehörigkeitsfunktion zu bestimmen. Für die linguistischen Werte von A_1 können die Zugehörigkeitsfunktionen aus Beispiel 4 verwendet werden:

$$\mu_{\alpha_{1,1}} : N^0 \rightarrow [0;1] \text{ mit } \mu_{\alpha_{1,1}}(x) = \text{clramp}_{40000,60000,1}(x),$$

$$\mu_{\alpha_{1,2}} : N^0 \rightarrow [0;1] \text{ mit } \mu_{\alpha_{1,2}}(x) = \text{peak}_{5000,40000,60000,1}(x)$$

$$\text{und } \mu_{\alpha_{1,3}} : N^0 \rightarrow [0;1] \text{ mit } \mu_{\alpha_{1,3}}(x) = \text{flramp}_{5000,40000,1}(x).$$

Für die linguistischen Werte von A_2 bestehen die Definitionsmengen der Zugehörigkeitsfunktionen nur aus den Elementen „intern“ und „extern“:

$$X_2 = \{\text{intern}, \text{extern}\}.$$

Die Zugehörigkeitsfunktionen sind wie folgt definiert:

$$\mu_{\alpha_{2,1}} : X_2 \rightarrow [0;1] \text{ mit } \mu_{\alpha_{2,1}}(x) = \text{sgl}_{\text{intern},1}(x)$$

$$\text{und } \mu_{\alpha_{2,2}} : X_2 \rightarrow [0;1] \text{ mit } \mu_{\alpha_{2,2}}(x) = \text{sgl}_{\text{extern},1}(x).$$

Die Zugehörigkeitsfunktionen der linguistischen Werte der linguistischen Variablen B sind wie folgt definiert:

$$\mu_{\beta_1} : [0;1] \rightarrow [0;1] \text{ mit } \mu_{\beta_1}(y) = \text{clramp}_{0,7,0,8,1}(y),$$

$$\mu_{\beta_2} : [0;1] \rightarrow [0;1] \text{ mit } \mu_{\beta_2}(y) = \text{trap}_{0,15,0,3,0,7,0,8,1}(y)$$

$$\text{und } \mu_{\beta_3} : [0;1] \rightarrow [0;1] \text{ mit } \mu_{\beta_3}(y) = \text{flramp}_{0,15,0,3,1}(y).$$

Im nächsten Schritt ist die Regelbasis des Fuzzy-Inferenz-Systems zu bilden, die für die Fuzzy-Inferenz benötigt werden. Aus den obigen drei Regeln ergeben sich die folgenden drei Fuzzy-Regeln:

$$\mu_{\alpha_{1,1}} \Rightarrow \mu_{\beta_1}, \mu_{\alpha_{1,2}} \vee \mu_{\alpha_{2,2}} \Rightarrow \mu_{\beta_2} \text{ und } \mu_{\alpha_{1,3}} \vee \mu_{\alpha_{2,1}} \Rightarrow \mu_{\beta_3}.$$

Ein Prozess-Tailorer des Unternehmens möchte nun für die folgende Projektsituation wissen, inwieweit es erforderlich ist, dass Gatekeeper mit weitreichender Entscheidungsbefugnis am Gate-Review teilnehmen:

$$x = (55000, \text{extern}) \in N^0 \times \{\text{intern}, \text{extern}\}.$$

Die Fuzzy-Werte aus der Fuzzifizierung ergeben sich zu

$$\varphi_1 = \mu_{\alpha_{1,1}}(55000) = 0,75, \varphi_2 = \mu_{\alpha_{1,2}}(55000) \vee \mu_{\alpha_{2,2}}(\text{extern}) = 1$$

$$\text{und } \varphi_3 = \mu_{\alpha_{1,3}}(55000) \vee \mu_{\alpha_{2,1}}(\text{extern}) = 0.$$

Innerhalb der Fuzzy-Inferenz werden nun die Outputfunktionen o_g bestimmt. Im nächsten Schritt – der Defuzzifizierung – müssen die Mean of Maxima $mom(o_g)$ bestimmt werden, um letztlich das Center of Maximum bestimmen zu können. Sie können grafisch ermittelt werden, allerdings ist die genaue Berechnung vergleichsweise einfach, da alle Outputfunktionen trapezförmig verlaufen. Die Mean of Maxima berechnen sich zu

$$mom(o_1) = 0,8875, mom(o_2) = 0,5 \text{ und } mom(o_3) = 0,5.$$

Das Center of Maximum ergibt sich damit zu

$$com_B \approx 0,67.$$

Das Center of Maximum ist der scharfe Ausgangswert des Fuzzy-Inferenz-Systems. Dieser kann nun durch eine Filterfunktion auf eine leichter zu interpretierende ordinale Skala abgebildet werden. Hierzu wird die folgende Filterfunktion genutzt:

$$\sigma : [0;1] \rightarrow \{\text{rot, orange, gelb, hellgrün, grün}\}$$

$$\sigma(x) = \begin{cases} \text{rot,} & \text{für } 0 \leq x \leq 20 \\ \text{orange,} & \text{für } 20 < x \leq 40 \\ \text{gelb,} & \text{für } 40 < x \leq 60 \\ \text{hellgrün,} & \text{für } 60 < x \leq 80 \\ \text{grün,} & \text{für } 80 < x \leq 100 \end{cases}$$

Die Filterfunktion liefert damit $\sigma(0,67) = \text{hellgrün}$ und signalisiert dem Prozess-Tailorer, dass die Anwesenheit von Gatekeepern mit weitreichender Entscheidungsbefugnis im Gate-Review nahe gelegt wird.

10.3.4 Unterstützung des Tailorings durch Softwarewerkzeuge

Das Beispiel aus Abschnitt 10.3.3 zeigt, dass sich die Berechnung des Centers of Maximum selbst bei einer kleinen Anzahl linguistischer Variablen aufwändig gestaltet. Ferner müssen Zugehörigkeitsfunktionen und Regeln verwaltet werden. Die Unterstützung durch ein Softwarewerkzeug erscheint daher sinnvoll. Im Rahmen einer Masterarbeit [Peucker '07] wurde ein solches Werkzeug im Fachgebiet Software Engineering der Gottfried Wilhelm Leibniz Universität Hannover entwickelt – der so genannte *Quality-Gate-Tailoring-Assistent* (kurz: *QGT*).

Der QGT ermöglicht das Fuzzy-Tailoring eines Quality-Gate-Referenzprozesses auf Grundlage einer Projektsituation (vgl. Abbildung 91). Dabei kann jeweils entschieden werden, ob ein Projektmerkmal bei der Anpassung berücksichtigt werden soll oder nicht.

Projektgegenstand		3	{Embedded, Komplex, Software, Integration}	<input checked="" type="checkbox"/>
Projektrolle		6	{AG:AN} {1:1, 1:N, 1:1:U, 1:1:O, 0:1:U, 0:1:O}	<input checked="" type="checkbox"/>
Systemlebenszyklusausschnitt		1	{Entwicklung, Wartung, Weiterentwicklung}	<input checked="" type="checkbox"/>
Fertigprodukte		0	{nein, ja}	<input checked="" type="checkbox"/>
Hohe Realisierungsrisiken		1	{nein, ja}	<input checked="" type="checkbox"/>
Safety und Security		0	{nein, ja}	<input checked="" type="checkbox"/>
Projekttyp		2	{AG, AN, AG/AN}	<input checked="" type="checkbox"/>

Abbildung 91: Eingabe der Projektsituation im QGT (am Beispiel V-Modell XT)

Das Ergebnis der Defuzzifizierung wird baumartig dargestellt (Abbildung 92). Auswählbare Elemente können dabei in verschiedene Kategorien (diese können zum Beispiel den anpassbaren Quality-Gate-Teilkonzepten entsprechen) eingeteilt werden, um die Übersicht zu erhöhen. Für jede Stellgröße kann ferner der Einfluss jeder einzelnen Eingangsgröße auf eine Stellgröße dargestellt werden (aufgeklappter Zweig des Baumes), um die Nachvollziehbarkeit des Tailoring-Verfahrens zu erhöhen. Die Darstellung orientiert sich hierbei an dem Fuzzy-Tailoring-Werkzeug nach Ittner [Ittner '05].



Abbildung 92: Darstellung des Tailoring-Ergebnisses (am Beispiel V-Modell XT)

Der QGT implementiert eine Filterfunktion mit einer Wertemenge

$$S_{\text{ordinal}} \subset \{\text{rot, orange, gelb, hellgrün, grün}\}.$$

Auswählbare Elemente mit roter Bewertung werden automatisch deaktiviert (markiert durch Kreuz), während auswählbare Elemente mit grüner Bewertung automatisch aktiviert werden (markiert durch Haken). Alle Farben dazwischen werden mit einem Fragezeichen markiert und erfordern eine Entscheidung durch den Prozess-Tailorer. Falls notwendig können jedoch auch rot und grün markierte Elemente aktiviert bzw. deaktiviert werden.

Die Verwaltung der Zugehörigkeitsfunktionen und Fuzzy-Regeln erfolgt über eine grafische Oberfläche (Abbildung 93).

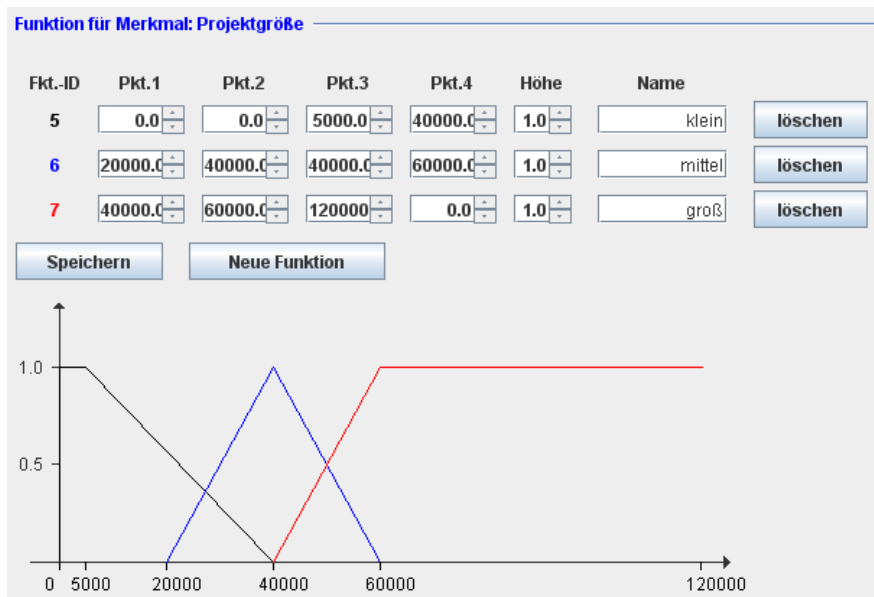


Abbildung 93: Eingabe einer Zugehörigkeitsfunktion

Das Ergebnis des Tailorings, also der Quality-Gate-Prozess wird erst sichtbar, wenn alle Entscheidungen getroffen sind (Abbildung 94).



Abbildung 94: Quality-Gate-Prozess als Ergebnis des Tailorings (am Beispiel V-Modell XT)

10.3.5 Einschränkungen und Herausforderungen eines Fuzzy-Inferenz-Systems

Die Unterstützung des Tailorings durch das vorgestellte Fuzzy-Inferenz-System weist verschiedene Einschränkungen und Herausforderungen auf. Sie sollen im Folgenden erläutert werden:

- Aufgrund der Bildung des gewichteten arithmetischen Mittels und der Bildung des Mean of Maximum ist die Wahrscheinlichkeit gering, dass das Center of Maximum genau die Randwerte 0 oder 1 annimmt. Stochastisch gesehen besitzen sie als Einzelwerte ohnehin die Auftretenswahrscheinlichkeit 0. Folglich sind scharfe Ausgangswerte, die hundertprozentig klar auf die Auswahl oder Abwahl eines auswählbaren Elementes deuten, unwahrscheinlich. Durch die

Nutzung einer Filterfunktion, durch die das Einheitsintervall $[0;1]$ in Teilintervalle partitioniert kann, wird diese Problematik entschärft.

- Falls andere Zugehörigkeitsfunktionen als in Abbildung 87 verwendet werden (insbesondere also nicht lineare Funktionen), so wird die Bestimmung des Mean of Maximum erschwert. Viele Aussagen lassen sich zunächst (zumindest stückweise) linear annehmen, ob jedoch Aussagen im Bereich des Prozess-Tailorings bestehen, die nicht durch derartige Funktionen modelliert werden können, kann in dieser Arbeit nicht zweifelsfrei geklärt werden.
- Das vorgestellte Fuzzy-Inferenz-System beachtet nur Stellgrößen, deren linguistische Werte in einer Regel als Konklusion vorhanden sind. Ein auswählbare Element kann nur dann beim Tailoring berücksichtigt werden, wenn es in der Konklusion einer Regel vorkommt. Ansonsten kann keine Aussage über die Ab- oder Auswahl des Elementes getroffen werden. In diesem Fall gilt es als abgewählt.
- Das vorgestellte Fuzzy-Inferenz-System ermöglicht nicht die Modellierung von Wechselwirkungen zwischen den Stellgrößen. Sind B_1 und B_2 Stellgrößen und β_1 und β_2 jeweilige Stellgrößen, so kann beispielsweise eine Implikation der Form

$$\beta_1 \Rightarrow \beta_2$$

nicht formuliert werden. Die Verantwortung, derartige Wechselwirkungen zu erkennen, liegt beim Gate-Management und in letzter Instanz beim Prozess-Tailorer.

- Zum Aufbau eines Fuzzy-Inferenz-Systems wird eine Vielzahl von Zugehörigkeitsfunktionen benötigt. Diese Zugehörigkeitsfunktionen beruhen auf Erfahrung und sind zu validieren. Der Aufbau eines Fuzzy-Inferenz-Systems benötigt also Ressourcen. Es soll ein Fuzzy-Inferenz-System mit n Eingangsgrößen und m Stellgrößen betrachtet werden. Ist n_{\max} die maximale Anzahl von linguistischen Werten der Eingangsgrößen und m_{\max} die entsprechende Anzahl für die Stellgrößen, dann besitzt ein Fuzzy-Inferenz-System mindestens

$$n + m,$$

maximal

$$n \cdot n_{\max} + m \cdot m_{\max}$$

und im Mittel

$$n \cdot \frac{2 + n_{\max}}{2} + m \cdot \frac{2 + m_{\max}}{2}$$

Zugehörigkeitsfunktionen. Für den mittleren Fall sei dabei angenommen, dass jede linguistische Variable mindestens zwei linguistische Werte besitzt.

Ein Fuzzy-Inferenz-System mit den Parametern

$$n = 10, n_{\max} = 4, m = 30 \text{ und } m_{\max} = 4$$

besitzt damit also mindestens 40, maximal 160 und im Mittel 120 Zugehörigkeitsfunktionen.

10.3.6 Umgang mit den Herausforderungen und Einschränkungen

Von den genannten Herausforderungen und Einschränkungen stellt insbesondere der Umgang mit der Vielzahl an benötigten Zugehörigkeitsfunktionen ein großes Problem für das Gate-Management dar. Auf den Umgang mit den anderen Herausforderungen und Einschränkungen wurde bereits im vorherigen Abschnitt eingegangen.

Ein Unternehmen, das einen anpassbaren Quality-Gate-Referenzprozess auf Basis des Fuzzy-Inferenz-Systems von Grund auf ausgestalten möchte, sieht sich der Herausforderung gestellt, eine große Anzahl von Zugehörigkeitsfunktionen und basierend darauf Regeln erstellen zu müssen. Um dies erfolg-

reich zu meistern, existieren drei Strategien, die verschiedene vermutete Vor- und Nachteile aufweisen:

- **Maximale Anpassungsstrategie:** Alle Zugehörigkeitsfunktionen werden vor dem ersten Einsatz des Quality-Gate-Referenzprozess so gut wie möglich aus den bestehenden Erfahrungen hergeleitet. Dieses Vorgehen ist jedoch eher aufwändig, da zunächst Erfahrungen idealerweise durch einen speziellen Experience-Engineer in einem vergleichsweise kurzen Zeitraum aktiviert, gesammelt und aufbereitet, gespeichert und als Projektmodell, auswählbare Elemente, Zugehörigkeitsfunktionen und Regeln wiederverwendet werden müssen (vgl. hierzu Abschnitt 11.1). Idealerweise liegt eine ausreichende Menge von Erfahrungen schon in fester Form vor, weil beispielsweise bereits eine Experience Base genutzt wird. Nachteilig ist weiterhin, dass der Quality-Gate-Referenzprozess vergleichsweise spät genutzt werden kann. Der Vorteil liegt darin, dass in jedem Projekt ein vergleichsweise passender Quality-Gate-Prozess unmittelbar nach Ausgestaltung des Quality-Gate-Referenzprozess eingesetzt werden kann. Eine Pflege durch das Gate-Management ist notwendig, fällt jedoch weniger aufwändig als bei den anderen beiden Strategien aus.
- **Boolesche Anpassungsstrategie:** Alle Zugehörigkeitsfunktionen werden zunächst als uneigentliche Fuzzy-Funktionen modelliert und basierend hierauf Regeln erstellt. Dies bedeutet, dass die Möglichkeit zur unscharfen Modellierung vollständig ausgeblendet wird. Hierbei kann nicht völlig auf Erfahrungen verzichtet werden, da die Regeln ja nicht willkürlich gewählt werden können. Die Zugehörigkeitsfunktionen werden im Rahmen der Pflege durch das Gate-Management nach und nach in (falls dies sinnvoll ist) eigentliche Fuzzy-Funktionen transferiert. Der Nachteil hierbei ist, dass ein Prozess-Tailorer zunächst mit gewisser (vermutlich jedoch vergleichsweise geringer) Wahrscheinlichkeit ungeeignete Quality-Gate-Prozesse als Ergebnis des Fuzzy-Inferenz-Systems erhält und daher von Hand anpassen muss. Im Rahmen dieser nachgeordneten Anpassung durch den Prozess-Tailorer können jedoch Erfahrungen gewonnen werden, die wiederum zur Verbesserung der Zugehörigkeitsfunktionen und Regeln genutzt werden können. Der Vorteil ist, dass die Erstellung der Zugehörigkeitsfunktion und Regeln schneller erfolgt und damit auch der Einsatz des Quality-Gate-Referenzprozesses schneller erfolgen kann.
- **Minimale Anpassungsstrategie:** Die letzte Strategie benötigt zunächst keine Erfahrungen, da der gleiche Quality-Gate-Prozess in jedem Projekt genutzt wird. Nach und nach können jedoch aufgrund abgeschlossener oder fehlgeschlagener Projekte Erfahrungen zur Ausgestaltung des Tailoring-Verfahrens genutzt werden. Von den drei vorgestellten Strategien ist die *Minimale Anpassungsstrategie* diejenige Strategie mit dem geringsten anfänglichen Aufwand; ein Ersteinsatz ist daher wesentlich eher möglich. Nachteilig ist, dass der Quality-Gate-Prozess vermutlich nicht zu jedem Projekt passt und daher vom Prozess-Tailorer im größeren Maße von Hand angepasst werden muss. Diese händische Anpassung kann aber – ähnlich wie bei booleschen Anpassungsstrategie – zum Erfahrungsgewinn genutzt werden. Die *Minimale Anpassungsstrategie* ist ebenfalls gut geeignet, wenn viele Erfahrungen erst in zukünftigen Projekten gesammelt werden können und Erfahrungen aus vergangenen Projekten kaum dokumentiert sind.

Gleich welche Anpassungsstrategie durch ein Unternehmen verfolgt wird, so ist bei konsequent und kontinuierlich durchgeführten Erfahrungskreisläufen über einen längeren Zeitraum (mehrere Jahre) vermutlich folgendes zu erwarten:

- Alle Strategien führen letztlich zu einem Fuzzy-Inferenz-System gleicher Reife.
- Der Gesamtaufwand zur Erstellung des Fuzzy-Inferenz-Systems ist identisch.

Abbildung 95 a) zeigt für alle drei Strategien die hypothetische Entwicklung der Ressourcen, die für die Entwicklung eines Fuzzy-Inferenz-Systems aufgewendet werden müssen. Die eingeschlossene Fläche unter jeder der Aufwandsfunktionen ist gleich und zeigt, dass der Aufwand jeweils gleich ist. Abbildung 95 b) zeigt die daraus resultierende Reife des Fuzzy-Inferenz-Systems, die sich jeweils durch eine Integration der Aufwandsfunktionen ergibt.

Der Funktionsverlauf lässt jeweils wie folgt interpretieren:

- Der Aufwand zur Erstellung des Fuzzy-Inferenz-Systems bei der Strategie *Maximale Anpassungsstrategie* ist zunächst sehr hoch, sinkt schnell jedoch ab. Erst wenn die Reife des Fuzzy-Inferenz-Systems sehr hoch ist, kann der Quality-Gate-Referenzprozess erstmalig eingesetzt werden. Ab diesem Zeitpunkt ist der notwendige Aufwand zur Pflege des Fuzzy-Inferenz-Systems sehr gering.
- Der Aufwand zur Erstellung eines Fuzzy-Inferenz-Systems bei der Strategie *Boolesche Anpassungsstrategie* ist zunächst mittelhoch, da lediglich uneigentliche Fuzzy-Funktionen erstellt werden müssen. Der Aufwand sinkt vergleichsweise langsam ab, da der Aufwand zur Pflege nach erstmaligem Einsatz des Quality-Gate-Referenzprozesses entsprechend hoch ist.
- Der Aufwand zur Erstellung des Fuzzy-Inferenz-Systems bei der Strategie *Minimale Anpassungsstrategie* ist anfangs sehr gering, da kein Fuzzy-Inferenz-System erstellt werden muss. Nach dem Ersteinsatz des Quality-Gate-Referenzprozesses steigt der Aufwand jedoch an, da das Fuzzy-Inferenz-System mit den ersten Erfahrungen von Grund auf neu erstellt werden muss. Danach sinkt der Aufwand ähnlich wie bei der Strategie *Boolesche Anpassungsstrategie*.

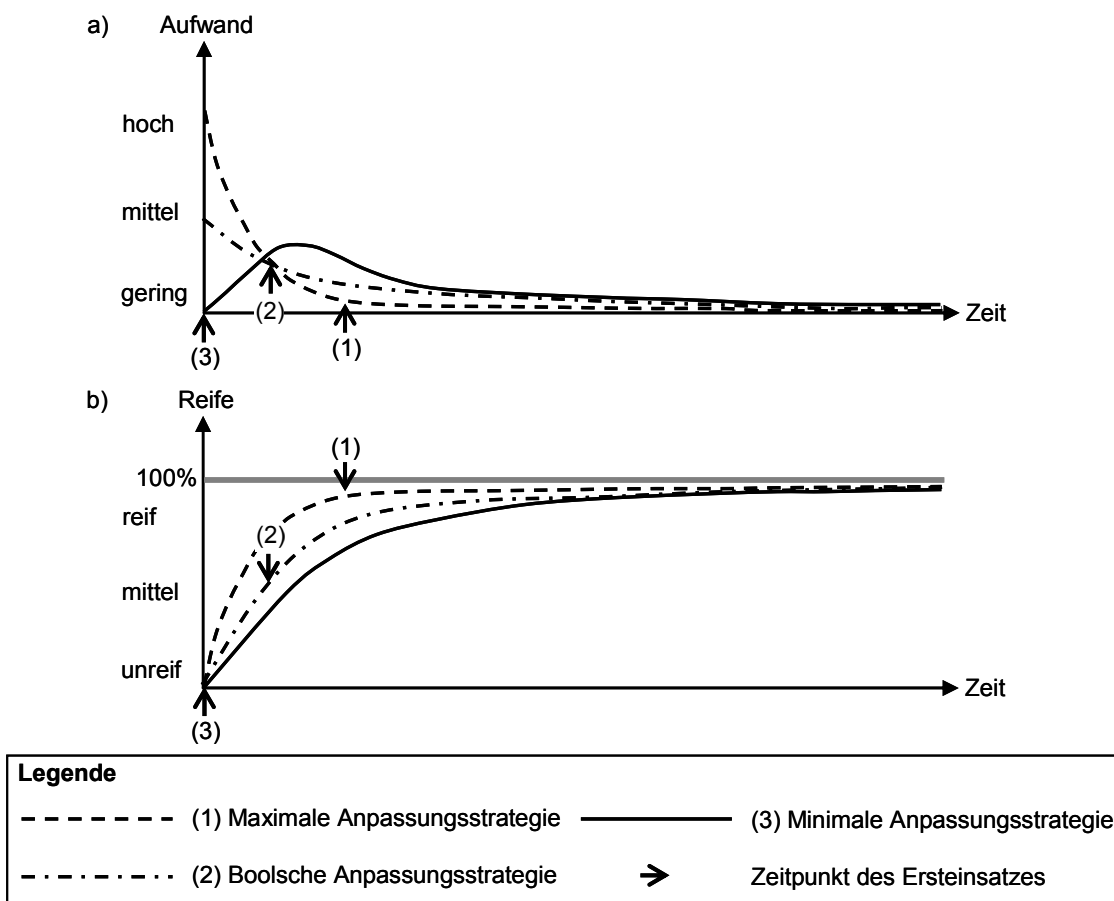


Abbildung 95: Hypothetische Entwicklung des Aufwands und der Reife

10.4 Auswählbare Elemente

Das Prozessmanagement eines Unternehmens, das einen Quality-Gate-Referenzprozess auf Grundlage des Frameworks ausgestalten möchte, muss auch die auswählbaren Elemente des Quality-Gate-Referenzprozesses definieren, falls der Quality-Gate-Prozess von der Projektsituation abhängt und das Tailoring-Verfahren formalisiert werden soll. Die auswählbaren Elemente hängen dabei von der Ausgangssituation des Unternehmens ab.

Die weiteren Betrachtungen dieses Abschnitts beziehen sich auf das Fuzzy-Inferenz-System (Abschnitt 10.3) als ein generisches formales Tailoring-Verfahren.

Die auswählbaren Elemente eines Quality-Gate-Referenzprozesses lassen sich durch die Quality-Gate-Teilkonzepte des abstrahierenden Domänenmodells kategorisieren. Die Kategorisierung ist eindeutig (vgl. Abbildung 96).

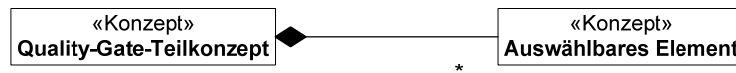


Abbildung 96: Auswählbare Elemente können kategorisiert werden

Damit lässt sich der Begriff des auswählbaren Quality-Gate-Teilkonzeptes definieren.

Definition: Ein anpassbares Quality-Gate-Teilkonzept ist ein Konzept, das mindestens ein auswählbares Element besitzt.

Definition 24: Anpassbares Quality-Gate-Teilkonzept

Beispielsweise ist die Anwesenheit von Gatekeepern mit weitreichender Entscheidungsbefugnis im Gate-Review ein mögliches auswählbares Element des Quality-Gate-Teilkonzeptes Gate-Review. Weitere auswählbare Elemente des Gate-Reviews sind der Schreibtischtest, das Walkthrough und die Inspektion.

Des Weiteren wird nun der Begriff der möglichen Variante eines Quality-Gate-Teilkonzeptes eingeführt.

Definition: Sei Q ein anpassbares Quality-Gate-Teilkonzept und $A_Q = \{a_1, \dots, a_n\}$ die Menge der auswählbaren Elemente von Q . Eine mögliche Variante des Quality-Gate-Teilkonzeptes Q ist eine Teilmenge \bar{A}_Q der Potenzmenge von A_Q :

$$\bar{A}_Q \subseteq P(A_Q).$$

Definition 25: Mögliche Variante eines Quality-Gate-Teilkonzeptes

Leider ist nicht jede mögliche Variante auch widerspruchsfrei. Beispielsweise kann im selben Quality Gate nicht gleichzeitig das Gate-Review durch einen Schreibtischtest und eine formale Inspektion realisiert werden. Darüber hinaus muss die Variante vollständig sein. So kann ein Gate-Review nur durch genau eine der obigen Sitzungsarten realisiert werden.

Aus diesem Grund wird nun noch der Begriff der gültigen Variante definiert.

Definition: Sei $A_Q = \{a_1, \dots, a_n\}$ die Menge der auswählbaren Elemente eines anpassbaren Quality-Gate-Teilkonzeptes Q . Eine gültige Variante (oder im Folgenden auch kurz Variante) des Quality-Gate-Teilkonzeptes Q ist eine Teilmenge \bar{A}_Q der Potenzmenge A_Q :

$$\bar{A}_Q \subseteq P(A_Q)$$

mit den folgenden Eigenschaften:

$$\forall i, j \in \{1, \dots, |\bar{A}_Q|\} \text{ mit } i \neq j \text{ gilt: } \bar{a}_i, \bar{a}_j \in \bar{A}_Q \text{ sind widerspruchsfrei}$$

und

$$\bar{A}_Q \text{ ist vollständig.}$$

Definition 26: Gültige Variante eines Quality-Gate-Teilkonzeptes

Ob eine gültige Variante vorliegt kann in letzter Instanz – falls sie nicht schon durch das Tailoring-Verfahren geliefert wird – nur durch den Prozess-Tailorer bestimmt werden, da Wechselwirkungen zwischen Stellgrößen nicht berücksichtigt werden (vgl. Abschnitt 10.3.5). Das folgende Beispiel illustriert gültige Varianten für das Quality-Gate-Teilkonzept Gate-Review.

Beispiel: Für das Gate-Review sollen alle gültigen Varianten bestimmt werden. Auswählbare Elemente sind

a_1 = Anwesenheit von Gatekeepern mit weitreichender Entscheidungsbefugnis im Gate-Review,

a_2 = Schreibtischtest, a_3 = Walkthrough und a_4 = formale Inspektion.

Die Menge der auswählbaren Elemente ergibt sich also zu

$$\{a_1, a_2, a_3, a_4\}.$$

Gültige Varianten sind dann

$$\{a_1, a_2\}, \{a_1, a_3\}, \{a_1, a_4\}, \{a_2\}, \{a_3\} \text{ und } \{a_4\}.$$

Folglich können alle Sitzungsarten mit und ohne Gatekeeper mit weitreichender Entscheidungsbefugnis durchgeführt werden.

Beispiel 5: Beispiel für gültige Varianten

Die folgende Tabelle 39 zeigt für jedes anpassbare Quality-Gate-Teilkonzept die auswählbaren Elemente.

Paket	Anpassbares Teilkonzept	Auswählbare Elemente
Struktur	Gate-Netzwerk	Auswahl, Anzahl und Anordnung von Quality Gates
	Inhalt	Kriterienerstellung
Review	Ersteller	Beteiligte Rollen an der Kriterienerstellung
	Checkliste	Kriterien und Ergebnisse (nur bei festen Kriterienkatalog mit Auswahl), Operationalisierung der Kriterien
	Gate-Review	Variante des Gate-Reviews, Anwesenheit von Gatekeepern mit weitreichender Entscheidungsbefugnis, Notwendigkeit eines separaten Gate-Verantwortlichen
	Gutachter	Anzahl der Gutachter im Gate-Review, Erfahrung und Fachkenntnisse der Gutachter
	Projektvertreter	Rollen, die das Projekt im Gate-Review vertreten
Steuerung	Protokoll	Ausführlichkeit des Protokolls
	Gate-Verantwortlicher	Notwendige Fachkenntnisse und Erfahrung des Gate-Verantwortlichen
	Gatekeeper	Anzahl, Erfahrung und Fachkenntnisse der Gatekeeper

Tabelle 39: Übersicht der auswählbaren Elemente

Nicht jedes Teilkonzept ist bei jeder der beiden grundlegenden Strategien für Quality Gates anpassbar. Insbesondere ist bei der Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* das Gate-Netzwerk nicht anpassbar. Checklisten sind nur dann anpassbar, wenn die Auswahl von Kriterien erlaubt ist. Weiterhin ist die Ausführlichkeit des Protokolls anpassbar. Alle weiteren Teilkonzepte sind hingegen nicht anpassbar, da ansonsten die Vergleichbarkeit zwischen den Projekten erschwert wird. Bei der

Strategie *Quality Gates als flexible Qualitätsstrategie* sind hingegen alle Teilkonzepte anpassbar. Tabelle 40 fasst zusammen, welche Teilkonzepte jeweils anpassbar sind.

		Anpassbare Teilkonzepte									
		Gate-Netzwerk	Kriterien-erstellung	Ersteller	Checkliste	Gate-Review	Gutachter	Projektvertreter	Protokoll	Gate-Verantwortlicher	Gatekeeper
Strategie	Quality Gates als einheitliche Qualitäts-Leitlinie	–	–	–	X*	–	–	–	X	–	–
	Quality Gates als flexible Qualitätsstrategie	X	X	X	X	X	X	X	X	X	X
Legende											
–		nicht anpassbar									
X		Anpassbar beim Projektstart und während der Durchführungsphase									
X*		Anpassbar beim Vorgehen <i>Kriterienkatalog mit Auswahl</i>									

Tabelle 40: Anpassbare Teilkonzepte nach Strategie

10.5 Projektmodell

Ein Quality-Gate-Prozess liegt bereits nach dem Projektstart zumindest hinsichtlich seines Gate-Netzwerkes verbindlich fest. Er ist anhand der Projektsituation bestimmt worden, die in der Phase Projektstart ermittelt wurde. Allerdings sind zu einem derartig frühen Zeitpunkt nur wenige Merkmalswerte des Projektes bekannt, so dass die Projektsituation nur eine grobe Beschreibung des Projektes darstellt. Erst in der Planungsphase oder in der Durchführungsphase kann die Projektsituation genauer beschrieben werden und eine genauere Anpassung der weiteren Quality-Gate-Teilkonzepte, insbesondere der Checkliste vorgenommen werden.

Tabelle 41 fasst die relevanten Projektmerkmale mit ihren relevanten Projektmerkmalswerten zusammen.

Qualitativ/ Quantitativ	Projektmerkmal	Mögliche Merkmalswerte
Quantitativ (nominale Skalen)	Mission des Projektes	Neuentwicklung, Wartung, Migration, Weiterentwicklung und Integrationsprojekt
	Projektgegenstand	Eingebettete Software, Komplexes System, andere Software
	Konkreter Einsatzzweck	Automotive, Medizin, Öffentlicher Dienst/Behörden, Versicherungen/Banken, Internet/E-Commerce, betriebliche Anwendungssoftware etc.
	Benötigte Technologien und Methoden	Programmiersprachen, Zielarchitekturen, Werkzeuge Methoden und weitere Technologien
	Qualitäts-Anforderungen	Effizienz, Flexibilität, Integrität, Korrektheit, Portabilität, Testbarkeit, Wartbarkeit, Wiederverwendbarkeit, Verwendbarkeit, Zuverlässigkeit
Qualitativ (ordinale Skalen) oder Quantitativ	Neuartigkeit der Technologien und Methoden	bekannt, größtenteils bekannt, größtenteils unbekannt, unbekannt
	Benötigte Ressourcen und Umfang	Zeitlicher, monetärer und personeller Rahmen und Umfang des Projektes
	Stabilität der Anforderungen	stabil, eher stabil, eher instabil, instabil
	Strategische Relevanz für das Unternehmen	unwichtig, wichtig, sehr wichtig, überlebenswichtig
	Risiko des Projektes (Wahrscheinlichkeit des Scheiterns des Projektes)	gering, möglich, hoch, sehr hoch
	Potenzielles Schadensausmaß im Falle eines Softwarefehlers	keiner, gering, finanzieller Schaden, Verletzung von Menschen, ein Menschenleben, viele Menschenleben

Tabelle 41: Relevante Projektmerkmalswerte mit Merkmalswerten

Dementsprechend sind folgende Merkmalswerte beim Projektstart nicht bekannt, sie können jedoch in der Durchführungsphase zur weiteren Anpassung genutzt werden:

- Genauere Schätzungen zum Umfang, zum Beispiel in Function Points, Lines of Codes oder Mannstunden.
- Genaueres Wissen über die notwendigen Technologien und damit zu möglichen Risiken.
- Ein genauerer Überblick über die Anforderungen.

10.6 Beziehungen zwischen Projektmerkmalen und anpassbaren Quality-Gate-Teilkonzepten

Zwischen den Projektmerkmalen eines Projektmodells und den anpassbaren Quality-Gate-Teilkonzepten bestehen quantitative und qualitative Einflussbeziehungen bei der Anpassung. Die Einflussbeziehungen beschreiben inwieweit und ob ein Projektmerkmal und ein anpassbares Quality-Gate-Teilkonzept in miteinander in Beziehung stehen.

Eine qualitative Einflussbeziehung liegt dann vor, wenn die Auswahl eines Merkmalswerts eines Projektmerkmals m die Variante des Quality-Gate-Teilkonzeptes beeinflusst. Beispielsweise führt die Auswahl der Qualitäts-Anforderung „Testbarkeit“ dazu, dass ein leitender Testverantwortlicher bei

der Kriterienerstellung für das Quality Gate „Anforderungsphase abgeschlossen“ anwesend sein muss. Eine quantitative Beziehung liegt hingegen vor, wenn ausgeprägtere Merkmalswerte zu einer aufwändigeren (und damit auch anderen) Variante eines Quality-Gate-Teilkonzeptes führen. So führt zum Beispiel ein strategisch relevanteres Projekt auch zu einem Gate-Netzwerk mit einer höheren Anzahl an Quality Gates. Bei einer qualitativen Einflussbeziehung sind die Merkmalswerte eines Projektmerkmals m Elemente einer nominalen Skala. Bei einer quantitativen Einflussbeziehung müssen die Elemente mindestens einer ordinalen Skala entstammen, da nur dann festgestellt werden kann, ob ein Merkmalswert ausgeprägter ist als ein anderer. Tabelle 42 fasst alle Einflussbeziehungen zusammen.

Ein Projekt kann nun durch eine Projektsituation beschrieben werden. Hinsichtlich der Anpassung gilt, dass ein umfangreicheres, risikobehafteteres, wichtigeres, neuartigeres, sicherheitskritischeres Projekt mit instabileren Anforderungen auch einen aufwändigeren Quality-Gate-Prozess benötigt. Die Projektsituation kann (zum Teil) durch ein Diagramm beschrieben werden (vgl. Abbildung 97). In der Situation, wie sie Abbildung 97 dargestellt wird, wird ein Tailoring-Verfahren für Projekt C den aufwändigsten Quality-Gate-Prozess liefern.

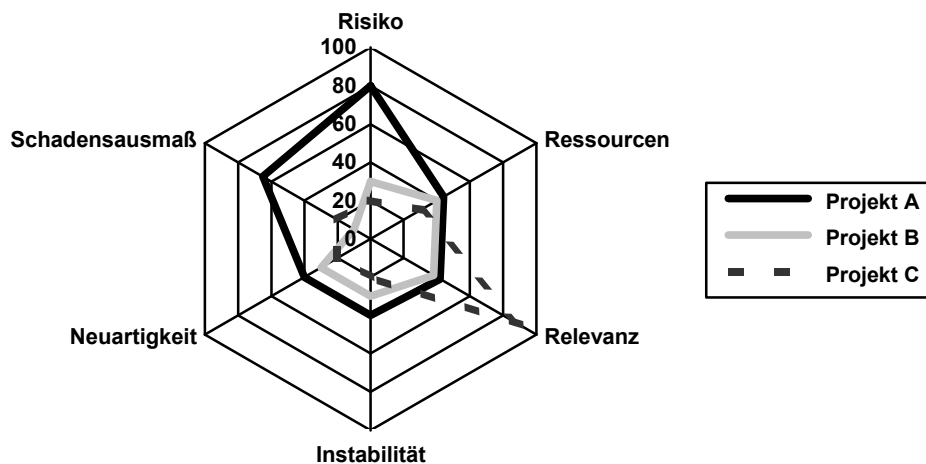


Abbildung 97: Einordnung verschiedener Projekte

Eine aufwändigere Variante eines Quality-Gate-Teilkonzeptes schlägt sich je nach Quality-Gate-Konzept verschieden nieder:

- Das Gate-Netzwerk weist mehr Quality Gates auf.
- Rollen werden durch mehr Personen besetzt. Personen müssen über ein größeres Fachwissen und Entscheidungsbefugnis besitzen.
- Kriterienerstellung und Gate-Review werden systematischer und aufwändiger durchgeführt.
- Das Protokoll ist ausführlicher.
- Es sind mehr Kriterien mit härteren Sollwerten und systematischeren und aufwändigeren Messungen notwendig.

		Anpassbare Teilkonzepte									
		Gate-Netzwerk	Kriterien-erstellung	Ersteller	Checkliste	Gate-Review	Gutachter	Projektvertreter	Protokoll	Gate-Verantwortlicher	Gatekeeper
Projektmerkmale	Mission	◇	–	–	◇	–	–	–	–	–	–
	Projektgegenstand	◇	–	◇	◇	–	◇	◇	–	–	◇
	Einsatzzweck	◇	–	◇	◇	–	◇	◇	–	–	◇
	Technologien und Methoden	–	–	◇	◇	–	◇	◇	–	–	◇
	Qualitäts-Anforderungen	–	–	◇	◇	–	◇	◇	–	–	◇
	Neuartigkeit der Technologien und Methoden	▲	◇/▲	▲	▲	▲	▲	▲	▲	▲	▲
	Ressourcen und Umfang	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
	Stabilität der Anforderungen	◇/▲	–	–	–	–	–	–	–	–	–
	Relevanz	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
	Risiko des Projektes	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
	Potenzielles Schadensausmaß	◇/▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
Legende											
◇ Qualitativer Einfluss ▲ Quantitativer Einfluss – kein Einfluss											

Tabelle 42: Beziehungen zwischen Merkmalen und anpassbaren Quality-Gate-Teilkonzepten

Mehrere Einflussbeziehungen verhalten sich sowohl qualitativ als auch quantitativ. Instabile Anforderungen legen ein iteratives bzw. agiles Vorgehen nahe. Dementsprechend muss auch das Gate-Netzwerk dem agilen Vorgehen angepasst werden. Hingegen legt die Entwicklung von sicherheitskritischer Software (also Software mit hohen potenziellen Schadensausmaß) ein wasserfallartiges und dokumentenzentriertes Vorgehen und damit ein entsprechendes Gate-Netzwerk nahe. Gleichzeitig sind jedoch mit erhöhter Sicherheitskritikalität und Instabilität der Anforderungen mehr Quality Gates notwendig.

Die Neuartigkeit der eingesetzten Techniken und Methoden verlangt es eventuell, Kriterien individuell erstellen zu können. In diesem sind also eher Kriterien unabhängig von einem Kriterienkatalog zu erstellen oder das Vorgehen Kriterienkatalog mit Ergänzung zu wählen. Die Einflussbeziehung zwischen der Neuartigkeit der eingesetzten Techniken und Methoden und der Kriterienerstellung ist also qualitativ und quantitativ.

10.7 Wechselwirkung zur Anpassung des Entwicklungs-Referenzprozesses

Die Anpassung eines Quality-Gate-Referenzprozesses kann nicht losgelöst von der Anpassung des Entwicklungs-Referenzprozesses betrachtet werden. So kann beispielsweise ein Produkt, das im Entwicklungsprozess nicht erstellt wird, weil es für das Projekt nicht benötigt wird, auch nicht im Quality-Gate-Prozess geprüft wird. Andersherum kann jedoch auch argumentiert werden, dass ein, vom Quality-Gate-Prozess gefordertes Produkt, vom Entwicklungsprozess zu liefern ist, damit eine bestimmte Qualität eingehalten werden kann (Steuerungsaspekt des Quality-Gate-Prozesses). Diese Diskrepanz ist durch den Prozess-Tailorer aufzulösen.

Abbildung 98 zeigt schematisch die Wechselwirkung bei der Anpassung.

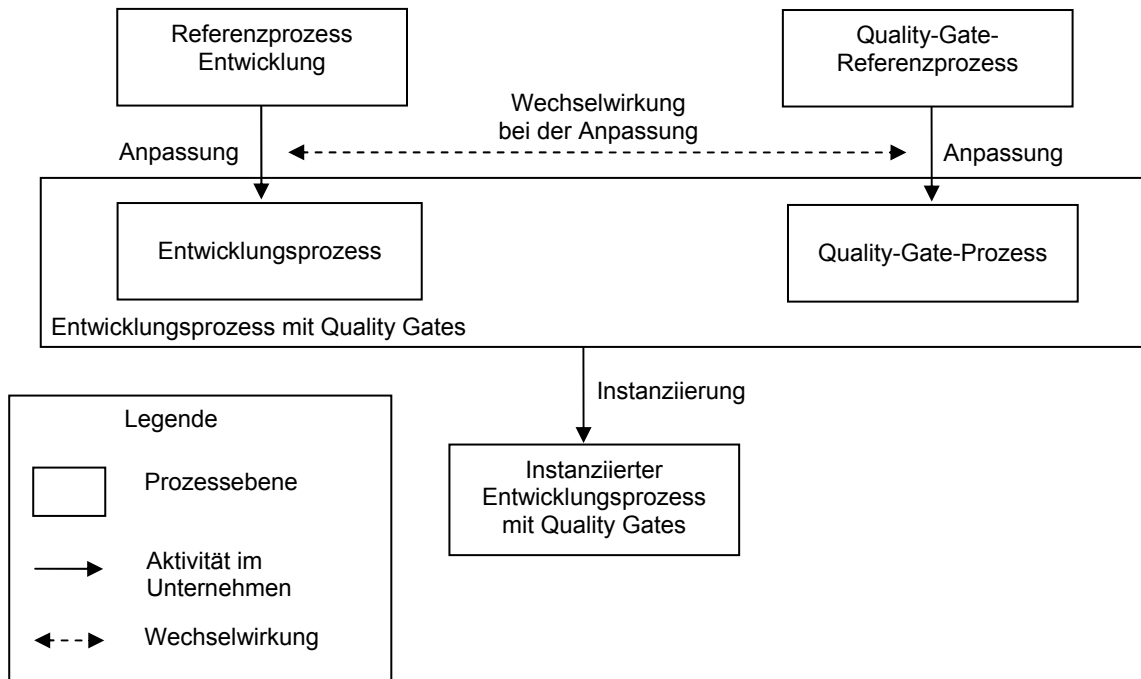


Abbildung 98: Schematische Einordnung der Wechselwirkung bei der Anpassung

Die folgende Auflistung zeigt die Wechselwirkungen, die sich bei der Anpassung des Quality-Gate-Referenzprozesses ergeben.

- **Checkliste:** Es können nur Kriterien geprüft werden, die sich auf Ergebnisse beziehen, die auch erstellt werden
oder
wird ein Ergebnis vom Quality-Gate-Prozess gefordert, so muss es in der geforderten Qualität erstellt werden.
- **Gate-Netzwerk:** Das gewählte Gate-Netzwerk muss zum Entwicklungsprozess passen (vgl. Abschnitt 6.1).
- **Projektvertreter:** Es können keine Projektvertreter am Gate-Review teilnehmen, die nicht im Entwicklungsprozess als Rollen existieren.

11 Verbesserung von Quality-Gate-Referenzprozessen

Viele Unternehmen setzen bereits einen Quality-Gate-Referenzprozess ein [Griffin '97]. Diese Unternehmen sind nur in geringem Maße daran interessiert, ihren Quality-Gate-Referenzprozess von Grund auf neu auszugestalten. Vielmehr suchen sie nach (mitunter einfachen) Möglichkeiten, ihren Quality-Gate-Referenzprozess noch effektiver und effizienter zu gestalten. Verschiedene Möglichkeiten der Verbesserung bieten sich an:

- Der Aufbau eines Erfahrungskreislaufes, um langfristig den etablierten Quality-Gate-Referenzprozess noch weiter zu optimieren. Dazu gehört vor allem die Identifikation geeigneter und ungeeigneter Kriterien.
- Der Einsatz von Softwarewerkzeugen zur Unterstützung der verschiedenen Aktivitäten mit dem Ziel einer Aufwandsreduktion.

11.1 Verbesserung durch Erfahrungskreisläufe

Zentrale Voraussetzung für Verbesserungsmaßnahmen ist die Ermittlung von Erfahrungen. Eine Erfahrung kann aus einem wahrgenommenen Erlebnis gewonnen werden. Sie besteht im Idealfall nach Schneider [Schneider '07b] aus drei Teilen:

- Einer Beobachtung, die das wahrgenommene Erlebnis und seinen Kontext objektiv beschreibt.
- Einem Gefühlswert, durch den die Beobachtung subjektiv bewertet wird.
- Einer Schlussfolgerung, die Hinweise darauf gibt, wie in einem ähnlichen Kontext zukünftig verfahren werden sollte. Durch die Angabe einer Schlussfolgerung wird die Erfahrung erst wiederverwendbar.

Erfahrungen können über Erfahrungskreisläufe identifiziert werden und so in Verbesserungsmaßnahmen einfließen. Erfahrungskreisläufe folgen einem Schema bestehend aus den folgenden Aktivitäten [Liro '04]:

- **Aktivieren:** Hierbei geht es darum, die Erfahrungen von Personen zu aktivieren. Übliche Vorgehensweisen sind Umfragen und Interviews. Am leichtesten fällt die Aktivierung kurz nachdem eine Person in einem bestimmten Kontext (z.B. einer soeben erledigten Aufgabe) eine Erfahrung gemacht hat, da die Erfahrung dann noch am stärksten präsent ist. Einige Techniken machen sich dies zu nutze (vgl. LIDs nach [Schneider '00]). Ein wichtiger weiterer Aspekt ist die Möglichkeit zu einer unaufwändigen Kommunikation von Erfahrungen. Erscheint die Kommunikation einer Erfahrung für einen Erfahrungsträger zu aufwändig, so wird er seine Erfahrungen wahrscheinlich nicht externalisieren.
- **Sammeln und Aufbereiten:** Ziel ist es hierbei, die Erfahrungen dauerhaft in einer standardisierten Form zu speichern, so dass eine spätere Nutzung leichter fällt. Eventuell ist es dabei möglich, Erfahrungen zu verallgemeinern – also auf einen größeren Kontext auszudehnen.
- **Speichern:** Durch das Speichern sollen Erfahrungen dauerhaft (zentral) zugänglich gemacht werden, so dass sie zukünftig bei verschiedenen Gegebenheiten genutzt werden können. Eine so genannte Experience Base bietet hierbei die Möglichkeit zur dauerhaften Speicherung [Basili et al. '94a].
- **Wiederverwenden:** Hierbei geht es um die Nutzung der Erfahrungen in einem konkreten Kontext. Dieser Vorgang ist aktiv durch einen Experience-Engineer zu unterstützen, da sonst die Erfahrungen unter Umständen ungenutzt bleiben. Er hat die Aufgabe, die nötigen Anreize zu schaffen, die für die Wiederverwendung der Erfahrungen sorgen. Durch die Wiederverwendung können Erfahrungen bestätigt werden oder nicht. Daher ist eine erneute Aktivierung der Erfahrungen notwendig.

Innerhalb eines Erfahrungskreislaufes lassen sich zwei Rollen identifizieren:

- **Erfahrungsträger:** Dabei handelt es sich um diejenige Rolle, die die Erfahrungen gemacht hat. Sie verwendet die gespeicherten Erfahrungen in einem bestimmten Kontext wieder und sammelt dabei Erfahrungen, die wiederum aktiviert werden können. Üblicherweise sind Erfahrungsträger Rollen, die aktiv an Projekten mitwirken. Insbesondere sind dies also Entwickler und Manager.
- **Experience-Engineer:** Die Aufgabe eines Experience-Engineers besteht darin, die Aktivierung der Erfahrungen aktiv durchzuführen und zu konzipieren. Weiterhin ist er für das Sammeln und Aufbereiten, das Speichern der Erfahrungen und die Wiederverwendung der Erfahrungen verantwortlich. Der Experience-Engineer besitzt idealerweise Wissen über die Domäne und ist häufig Teil des Qualitätsmanagements. Er hat Erfahrungen damit, Fragebögen zu erstellen und Interviews durchzuführen.

Ein sinnvoller Startpunkt ist die Aktivität „Aktivieren“, da hierdurch eine Ausgangsmenge von Erfahrungen gewonnen werden kann.

Aus den obigen vier Aktivitäten kann ein Erfahrungskreislauf folgender Gestalt abgeleitet werden:

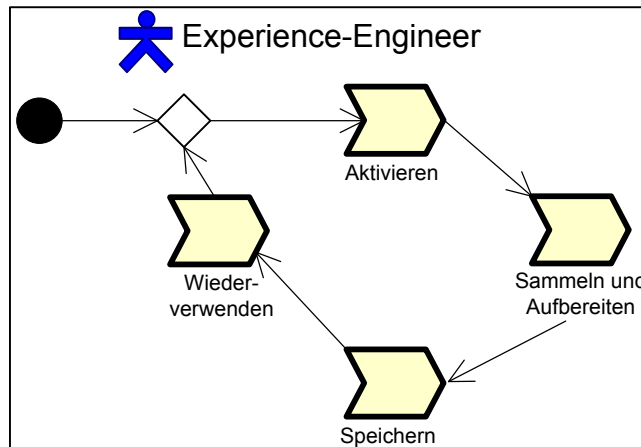


Abbildung 99: Typische Aktivitäten eines Erfahrungskreislaufs

11.2 Überlegungen zur Verbesserung der Effektivität und Effizienz von Quality Gates

Eine Verbesserung der Effektivität kann vorrangig durch geeignete Kriterien, eine genügend große Anzahl von Quality Gates, das Treffen einer geeigneten Entscheidung und durch eine optimale personelle Besetzung der Rollen erreicht werden. Kriterien können durch verschiedene Vorgehensweisen zu verschiedenen Zeitpunkten individuell oder auf Basis eines Kriterienkatalogs mehr oder weniger systematisch erstellt werden (vgl. Abschnitt 7.1). Insbesondere müssen ein vorhandener Kriterienkatalog und eventuelle Auswahlregeln für Kriterien über den Rückfluss von Erfahrungswissen gepflegt werden, da ein Kriterienkatalog mit effektiven Kriterien nicht „schlagartig“ optimal ausgestaltet werden kann.

Kriterien, sowie die Ausgestaltung der Rollen, der Entscheidungsunterstützung und des Gate-Reviews sind insbesondere dann effektiv,

- wenn möglichst wenige Projekte mit erheblichen Qualitätsdefiziten (bedingt) fortgesetzt werden (Fehler 1. Art möglichst gering) und
- wenn möglichst wenige Projekte mit geringfügigen oder keinen Qualitätsdefiziten abgebrochen oder zurückgestellt werden müssen oder ein Quality Gates in diesen Projekten wiederholt werden muss (Fehler 2. Art möglichst gering).

Tabelle 43 fasst alle möglichen richtig und falsch getroffenen Entscheidungen zusammen.

		Qualitätsdefizite im Projekt	
		keine oder geringfügige Qualitätsdefizite	erhebliche Qualitätsdefizite
Entscheidung	Entscheidung im Quality Gate positiv	richtig positiv	falsch positiv (Fehler 1. Art)
	Entscheidung im Quality Gate negativ	falsch negativ (Fehler 2. Art)	richtig negativ

Tabelle 43: Fehler 1. und 2. Art bei der Entscheidung

Im Gegensatz zu einem technischen Review, ist ein Quality Gate auch dann effektiv, wenn nur wenige Mängel im Gate-Review gefunden werden. Dies ist auf die qualitätssteuernde Wirkung der Kriterien zurückzuführen. Insbesondere wenn Kriterien schon im Vorfeld eines Quality Gates bekannt sind (dies ist der optimale und übliche Fall), werden die Entwickler bestrebt sein, diese Kriterien auch zu erfüllen.

Die Effizienz eines Quality-Gate-Referenzprozesses kann auf verschiedene Weise gesteigert werden. Ein Hauptpunkt ist die Ausgestaltung und Unterstützung effizienter Gate-Reviews. Dazu ist jeweils zu überlegen, ob eine Sitzung notwendig ist, da sie aufgrund der terminlichen und räumlichen Koordination verzögernd auf das Projekt wirken kann und vergleichsweise hohe Kosten verursacht.

Ein weiterer Punkt, der die Effizienz eines Quality-Gate-Prozesses beeinflusst, ist die Anzahl der beteiligten Personen. Eine simple Regel lautet hierbei, jede Rolle sollte mit so wenigen Personen wie möglich und mit so vielen Personen wie nötig besetzt werden. Die optimale Anzahl kann nur über Erfahrungswissen ermittelt werden. Eine ähnliche Frage stellt sich für die Besetzung mehrerer Rollen durch eine Person ebenfalls mit dem Ziel der Personaleinsparung.

Eine weitere Möglichkeit zur Verbesserung der Effizienz ist die zentrale Archivierung aller Dokumente, so dass ein Zugriff schnell und reibungslos erfolgen kann. Hierbei sind insbesondere Checklisten, eingereichte Produkte und Protokolle für jedes Quality Gate zu archivieren.

Letztlich kann die im Gate-Review stattfindende Messung durch den Einsatz von Werkzeugen effizienter gestaltet werden. Hierzu können Werkzeuge (in aller Regel Softwarewerkzeuge) zur automatischen Bestimmung der Messwerte verwendet werden, um eine aufwändige manuelle Messung zu vermeiden.

11.3 Analyse der Informationsflüsse

FLOW [Schneider et al. '05] ist ein Forschungsprojekt des Fachgebiets Software Engineering der Gottfried Wilhelm Leibniz Universität Hannover. FLOW bietet eine Notation an, mit der Informations- und Erfahrungsflüsse zwischen Dokumenten und Rollen modelliert werden können. Auf diese Weise können Informationsflüsse modelliert werden, die durch traditionell dokumentenzentrierte Prozessmodellierungssprachen (zum Beispiel SPEM) nicht modelliert werden können. Klassische Muster sind die direkte verbale Kommunikation zwischen zwei Rollen und die Kommunikation zwischen zwei Rollen über ein Dokument. Durch die Modellierung lassen sich Probleme im Informationsfluss identifizieren und möglicherweise beheben.

In Projekten findet in der Regel ein Großteil der Kommunikation verbal zwischen Rollen ohne die Zwischenspeicherung in Dokumenten statt. Diese Art der Kommunikation ist schnell und zielgerichtet, allerdings ist die Information nur bedingt wieder abrufbar. Informationen können verloren gehen oder werden bei mehrmaligem Abruf verschieden wiedergegeben. Die Verfestigung in Dokumenten ist hingegen zeitaufwändig, allerdings gehen keine Informationen verloren. Der Abruf ist stets mit gleicher Qualität möglich – Unterschiede ergeben sich lediglich durch Unterschiede in der Interpretation.

FLOW bietet verschiedene Notationselemente an, die in Tabelle 44 beschrieben werden. In FLOW existieren zwei Typen von Informationsspeichern: Rollen bzw. Personen und Dokumente. Rollen, die

in der Regel durch eine Gruppe von Personen besetzt werden, werden durch hintereinander „gestapelte“ Rollensymbole gekennzeichnet. Informationen aus Dokumenten sind stabil, und wiederholt (jedoch nur langsam) abrufbar. Informationsflüsse aus Dokumenten werden symbolisiert durch durchgezogene Pfeile gekennzeichnet. Das Informationswissen von Rollen bzw. Personen ist flüchtig, jedoch schnell und zielgerichtet abrufbar. Informationswissen aus Rollen wird durch gestrichelte Pfeile gekennzeichnet. Erfahrungsflüsse sind spezielle Informationsflüsse, sie werden durch graue Pfeile dargestellt. In der FLOW-Notation können Aktivitäten ähnlich wie in der IDEF0-Prozessnotation [IDEF0 '93] dargestellt werden. Aktivitäten benötigen Eingaben und liefern Ausgaben. Informationen von Rollen bzw. Personen oder aus Dokumenten können steuernd oder unterstützend wirken.


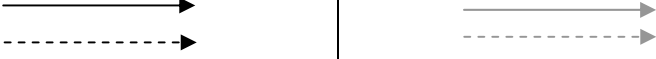
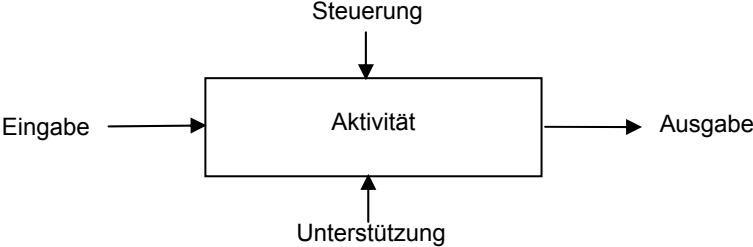
Notationselement	Symbol
Rolle bzw. Personen, Gruppe und Dokument	
Informations- und Erfahrungsflüsse	
Aktivität	

Tabelle 44: Notationselemente in FLOW

11.4 Probleme durch nichtverfestigte Erfahrungen und fehlendes Gate-Management

Ein Gate-Review involviert nahezu alle Rollen, die innerhalb von Quality-Gate-Referenzprozessen definiert sind. Folglich fließen innerhalb des Gate-Reviews viele projektspezifische Informationen, die vor allem der Bewertung und Steuerung des Projektes dienen. Betrachtungen von Gate-Reviews in studentischen Softwareprojekten (vgl. Abschnitt 13.1) zeigen, dass ebenfalls Erfahrungen ausgetauscht werden. Diese Erfahrungen beschäftigen sich dabei inhaltlich nicht ausschließlich mit nicht erfüllten Kriterien, sondern auch mit dem Quality-Gate-Prozess selbst. Typische Erfahrungen beschäftigen sich inhaltlich mit ungeeigneten Kriterien oder ungeeigneten Gatekeepern. Diese Erfahrungen werden mehr oder weniger bewusst an den Gate-Verantwortlichen weitergereicht (vgl. Abbildung 100).

Wie bereits festgestellt wurde, ist die Kommunikation über nicht verfestigende Erfahrungsflüsse schnell, jedoch verblasst das kommunizierte Wissen schnell, wenn es nicht verfestigt wird. Dieser Effekt trifft ebenfalls bei den nicht verfestigten Erfahrungsflüssen im Gate-Review auf. Innerhalb der Sitzung des Gate-Reviews wurden einige Kriterien als ungeeignet identifiziert. Diese Erfahrungen wurden dem Gate-Verantwortlichen insbesondere von den Gutachtern mitgeteilt. Bei später durchgeführten Projekten wurden dieselben ungeeigneten Kriterien jedoch wieder verwendet und häufig abermals als ungeeignet identifiziert. Zwei Gründe können hierfür verantwortlich gemacht werden:

- Es existierte kein Gate-Management, das für die Pflege des Quality-Gate-Referenzprozesses verantwortlich war. Änderungen wurden daher nicht vorgenommen.
- Erfahrungen wurden nicht verfestigt, so dass selbst beim Vorhandensein eines Gate-Managements, die erforderliche Pflege aufgrund verloren gegangener Erfahrungen (Gate-Verantwortlicher als flüchtiger Erfahrungsspeicher) nur eingeschränkt hätte stattfinden können.

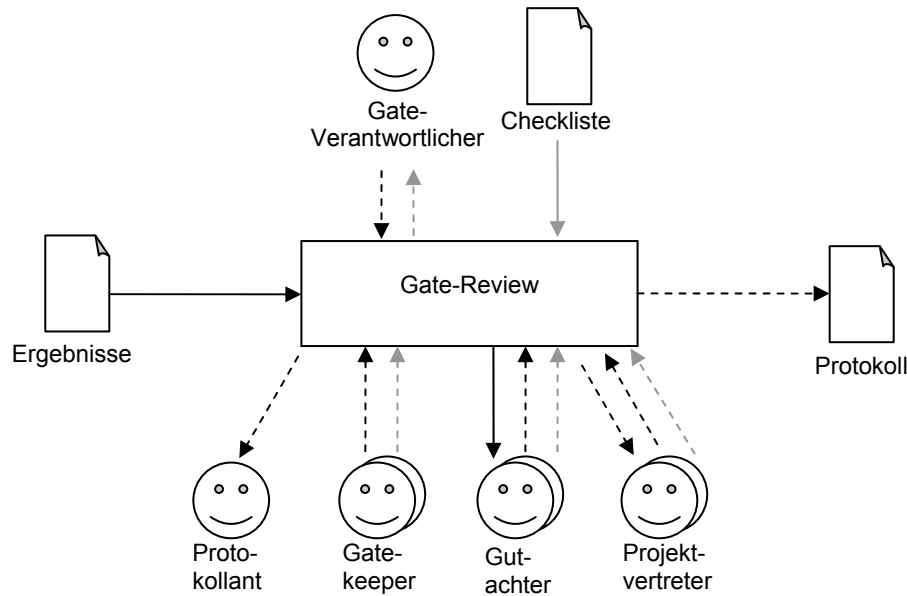


Abbildung 100: FLOW-Sicht auf ein Gate-Review

Eine Möglichkeit diese Problematik aufzulösen, ist die Sammlung der Erfahrungen in einem festen Erfahrungsspeicher und die Etablierung eines Gate-Managements. Das Gate-Management kann dann zu einem späteren Zeitpunkt Aktivitäten zur Pflege anstoßen. Die gesammelten Erfahrungen dienen dabei als Ansatzpunkte für Pflegemaßnahmen. Erfahrungen können von den beteiligten Personen direkt in den temporären Erfahrungsspeicher eingetragen werden oder über den Gate-Verantwortlichen in den Erfahrungsspeicher gelangen (vgl. Abbildung 101). Letzteres hat den Vorteil, dass die Erfahrungen in der Regel schon im gewissen Maße aufbereitet sind und daher durch das Gate-Management leichter genutzt werden können.

Erfahrungsträger sind dabei insbesondere der Gate-Verantwortliche, die Gatekeeper, die Gutachter und die Projektvertreter. Das Eintragen der Erfahrungen in den temporären Erfahrungsspeicher ist ein problematischer Schritt, da die Beteiligten Personen nur Erfahrungen beisteuern werden, wenn das Eintragen ohne großen Aufwand möglich ist und ein Nutzen für die Beteiligten erkennbar ist.

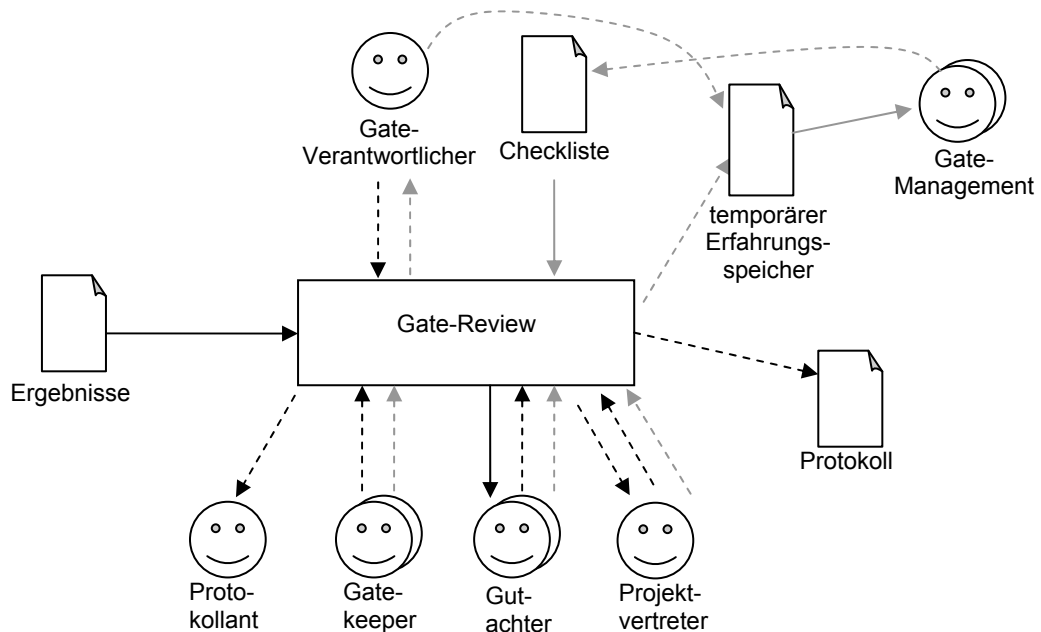


Abbildung 101: Gate-Review mit verfestigten Erfahrungsflüssen

11.5 Softwarewerkzeuge zur Unterstützung von Quality-Gate-Prozessen

Ein Quality-Gate-Prozess kann in vielfältiger Weise durch Software unterstützt werden. Vorrangige Ziele der Unterstützung durch Softwarewerkzeuge sind dabei:

- Eine Unterstützung der Anpassung eines Quality-Gate-Referenzprozesses (vgl. Abschnitt 10.3.4),
- eine Unterstützung der Einplanung von Quality Gates als Teil Instanziierung eines Quality-Gate-Prozesses,
- eine Unterstützung mit dem Ziel der übersichtlichen Darstellung verschiedener Projekte hinsichtlich ihrer erreichten Quality Gates (Quality Gates als Fortschrittsmaß),
- eine Unterstützung der Erstellung und Operationalisierung von Kriterien,
- eine Unterstützung der Entscheidungsfindung,
- eine Unterstützung der Durchführung des Gate-Reviews
- und eine Unterstützung des Erfahrungskreislaufes.

Die empirische Erhebung (vgl. Abschnitt 4.3.3.9) zeigt, dass spezielle Softwarewerkzeuge nur äußerst selten zum Einsatz kommen. Dies wird ebenfalls durch Cooper bestätigt [Cooper '01]. In vielen Unternehmen besteht daher diesbezüglich Potential zur Intensivierung des Einsatzes von Softwarewerkzeugen.

Für einige Aspekte können erhältliche Softwarewerkzeuge isoliert eingesetzt werden, jedoch fehlt ein integriertes und spezielles Softwarewerkzeug, das alle Aspekte unterstützt. Infolgedessen sind Informationen über verschiedene Softwarewerkzeuge verteilt und daher häufig redundant oder in verschiedenen Versionen vorhanden. So kann die Einplanung durch Projektmanagement-Softwarewerkzeuge unterstützt werden. Das Gate-Review kann durch Review-Werkzeuge unterstützt werden. Für technische Review existiert bereits eine Vielzahl an Softwarewerkzeugen [Stanic '06], wobei fast alle niemals in einem größeren Maßstab in Unternehmen eingesetzt wurden.

Die folgende Tabelle zeigt die Softwarewerkzeuge, die als Teilleistung dieser Arbeit entstanden sind (Zeilen) und die jeweiligen durch sie unterstützten Aspekte (Spalten). Alle Softwarewerkzeuge befinden

den sich noch in einem prototypischen Stadium. Insbesondere existieren zum Teil noch Programmfehler oder Schwächen in der Bedienung. Auch werden Informationen im Moment noch teilweise redundant gehalten, obwohl eine Integration der einzelnen Softwarewerkzeuge möglich wäre.

Unterstützter Aspekt	QGT [Peucker '07]	NetQGate [Greive '05, Scholz '06]	RiskQG [Wolski et al. '08]
Anpassung eines Quality-Gate-Referenzprozesses	X	X [▼]	–
Einplanung von Quality Gates	X [♦]	X	X [♦]
Darstellung des Projektfortschritts	–	X	–
Erstellung und Operationalisierung von Kriterien	X [▲]	X	X [▲]
Entscheidungsfindung	–	–	X
Durchführung eines Gate-Reviews	–	X	–
Erfahrungskreislauf	–	X	–
Legende			
–	nicht unterstützt	X	Aspekt wird unterstützt
▼	Unterstützung nur für Kriterien	♦	Nur Auswahl der Quality Gates
▲	Nur Auswahl von Kriterien		

Tabelle 45: Übersicht der entwickelten Softwarewerkzeuge

11.5.1 Unterstützung der Einplanung von Quality Gates

Die zeitliche und personelle Einplanung von Quality Gates findet in der Planungsphase des Projektes statt. Sie verwendet die Ergebnisse des Tailorings – also den Quality-Gate-Prozess, der für die Projektsituation bestimmt wurde. In der Planungsphase wird geklärt, an welchen Terminen die Quality Gates liegen und damit auch, wann die Gate-Reviews stattfinden müssen. Personell sind insbesondere die Rollen Gate-Verantwortlicher, Kriterienersteller, Gutachter, Gatekeeper, Protokollant und Projektvertreter zu besetzen.

NetQGate unterstützt die zeitliche und personelle Einplanung von Quality Gates. Die zeitliche und personelle Einplanung kann bereits durch ein gängiges Projektmanagement-Softwarewerkzeug erfolgen. Allerdings ist ein derartiges Werkzeug zu generisch und umfangreich, um damit lediglich Quality Gates einzuplanen. Eine speziellere und integrierte Lösung, die die Terminologie von Quality-Gate-Referenzprozessen verwendet und gleichzeitig weitere Aspekte (vgl. obige Liste) unterstützt, erscheint daher sinnvoller.

Als Workflow-System leistet NetQGate ausreichend Hilfestellung, so dass auf die Rollen Gate-Verantwortlicher und Protokollant zumindest weitestgehend verzichtet werden kann. Dies wird dadurch realisiert, dass

- die Aktivitäten (vgl. Abbildung 108) eines bestimmten Gate-Reviews durch eine strukturierte Abfolge von Bildschirmmasken unterstützt und dadurch das Gate-Review angeleitet wird und
- das Protokoll als Nebenprodukt ohne größeren Aufwand entsteht.

11.5.2 Darstellung des Projektfortschritts

NetQGate verwaltet die instanziierten Quality-Gate-Prozesse beliebig vieler Projekte. Weiterhin ist dabei jeweils der Status hinsichtlich der erfüllten Quality Gates für jedes Projekt einsehbar. Auf diese Weise wird eine kompakte verdichtete Darstellung des Projektfortschritts für das Projektmanagement

und vor allem für das Qualitätsmanagement bereitgestellt. Dadurch ist sofort erkennbar, welche Projekte dem Zeitplan „qualitativ hinterherlaufen“. Diese Sicht ist vor allem für die die grundlegende Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* relevant.

Abbildung 102 zeigt beispielhaft die Übersicht des Projektfortschritts für ein ausgewähltes Projekt. Im rechten Teil ist ein Projekt mit drei Quality Gates zu erkennen (Zeilen in der Tabelle). Die Spalte *Name* enthält die jeweilige Bezeichnung des Quality Gates. Jedes Quality Gate ist umrahmt, wobei ein grüner Rahmen für ein bestandenes und ein orangefarbener Rahmen (nicht zu sehen in Abbildung 102) für das nächste noch zu bestehende Quality Gate steht. Ein schwarzer Rahmen (nicht zu sehen in Abbildung 102) steht für ein zukünftiges (jedoch nicht für das nächste) Quality Gate. Ein roter Rahmen steht für ein abgebrochenes Projekt. Für jedes Quality Gate sind verschiedene Informationen verfügbar. Insbesondere kann das Protokoll abgerufen werden, damit ein Qualitätsmanager den Grund für die Fortschrittsentscheidung einsehen kann. Im linken unteren Teil sind alle Projekte aufgelistet. Sie können ausgewählt werden, um ihren qualitativen Fortschritt ebenfalls einzusehen.

Benutzer: Thomas Flohr
 Rollen: Quality Management, Gatekeeper
 Passwort ändern
 Logout

Pfad: Home > Projekt: NEH-Sortieranlage > Quality Gates
Quality Gates - NEH-Sortieranlage Hilfe

Neues Quality Gate anlegen

Name	Gatekeeper	Dokumente	Checklisten	
Fertig für Entwurf (bestanden) Deadline: 09.11.2006 16:00 Checkliste anzeigen	• Thao Nguyen	• Anforderungsspezifikation • Vorläufiges Planungsdokument • Rechtserklärung, die die Nutzungsrechte für Quelltexte und Dokumente klärt (eingescannt) Bearbeiten	• Allgemeine formale Anforderungen an alle Dokumente • Anforderungen an die Anforderungsspezifikation • Anforderungen an das vorläufige Planungsdokument Bearbeiten	Bearbeiten Löschen
Fertig für Implementierung (bestanden) Deadline: 07.12.2006 16:00 Checkliste anzeigen	• Daniel Lübke	• Entwurfsdokument • Schnittstellenbeschreibung zwischen großen Einheiten • Zeitplan für Implementierung und Tests Bearbeiten	• Allgemeine formale Anforderungen an alle Dokumente • Anforderungen an das Entwurfsdokument und die Schnittstellenbeschreibung • Anforderungen an den Zeitplan Bearbeiten	Bearbeiten Löschen
Fertig für Abnahme (nicht bestanden) Deadline: 18.01.2007 16:00 Checkliste anzeigen	• Thomas Flohr	• Quellcode der Anwendung • Installierbare Anwendung • Technische Beschreibung • API (Schnittstellenbeschreibung) • Bedienungsanleitung • Testplan Bearbeiten	• Allgemeine formale Anforderungen an alle Dokumente • Anforderungen an die Quelltexte • Anforderungen an die kompilierte Anwendung • Anforderungen an die API-Schnittstellenbeschreibung • Anforderungen an die Bedienungsanleitung • Anforderungen an die Technische Beschreibung • Ausreichende Tests und Erprobungen Bearbeiten	Bearbeiten Löschen

NetQGate
 Personen
 Teams
 Projekt-Merkmale
 Checklisten-Vorlagen
 Projekte
 Bewertungen

Projekte
 PWS TP A
 PWS TP B
 PWS-Gesamt
 EVE-XML
 GOMedit
 KeysignOrg1
 KeysignOrg2
 RiskOG
 SWQuant
 UMLPaint1
 UMLPaint2
 SOA-Me Client
 SOA-Me Integrationsserver
 SOA-Me Kompositionseditor
 ProcessPatternMatcher

Abbildung 102: Projektfortschritt in NetQGate für ein ausgewähltes Projekt

In einer weiteren Darstellung (vgl. Abbildung 103) kann für alle Projekte (neben weiteren Informationen) jeweils die Restzeit für die Einreichung der Produkte für das nächste Quality Gate eingesehen werden. Auf diese Weise kann das Management diejenigen Projekte identifizieren, die demnächst ein Quality Gate durchschreiten müssen. Für diese Projekte muss ein Gate-Verantwortlicher dann die Organisationsaktivitäten des Gate-Reviews einplanen.

Aktuelle Quality Gates
PWS-Gesamt <hr/> <u>Fertig für Abnahme des Releases</u> Restzeit: 5d 17h 41m 35s Checkliste: nicht gelesen Abgegebene Dateien: 0/7 Versuch Nr: 1
PWS TP B <hr/> <u>Fertig für Iteration 3</u> Restzeit: abgelaufen Checkliste: gelesen Abgegebene Dateien: 7/7 Versuch Nr: 2
PWS TP A <hr/> <u>Fertig für Iteration 3</u> Restzeit: abgelaufen Checkliste: gelesen Abgegebene Dateien: 1/7 Versuch Nr: 2

Abbildung 103: Übersicht über die nächsten Quality Gates

11.5.3 Unterstützung der Kriterienerstellung und Operationalisierung

NetQGate verwaltet einen Kriterienkatalog (vgl. Abbildung 104) und unterstützt das Vorgehen *Kriterienkatalog mit Auswahl*. Kriterien des Kriterienkatalogs sind nach Quality Gates geordnet (zweite Spalte von links). Dabei ist es möglich, dass ein Kriterium zu mehreren Quality Gates gehören kann. Innerhalb der dritten Spalte von links sind die jeweiligen Kriterien zu sehen. Um die Übersicht zu wahren, können Kriterien in verschiedene Blöcke unterteilt werden (erste Spalte von links). So kann ein Block allgemeine Kriterien enthalten, die für alle Ergebnisse gelten. Weitere Blöcke enthalten dann die Kriterien, die jeweils nur für spezielle Ergebnisse gelten.

Wird ein Quality Gate in einem Projekt instanziiert, so werden automatisch die passenden Kriterien für das jeweilige Quality Gate ausgewählt. Ein simpler Tailoring-Algorithmus kann zusätzlich diejenigen Kriterien ermitteln, die am besten zu einer gegebenen Projektsituation passen. Durch die Kriterienersteller können im Anschluss nicht passende Kriterien abgewählt werden. Weiterhin können an jedes Kriterium Angaben zur Operationalisierung geheftet werden.

Name	Kategorien	Einträge	
Allgemeine formale Anforderungen an alle Dokumente	<ul style="list-style-type: none"> Fertig für Abnahme Fertig für Entwurf Fertig für Implementierung 	<ul style="list-style-type: none"> Die Dokumente sind durchgehend in deutscher oder englischer Sprache verfasst und entsprechen den Regeln dieser Sprache Die Dokumente liegen elektronisch im MS Word oder MS Powerpoint Format vor Das Layout der Dokumente ist konsistent Alle Dokumente besitzen eine Versionsnummer und ein Erstellungsdatum Die Gruppenbezeichnung ist vorhanden Zu unterschreibene Dokumente sind mit den nötigen Unterschriften, Zeit- und Orstangaben versehen und liegen vor (Bitte auf Papier mitbringen und bei diesem Quality Gate vorlegen). Die Vorgaben für Dateinamen wurden eingehalten (Vorgabe: SWP-[SS WS][JJJJ]-[Gruppenname]-[Dokumentenname]-[Versionsnummer].[Dateiextension]) 	Bearbeiten Löschen
Anforderungen an die Anforderungsspezifikation	<ul style="list-style-type: none"> Fertig für Entwurf 	<ul style="list-style-type: none"> Die Anforderungsspezifikation verwendet das verteilte Template Priorisierte Anforderungen des Kunden werden genannt Die Aufgabe ist beschrieben Es sind Angaben zur Anforderungsanalyse (Herkunft, wahrgenommene Kundentermine, etc.) vorhanden Es sind Angaben zu Rahmenbedingungen und zum Umfeld vorhanden. Insbesondere sind Anwender und die vorhandene Software, Hardware, Orgware benannt. Schnittstellen zu anderen Systemen sind erwähnt Use Cases sind vorhanden Es sind Angaben zu den Qualitätsanforderungen vorhanden. Interne Qualitätsstandards sind benannt. Die Qualitätsanforderungen des Kunden sind benannt und priorisiert Maßnahmen zum Erfüllen der Qualitätsanforderungen sind benannt (z.B. Richtlinien für die Dokumentation, zur konsistenten Formatierung des Quelltextes, Verwendung eines Versionskontrollsystems usw.) Risiken und mögliche Probleme sind benannt Ein Glossar ist vorhanden, welches wichtige Fachbegriffe und Abkürzungen erklärt Es wurden Abnahmetestfälle aufgestellt und dokumentiert, die das spätere Produkt erfüllen muss Die Anforderungsspezifikation wurde von der/dem SE-Berater(in) inhaltlich geprüft. Ein Use-Case-Diagramm mit korrekter UML-Syntax ist vorhanden. 	Bearbeiten Löschen

Abbildung 104: Ausschnitt aus dem Kriterienkatalog in NetQGate

Einen Teil der Operationalisierung bildet die Gewichtung der Kriterien. Die risikobasierte Gewichtung (vgl. Abschnitt 7.3) erfordert es, Kriterien mit Risiken in Beziehung zu setzen. Diese Information ist zu speichern. Um ein Kriterium nun gewichten zu können, müssen Risiken hinsichtlich ihrer Eintrittswahrscheinlichkeit und ihres Schadens bei Eintritt durch Kriterienersteller bewertet werden. Diese Aufgaben werden durch das Softwarewerkzeug *RiskQG* unterstützt. Abbildung 105 zeigt einen Ausschnitt der Risikobewertung in RiskQG. Im oberen Teil kann ein Risiko selektiert werden. Die eigentliche Bewertung findet auf einer Farbskala im unteren Teil der Oberfläche statt.

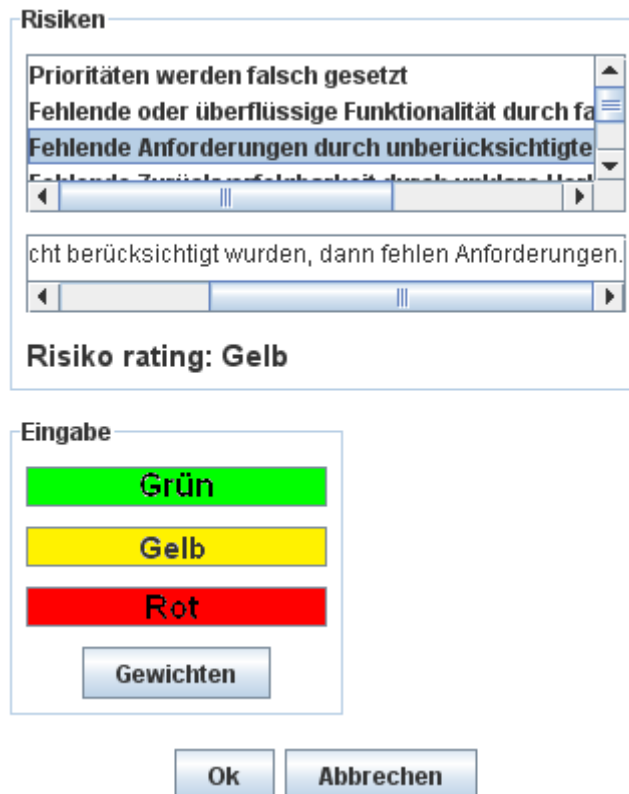


Abbildung 105: Risikobewertung durch RiskQG

Ähnliche Projektsituationen führen zu vergleichbaren Kriterien und folglich auch zu vergleichbaren möglichen Risiken. Deshalb liegt eine Anforderung darin, die Beziehung zwischen Kriterien, möglichen Risiken und ihrer Gewichtung dauerhaft zu speichern, um in ähnlicher Projektsituation gleich verfahren zu können. Eine weitere Anforderung stellt die Unterstützung der Entscheidung dar. Hierbei sind gemäß Abschnitt 9.3 der Erfüllungsgrad jedes Kriteriums und seine Gewichtung zu berücksichtigen. Abbildung 106 zeigt einen Ausschnitt aus RiskQG. Im oberen Teil kann der Erfüllungsgrad jedes Kriteriums bewertet werden. Im unteren Teil ist das Ergebnis der Entscheidungsunterstützung zu sehen. Ein Ampelindikator liefert einen Hinweis über die Fortsetzung des jeweiligen Projektes. Im rechten Teil werden die möglichen Risiken aufgezählt, die aus der Nichterfüllung resultieren.

The screenshot displays a software interface for risk assessment. At the top, there is a checklist with five items, each with a checkbox:

- Die Anforderungen des Kunden müssen priorisiert sein.
- Die Anforderungsspezifikation muss eine Versionsnummer und ein
- Angaben zum Ablauf der Anforderungsanalyse (Herkunft, wahrgen
- Ein Glossar muss vorhanden sein, das wichtige Fachbegriffe und Al
- Das Layout der Anforderungsspezifikation muss konsistent sein.

Below the checklist, there is a vertical traffic light indicator with three circles: a red circle at the top labeled "Rot", a yellow circle in the middle labeled "Gelb", and a green circle at the bottom labeled "Grün". To the right of the traffic light, a yellow box contains the text "Nacharbeit notwendig!". Below this, a text area titled "auftretende Risiken" contains the text "Fehlende oder überflüssige Funkti" and "Anforderungsspezifikation schwe". Below the text area is a horizontal scrollbar. Further down, there is a text area titled "Risikenbeschreibung" which is currently empty. At the bottom right, there is a button labeled "Beenden".

Abbildung 106: Bewertung der Kriterien und Ergebnis der Entscheidungsunterstützung

11.5.4 Unterstützung der Durchführung des Gate-Reviews

Im Gegensatz zur Software-Unterstützung für technische Reviews [Stanic '06], ist die Software-Unterstützung für Gate-Reviews wissenschaftlich nahezu nicht untersucht. Softwarewerkzeuge für technische Reviews unterstützen die verschiedenen Aktivitäten eines Reviews (vgl. Abschnitt 3.3.4). Klassische Aufgaben derartiger Softwarewerkzeuge sind nach MacDonald et al. [Macdonald et al. '96]:

- **Dokumentenmanagement:** Das Dokumentenmanagement unterstützt die übersichtliche Verwaltung und Versionierung verschiedener Dokumente. Dazu gehören Checklisten, Protokolle und eingereichte Dokumente.
- **Unterstützung der individuellen Vorbereitung:** Hierbei wird die individuelle Vorbereitung durch spezielle Lesetechniken (dazu gehört ebenfalls das checklistenbasierte Lesen) unterstützt. Ein Softwarewerkzeug kann hierbei anleitend wirken. Darüber hinaus kann die Dokumentation von Befunden dadurch unterstützt werden, dass Befunde ortsnah (beispielsweise in Form einer Anmerkung an ein fehlerhaftes Code-Fragment) notiert werden können.

- **Unterstützung der Sitzung:** Ziel ist hierbei eine zeitliche oder räumliche Entkoppelung der Sitzung, um Terminprobleme zu mindern und Reviews über Unternehmensstandorte hinweg zu realisieren. Softwarewerkzeuge können hierbei Video-Konferenzen (gleiche Zeit, verschiedener Ort) ermöglichen oder eine Diskussionsplattform bereitstellen (Zeit und Raum verschieden).
- **Leistungsbewertung:** Die Leistungsbewertung beschäftigt sich mit der Bewertung der Effektivität und Effizienz des technischen Reviews. Typische Maße befassen sich beispielsweise mit der Anzahl der Befunde (Effektivität) und der Anzahl der Befunde pro Zeiteinheit (Effizienz). Ein Softwarewerkzeug unterstützt die automatische Bestimmung der Messwerte oder erleichtert zumindest deren Ermittlung.

In Abschnitt 8.1 wurde hinsichtlich der Ausgestaltung des Gate-Reviews festgestellt, dass ähnliche Review-Varianten für das Gate-Review existieren und ihre Abläufe ähnlich sind. Deshalb erscheint eine Orientierung an den Softwarewerkzeugen zur Unterstützung von technischen Reviews angemessen. Prinzipiell lässt sich daher ein Softwarewerkzeug zur Unterstützung eines technischen Reviews auch zur Unterstützung des Gate-Reviews einsetzen, sofern es die Verwendung von Checklisten mit Kriterien und die Erstellung von Protokollen ermöglicht. Da Termin- und Raumprobleme ebenfalls bei der Sitzung eines Gate-Reviews auftreten können, ist die Unterstützung einer von Zeit oder Raum entkoppelten Sitzung ebenfalls wünschenswert. Häufig erweisen sich Softwarewerkzeuge zur Unterstützung von technischen Reviews in bestimmten Aspekten als ungeeignet zur Unterstützung eines Gate-Reviews, da

- sie häufig nur die Begutachtung bestimmter Ergebnisse (z.B. nur Quellcode) zulassen,
- sie insbesondere die individuelle Vorbereitung der Gutachter und die Zusammenführung zu einer Gesamtbefundliste unterstützen (die jedoch für das Gate-Review weniger stark benötigt werden),
- die Bezeichnungen für Prozessschritte, Artefakte und Ressourcen denen von technischen Reviews entsprechen
- und sie einen bestimmten und häufig starren Workflow vorschreiben.

Einen umfangreichen Überblick über momentan vorhandene Werkzeuge gibt Stanic [Stanic '06].

Vor allem sind das Dokumentenmanagement und die Unterstützung der Sitzung von Interesse. Die individuelle Vorbereitung fällt hingegen im Gate-Review weniger variantenreich aus – sie ist daher von geringerem Interesse. Im Wesentlichen ist hierbei die Checkliste mit operationalisierten Kriterien für alle Gutachter präsent. Eine Leistungsbewertung (im Sinne eines technischen Reviews) ist ebenfalls von geringerer Bedeutung, da weder die relative noch die absolute Anzahl von Befunden ausschlaggebend für die Effizienz und Effektivität des Gate-Reviews sind.

NetQGate unterstützt als Softwarewerkzeug das Dokumentenmanagement (vgl. Abbildung 107). Auf diese Weise kann das Qualitätsmanagement mit der großen Anzahl von Dokumenten umgehen, die im Rahmen der verschiedenen Projekte auftreten. Hierbei können eingereichte Dokumente, Checklisten und Protokolle nach Quality Gate und Projekt archiviert und von berechtigten Personen dauerhaft abgerufen werden. Berechtigte Personen sind dabei die jeweiligen Gutachter, die Gatekeeper, Projektvertreter und das Gate-Management. Mit Ausnahme des Gate-Managements haben die Rollen jedoch nur Zugriff auf die Dokumente von Projekten, an denen sie beteiligt sind.

Das Dokumentenmanagement kann ebenfalls durch ein separates Dokumentenmanagement-Werkzeug unterstützt werden, sofern Dokumente (über Metadaten) nach Quality Gates und Projekten kategorisiert werden und mit entsprechenden Zugriffsrechten geschützt werden können.

Abbildung 107 zeigt die Verwaltung von Dokumenten für ein momentan ausgewähltes Quality Gate. Im oberen Teil sind verschiedene Dokumente erkennbar. Im unteren Teil ist das Protokoll einsehbar, das zum Quality Gate gehört.

Projekt-Merkmale

[Checklisten-Vorlagen](#)

[Projekte](#)

[Bewertungen](#)

Projekte

[EVE-XML](#)

[GOMEdit](#)

[KeysignOrg1](#)

[KeysignOrg2](#)

[RiskQG](#)

[SWQuant](#)

[UMLPaint1](#)

[UMLPaint2](#)

[SOA-Me Client](#)

[SOA-Me Integrationsserver](#)

[SOA-Me Kompositionseditor](#)

[ProcessPatternMatcher](#)

[NEH-Sortieranlage - Team A](#)

[NEH-Sortieranlage - Team B](#)

[NEH-Sortieranlage - Team C](#)

[WüSin](#)

Aktuelle Quality Gates

[KeysignOrg1](#)

Fertig für Entwurf

Restzeit: 13d 17h 0m 22s

Checkliste: nicht gelesen

Abgegebene Dateien: 0/3

Versuch Nr. 1

Vorzulegende/Hochgeladene Dokumente

Name	Dateiname	Hochgeladen von	Upload-Datum	
Entwurfsdokument	swp-ws0607-ppm-entwurfsspezifikation-v1.0.doc		07.12.2006 13:26	Download
Schnittstellenbeschreibung zwischen großen Einheiten	swp-ws0607-ppm-entwurfsspezifikation-v1.0.doc		07.12.2006 13:27	Download
Zeitplan für Implementierung und Tests	swp-ws0607-ppm-entwurfsspezifikation-v1.0.doc		07.12.2006 13:28	Download

Allgemeine formale Anforderungen an alle Dokumente

Die Dokumente sind durchgehend in deutscher oder englischer Sprache verfasst und entsprechen den Regeln dieser Sprache ■

gelesen

Die Dokumente liegen elektronisch im MS Word oder MS Powerpoint Format vor ■

gelesen

Guidelines:

- ☞ Nicht für die Rechtserklärung.
- ☞ Quellcode, ausführbare Anwendungen, Schnittstellenbeschreibungen und ähnliche Dokumente sind ausgenommen.
- ☞ Die Nutzer des SE-Wikis zur Unterstützung des Requirements Engineerings können alternativ eine Anforderungsspezifikation im PDF-Format zum Quality Gate einreichen.

Das Layout der Dokumente ist konsistent ■

gelesen

Alle Dokumente besitzen eine Versionsnummer und ein Erstellungsdatum ■

gelesen

Guidelines:

- ☞ Nicht bei der Rechtserklärung
- ☞ Kriterium gilt sonst für alle Dokumente.

Die Gruppenbezeichnung ist vorhanden ■

Abbildung 107: Dokumentenmanagement in NetQGate

Weiterhin unterstützt NetQGate die Durchführung der Sitzung eines Gate-Reviews. Zwei Einsatzstrategien sind hierbei möglich:

- Die vollständige Einsatzstrategie:** NetQGate ermöglicht hierbei eine von Zeit und Raum entkoppelte Durchführung der Sitzung im Falle der Gate-Review-Variante Inspektion (vgl. Abbildung 108). Hierzu bietet NetQGate eine Diskussionsplattform an. Insbesondere kann dabei über strittige (weil nicht erfüllte) Kriterien diskutiert werden. Die Diskussion findet hierbei zwischen den Gutachtern und den Projektvertretern statt. NetQGate gibt hierbei einen Workflow vor. Auf diese Weise kann auf einen Gate-Verantwortlichen weitestgehend im Gate-Review verzichtet werden. Die einzige Aufgabe des Gate-Verantwortlichen besteht darin, für die zügige Abwicklung des Gate-Reviews zu sorgen, wenn einzelne Aktivitäten zu lange dauern. Weiterhin existiert keine separate Rolle Protokollant, da NetQGate ein Protokoll als Nebenprodukt erzeugt. Dazu ist es allerdings notwendig, dass die Gutachter nicht erfüllte Kriterien und die Gatekeeper die Entscheidung schriftlich in NetQGate begründen. Wichtig bei der vollständigen Einsatzstrategie ist, dass die Projektvertreter, Gutachter und Gatekeeper über die wichtigen Ereignisse im Workflow informiert werden (vorzugsweise über E-Mail), da davon auszugehen ist, dass sie nicht dauerhaft im NetQGate-System angemeldet sind. Auf diese Weise werden sie über den aktuellen Stand eines Gate-Reviews informiert. Im Sinne der Forschung im Bereich kollaborativer Workflow-Systeme werden derartige Benachrichtigungen unter dem Begriff *Awareness* geführt.
- Die hybride Einsatzstrategie:** NetQGate wird dazu in einer „real“ durchgeführten Sitzung (in Anwesenheit aller Rollen) als Softwarewerkzeug eingesetzt, das den Zugriff auf alle benötigten Dokumente zentral erlaubt (und damit erleichtert) und die Protokollierung der Bewertung der Kriterien und der Entscheidung ermöglicht. NetQGate dient hierbei also lediglich als Dokumentenmanagement-System und als Werkzeug zur Erstellung des Protokolls.

Der Ablauf bei vollständiger Einsatzstrategie stellt sich wie folgt dar:

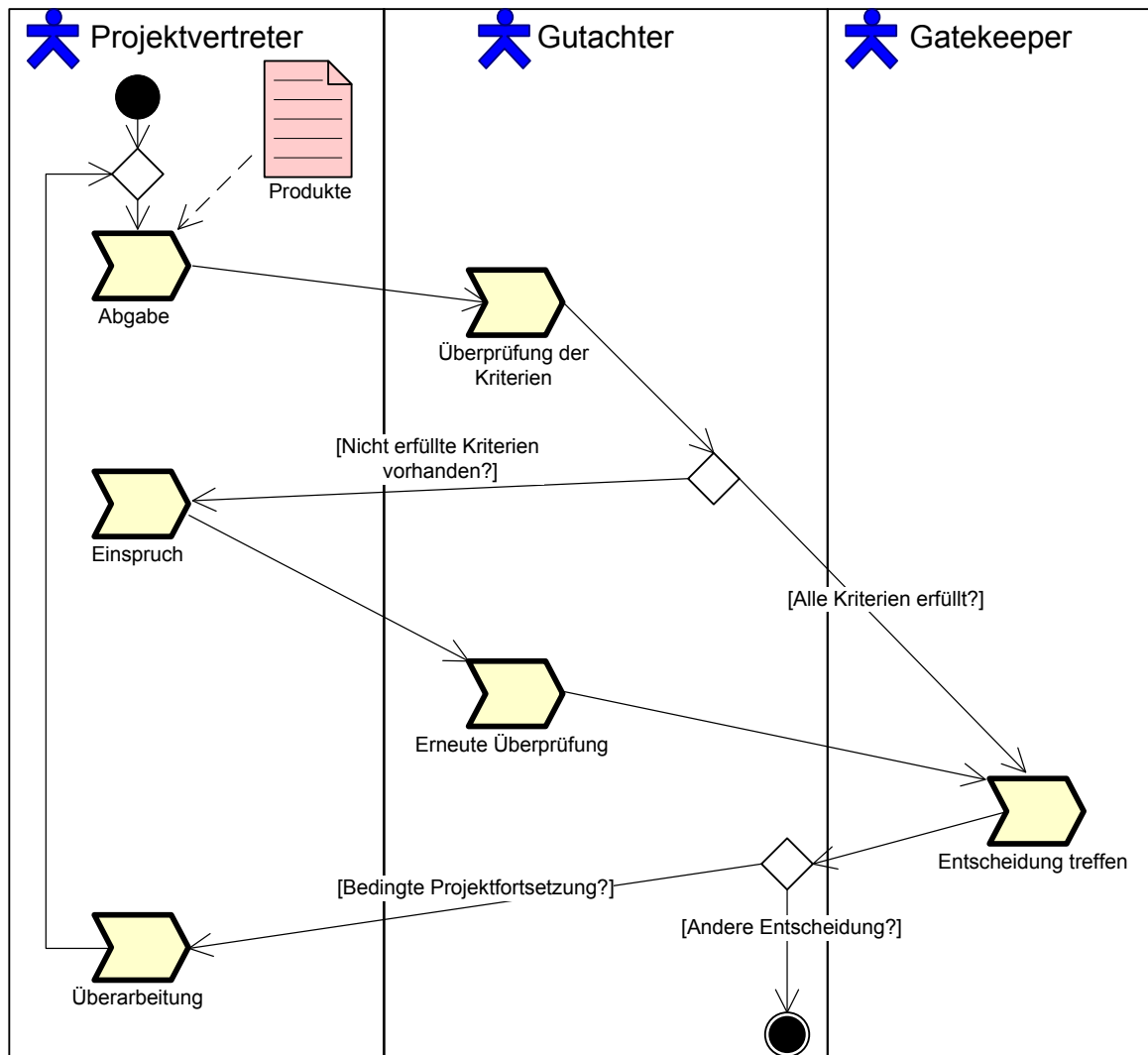


Abbildung 108: Aktivitätsdiagramm vollständige Einsatzstrategie in NetQGate

Die einzelnen Aktivitäten schlüsseln sich dabei wie folgt auf:

- **Abgabe:** Hierbei können die geforderte Dokumente bis zu einem vorgegebenen Termin (Deadline) durch die Projektvertreter in das NetQGate-System hochgeladen werden. Bevor das Hochladen möglich ist, muss jedes Kriterium der Checkliste als gelesen markiert werden. Auf diese Weise wird sichergestellt, dass alle Kriterien den Projektvertretern bekannt sind.
- **Überprüfung der Kriterien:** Die Überprüfung der Kriterien erfolgt durch die Gutachter. Von den Kriterienerstellern hinterlegte Angaben zur Operationalisierung, die an jedes Kriterium angeheftet werden können, helfen bei der Überprüfung der Kriterien. Nichterfüllte Kriterien können mit einer Begründung versehen werden, um den Projektvertretern einen Hinweis auf den Mangel zu geben.
- **Einspruch:** Bei Mängeln wird den Projektvertretern durch die Gutachter eine Möglichkeit zum Einspruch eingeräumt, da eventuell ein Irrtum hinsichtlich der Bewertung der Kriterien vorliegen könnte. Dazu kann eine Frist gesetzt werden, innerhalb der die Projektvertreter zu den nichterfüllten Kriterien Stellung nehmen können. Um die Projektvertreter davon zu rechtzeitig zu informieren, wird eine E-Mail verschickt, die Angaben zur Einspruchsfrist enthält. Die übliche Länge der Einspruchsfrist liegt zwischen zwei bis fünf Werktagen. Von längeren Einspruchsfristen sollte abgesehen werden, da sie das Projekt verzögern könnten.
- **Erneute Überprüfung:** Die Gutachter gehen hierbei gezielt auf die nichterfüllten Kriterien ein, zu denen die Projektvertreter Stellung bezogen haben. Sie urteilen dann erneut, ob das jeweilige Kriterium aufgrund der Stellungnahme als erfüllt bewertet werden kann oder nicht.

- **Entscheidung treffen:** Die Gatekeeper treffen hierbei eine Entscheidung über den Fortsetzung des Projektes. NetQGate bietet hierzu die Entscheidungen uneingeschränkte Projektfortsetzung, bedingte Projektfortsetzung, Wiederholung des Quality Gates und Projektabbruch an. Die Entscheidung wird den Projektvertretern per E-Mail mitgeteilt. Die Entscheidung wird zusammen mit allen Bewertungen von Kriterien und Stellungnahmen als Protokoll gespeichert.
- **Überarbeitung:** Die Überarbeitung setzt die bedingte Projektfortsetzung um. Für die Überarbeitung wird eine Frist gesetzt, nach der eine erneute Bewertung der Kriterien stattfindet.

NetQGate erstellt des Weiteren als Nebenprodukt unabhängig von der Einsatzstrategie ein Protokoll, das alle verpflichtenden Bestandteile (mit Ausnahme der Maßnahmen) enthält (vgl. Abbildung 83).

11.5.5 Unterstützung des Erfahrungskreislaufs

NetQGate unterstützt einen Erfahrungskreislauf, der die Aktivitäten „Aktivieren“, „Sammeln und Aufbereiten“, „Speichern“ und „Wiederverwenden“ gemäß Abschnitt 11.1 enthält. Der Erfahrungskreislauf beschränkt sich dabei nahezu einseitig auf die Verbesserung des Kriterienkataloges als wichtigsten Bestandteil eines Quality-Gate-Referenzprozesses, der der Strategie „Quality Gates als einheitliche Qualitäts-Leitlinie“ folgt. NetQGate enthält daher eine sehr spezielle Experience Base, die die Erfahrungen bezüglich der Kriterien eines Kriterienkataloges verwaltet.

Die Rolle des Experience-Engineers wird vom Gate-Management übernommen, da es zum einen über Domänenwissen und zum anderen ohnehin die Aufgabe der Pflege des Quality-Gate-Referenzprozesses übernimmt. Relevante Erfahrungsträger sind die Projektvertreter, die Gutachter, die Gatekeeper und das Gate-Management. Projektvertreter, Gutachter, Gatekeeper und Gate-Management besitzen in NetQGate die Möglichkeit, mit geringem Aufwand Kriterien mit einer Erfahrung zu versehen (vgl. Abbildung 109). Dies ist jederzeit möglich – insbesondere also auch unmittelbar nachdem eine Erfahrung mit einem Kriterium gemacht wurde. Dies ist also besonders während des Gate-Reviews der Fall, aber auch schon bereits im Vorfeld des Quality Gates, wenn es gilt, die Kriterien zu erfüllen. Auf diese Weise wird der, in Abschnitt 11.4 beschriebenen, Problematik entgegengewirkt. Gleichgültig, ob die hybride oder vollständige Einsatzstrategie gewählt wurde, können dabei Erfahrungen zu Kriterien aktiviert werden.

Erfahrungen in NetQGate besitzen eine Beobachtung und Schlussfolgerung (freier Text), eine emotionale Bewertung (Element einer vierstufigen ordinalen Skala) und einen Kontext (freier Text und Auswahlmöglichkeit). Der Kontext basiert auf einem Projektmodell und stellt jeweils die Projektsituation dar, für die die Erfahrung gilt. Bei der Eingabe einer Erfahrung wird automatisch der Kontext des aktuellen Projektes verwendet. Es ist jedoch möglich, diesen zu ändern.

Die so gesammelten Roh-Erfahrungen werden zunächst automatisch in einem temporären Erfahrungsspeicher gesammelt. Von hier aus können sie vom Gate-Management (dem Experience-Engineer) aufbereitet werden. Zur Unterstützung können die Erfahrungen gefiltert werden. NetQGate bietet dazu verschiedene Filter an, die kombiniert werden können und damit den Überblick über die Roh-Erfahrungen deutlich erleichtern:

- Anzeigen von unaufbereiteten Erfahrungen.
- Anzeigen von Erfahrungen mit einer Begründung.
- Erfahrungen, die innerhalb eines bestimmten Zeitraums aktiviert wurden.
- Erfahrungen, die von einer bestimmten Rolle erstellt wurden (Projektvertreter, Gutachter und Gate-Management).
- Erfahrungen, die in einem bestimmten Projekt gemacht wurden.
- Erfahrungen, die nur für ein bestimmtes Quality Gate gelten.

Die Aufbereitung umfasst die Überprüfung des Kontextes, der emotionalen Bewertung, der Beobachtung und der Schlussfolgerung. Dabei können ähnliche Erfahrungen zu einer Erfahrung zusammengefasst werden oder konträre Erfahrungen im NetQGate-System belassen werden, da das Tailoring-Verfahren in NetQGate dann beide Erfahrungen berücksichtigt. Alle aufbereiteten Erfahrungen wer-

den letztlich abgespeichert und im Regelwerk von NetQGate gespeichert. Jede aufbereitete Erfahrung stellt bereits eine Regel dar, sie bedarf also keiner weiteren Aufbereitung.

Die Anwendung liegt als ausführbares Archiv vor. ■

erfüllt nicht erfüllt Kriterium

gelesen

Guidelines:

- ☞ Liegt ein ausführbares jar-Archiv vor?
- ☞ Bei Anwendungen, die mehrere externe jars benötigen, sollten diese zusammen mit der Anwendung als gepacktes zip-Archiv ausgeliefert werden.
- ☞ Bei Web-Applikationen und Java als war-Archiv.

Bewertung dieses Checklistenpunktes: voll größtenteils weniger nicht geeignet

Begründung der Bewertung:
Für PHP-Projekte kann kein ausführbares Archiv erzeugt werden. Ausblenden

Bewertung soll für diese Quality Gate Kategorien gelten:

Fertig für Implementierung
 Fertig für Abnahme
 Fertig für Iteration 2
 Fertig für Iteration 3 Für alle

Bewertung soll für folgende Projektmerkmale gelten:
(Die gewählten Merkmalswerte dieses Projektes sind fett markiert)

Projektmerkmal	Werte	Für alle
Projekt-Größe	sehr klein klein mittel groß sehr groß	<input type="checkbox"/>
Team-Größe	klein (weniger als 5 Personen) mittel (zwischen 5 und 15 Personen) groß (mehr als 15 Personen)	<input type="checkbox"/>
Technologie	J2EE PHP C++ C# .Net Delphi Java	<input type="checkbox"/>

Emotion

Beobachtung +
Kontext +
Schlussfolgerung

Kontext

Abbildung 109: Aktivierung von Erfahrungen in NetQGate

NetQGate ermöglicht die Wiederverwendung der Erfahrungen bei Instanziierung einer Checkliste für ein Quality Gate (vgl. auch Abschnitt 11.5.3). Dabei ist es notwendig das Quality Gate oder die Projektsituation anzugeben.

11.5.6 Zielerreichung durch den Einsatz von NetQGate

Während des Einsatzes von NetQGate in den studentischen Softwareprojekten im Fachgebiet Software Engineering der Gottfried Wilhelm Leibniz Universität Hannover (vgl. Abschnitt 13.1) konnte in den Jahren 2006 und 2007 die Zielerreichung festgestellt werden. Des Weiteren konnten verschiedene Auswirkungen des Einsatzes beobachtet werden. Zielerreichung und Auswirkungen des Einsatzes werden im Folgenden beschrieben.

Die Zielerreichung stellt sich wie folgt dar:

- **Unterstützung der Einplanung von Quality Gates:** NetQGate unterstützt direkt nur die Einplanung von Quality Gate bei wasserfallartigem Vorgehen. Quality-Gate-Prozesse für andere Vorgehensweisen können jedoch über Umwege instanziiert werden.
- **Darstellung des Fortschritts auf Basis von Quality Gates:** Die Darstellung des Fortschritts ist übersichtlich und zielführend. Informationen zum Fortschritt konnten schnell abgerufen und genutzt werden.
- **Unterstützung der Erstellung und Operationalisierung von Kriterien:** Die übersichtliche Verwaltung eines Kriterienkatalogs ist durch NetQGate möglich. Die Unterteilung der Kriterien nach Quality Gates und in Kategorien hilft dabei insbesondere, die Übersichtlichkeit zu gewährleisten, da ein Kriterienkatalog schon in kleineren Szenarien > 50 Kriterien enthalten kann (vgl. Abschnitt 13.1.2). Die Gewichtung von Kriterien kann durch RiskQG vorgenommen werden, wodurch der Verwaltungsaufwand weiter reduziert werden kann. Weiterhin können bestimmte Konstellationen von Kriterien und gewichteten Risiken gespeichert werden,

um sie in ähnlicher wieder verwenden zu können. Das Ergebnis der Entscheidungsunterstützung kann durch RiskQG berechnet werden, wobei im Moment nicht alle erdenklichen Ausgestaltungen der Entscheidungsunterstützung in RiskQG modelliert werden können.

- **Unterstützung der Durchführung des Gate-Reviews:** Die Unterstützung einer zeitlich und örtlich entkoppelten Gate-Review-Sitzung funktioniert im Moment gut. Jedoch stellt sich dabei heraus, dass der in NetQGate realisierte Workflow zu starr ist, als dass damit sämtliche möglichen erdenklichen Gate-Reviews realisiert werden können. Dies ist ein bekanntes und ungelöstes Problem im Bereich der Softwarewerkzeuge für technische Reviews nach Hedberg [Hedberg '04]. Ein Unternehmen, das ein ähnliches Softwarewerkzeug einsetzen möchte, wird daher mit großer Wahrscheinlichkeit eine eigene Lösung entwickeln müssen. NetQGate als reines Dokumentenmanagement-System verringert hingegen den Aufwand einer manuellen Verwaltung der Dokumente und kann unabhängig von der genauen Ausgestaltung des Gate-Reviews eingesetzt werden. Selbst bei der hybriden Einsatzstrategie sind alle notwendigen Dokumente schnell und zentral verfügbar.
- **Unterstützung des Erfahrungskreislaufes:** Der Erfahrungskreislauf in NetQGate beschränkt sich einseitig auf die Verbesserung des Kriterienkataloges. Dies stellt zunächst keine allzu große Einschränkung dar, da die Kriterien das wichtigste Kontrollinstrument sind. Es ist positiv festzustellen, dass im Rahmen der studentischen Softwareprojekte von der Möglichkeit zur Hinterlegung von Erfahrungen rege Gebrauch gemacht wurde (vgl. Abschnitt 13.1). Weitere Erfahrungen (z.B. zu geeigneten Gatekeepern) sind im Moment außerhalb von NetQGate zu aktivieren (beispielsweise in Workshops). Weiterhin ist es im Moment nur seitens des Gate-Managements möglich, neue Kriterien in den Kriterienkatalog einzutragen. Alle anderen Rollen haben keine Möglichkeit, Kriterien vorzuschlagen.

Die folgenden Auswirkungen konnten beobachtet werden:

- Alle geforderten Produkte wurden stets termingerecht in das NetQGate-System hochgeladen. Dies ist eine Verbesserung gegenüber den Vorjahren, in denen Produkte häufig mit Verspätung eingereicht wurden.
- Ohne vorherige spezielle Einweisung in das integrierte Erfahrungssystem von NetQGate wird es speziell von Projektvertretern häufig missverstanden. Irrtümlich wird häufig angenommen, dass Projektvertreter den Erfüllungsgrad der Kriterien bewerten können und nicht die Eignung des Kriteriums. Erst nach einer speziellen Einweisung seitens des Gate-Managements tritt dieses Problem nicht auf.
- Bei der vollständigen Einsatzstrategie wird das Gate-Review entkoppelt von Zeit und Ort ausgeführt. Hierbei kann das Gate-Review psychologisch anders auf die Projektvertreter wirken als ein zur selben Zeit und am selben Ort durchgeführtes Gate-Review. Infolgedessen könnte es häufiger zu bedingten Projektfortsetzungen kommen, wenngleich dies im Moment noch nicht empirisch belegt werden kann (vgl. Abschnitt 13.1.1).

12 Möglichkeiten zur Verwendung des Frameworks

Die Art der Verwendung des Frameworks durch ein Unternehmen hängt von seiner Ausgangssituation ab (vgl. Abschnitt 1.2):

- Ein Unternehmen besitzt einen Referenzprozess für die Software-Entwicklung, der zumindest das Vorgehen auf Phasenebene beschreibt. Es ist jedoch kein Quality-Gate-Referenzprozess etabliert.
- Ein Unternehmen besitzt bereits einen Quality-Gate-Referenzprozess. Dennoch vermutet das Gate-Management des Unternehmens, dass der Quality-Gate-Referenzprozess nicht optimal oder vollständig ausgestaltet ist und somit Fehlschläge wahrscheinlich sind oder sogar gehäuft auftreten.
- Ein Unternehmen besitzt einen Quality-Gate-Referenzprozess, der weitestgehend ausgestaltet ist – Fehlschläge sind daher eher selten. Jedoch besteht noch Verbesserungspotential hinsichtlich der Effizienz und Effektivität.

Die folgenden Abschnitte beschreiben die möglichen Wege, die ein Unternehmen mit gegebener Ausgangssituation jeweils beschreiten kann.

12.1 Unternehmen ohne Quality-Gate-Referenzprozess

Als ergebnisorientierte Zeitpunkte können Quality Gates (und damit ein Quality-Gate-Referenzprozess) nur etabliert werden, wenn ein Referenzprozess im betroffenen Unternehmen vorliegt. Dieser Referenzprozess muss durch den Vorgang der Anpassung zumindest Entwicklungsprozesse liefern, die die Entwicklung mindestens auf Phasenebene – also vergleichsweise oberflächlich – beschreiben.

Der erste Schritt zur Etablierung eines Quality-Gate-Referenzprozesses liegt in der Festlegung der grundlegenden Strategie (entweder *Quality Gates als einheitliche Qualitäts-Leitlinie* oder *Quality Gates als flexible Qualitätsstrategie*) durch das Prozessmanagement. Bei dieser Festlegung kann das Prozessmanagement durch die Zielsetzungen einer höheren Management-Ebene beeinflusst werden. Wird die Entscheidung über die Festlegung der Strategie nicht bereits auf einer höheren Management-Ebene getroffen, so kann der folgende Entscheidungsbaum zur Bestimmung der Strategie eingesetzt werden.

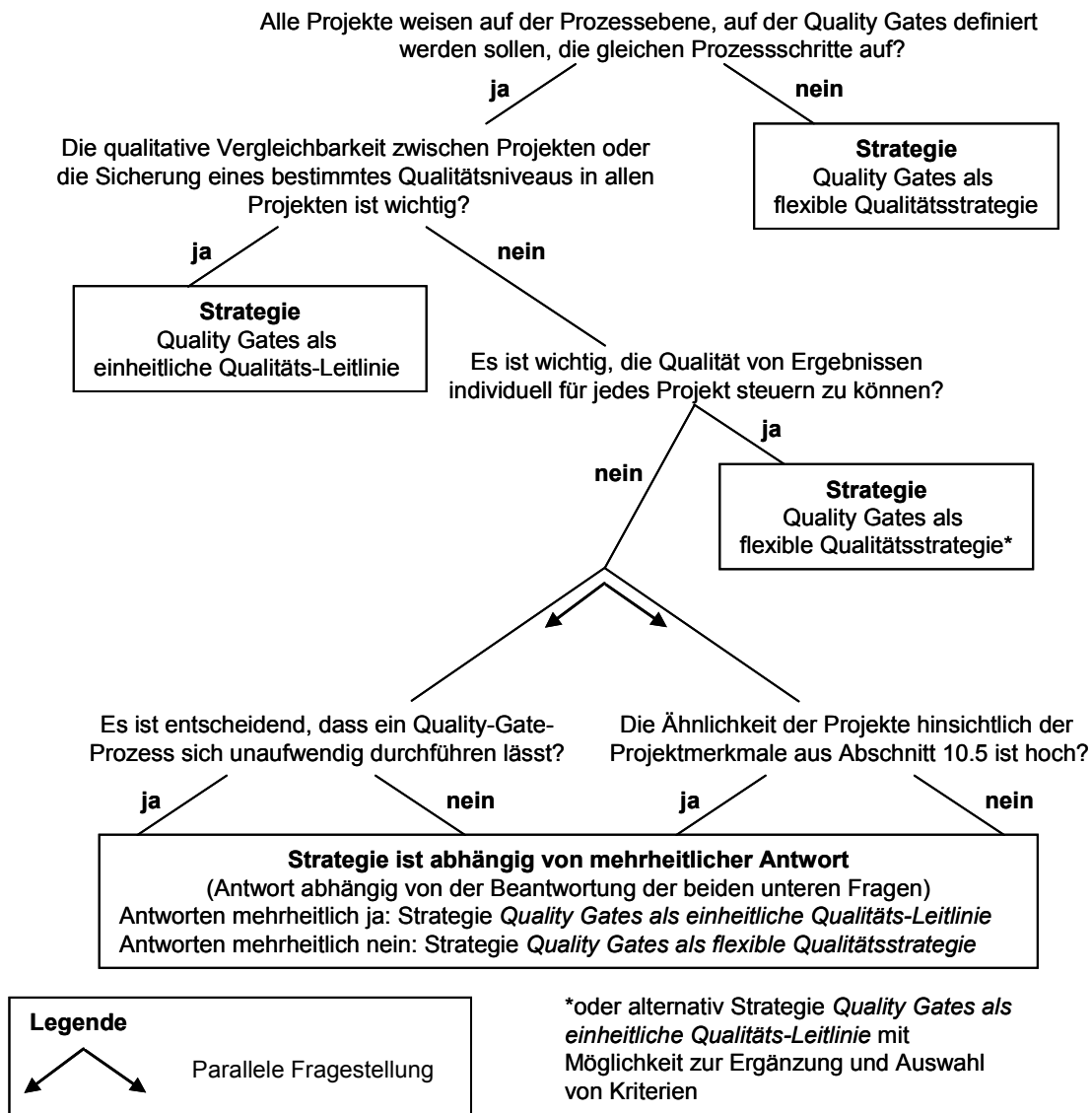


Abbildung 110: Entscheidungsbaum zur Identifikation der grundlegenden Strategie

Je nachdem, wie ein Prozessmanagement die Fragen beantwortet, wird eine bestimmte grundlegende Strategie festgelegt. Im oberen Teil des Entscheidungsbaums kann die Entscheidung bereits eindeutig getroffen werden, wenn entweder die Projekte (in denen ein Quality-Gate-Prozess zum Einsatz kommen soll) keine einheitlichen Prozessschritte aufweisen bzw. die qualitative Vergleichbarkeit der Projekte untereinander oder ein einheitliches Qualitätsniveau wichtig sind. Weiterhin ist die grundlegende Strategie *Quality Gates als flexible Qualitätsstrategie* dann zu wählen, wenn die Qualität von Ergebnissen für jedes Projekt individuell gesteuert werden soll. Alternativ kann es in diesem Fall auch schon ausreichen, die grundlegende Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* zu wählen und die Ergänzung und Auswahl von Kriterien zu erlauben. Das heißt beispielsweise, dass Kriterien individuell festgelegt werden sollen. Kann aus der Beantwortung dieser Fragen keine grundlegende Strategie abgeleitet werden, so kann durch die Beantwortung der beiden Fragen im unteren Teil des Entscheidungsbaumes eine Entscheidung nahe gelegt werden. Im Extremfall kann hierbei jedoch auch die jeweils andere Strategie gewählt werden, wobei sie dann jedoch schwieriger zu etablieren ist.

Nachdem die Entscheidung über die grundlegende Strategie getroffen wurde, kann das Prozessmanagement mit der Ausgestaltung eines Quality-Gate-Referenzprozesses beginnen, der hinsichtlich des Aufwands und Inhalts zum Unternehmen passt. Die Ausgestaltung hängt dabei von der identifizierten grundlegenden Strategie ab. Die Aufgabe der Ausgestaltung kann zunächst durch das Prozessmanagement erfolgen oder aber auch bereits durch ein benanntes Gate-Management.

Im Falle der Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* muss ein Gate-Netzwerk identifiziert werden, das zum Referenzprozess passt (vgl. Abschnitt 6.1 und Abschnitt 0). Dieses Gate-Netzwerk wird dann in allen Projekten verwendet. Ferner wird ein Kriterienkatalog erstellt, der die Kriterien für jedes Quality Gate enthält. Die Erstellung des Kriterienkatalogs kann mit unterschiedlicher Systematik erfolgen (vgl. Abschnitt 7.1). Das Prozessmanagement kann sich ferner dazu entscheiden, die Auswahl und Ergänzung von bzw. um Kriterien zuzulassen, um die Strategie etwas flexibler zu gestalten. Idealerweise wird hierzu ein formalisiertes Tailoring-Verfahren ausgestaltet, das den Prozess-Tailorer bei der Auswahl der Kriterien unterstützt. Des Weiteren sollte eine Entscheidungsunterstützung etabliert werden, um Entscheidungen projektübergreifend ähnlich treffen zu können. Gate-Review und Kriterienerstellung (in diesem Fall Operationalisierung, Ergänzung und Auswahl) müssen ebenfalls in jedem Projekt ähnlich verlaufen. Folglich sind auch die Rollen Gatekeeper, Gutachter, Gate-Verantwortlicher und Ersteller hinsichtlich ihres Profils für alle Projekte gleich. Lediglich das Protokoll kann für jedes Projekt unterschiedlich ausführlich ausfallen (vgl. Tabelle 40).

Falls die grundlegende Strategie *Quality Gates als flexible Qualitätsstrategie* festgelegt wurde, so sind generell alle Quality-Gate-Teilkonzepte anpassbar (vgl. Tabelle 40). Dies spiegelt sich in der Ausgestaltung wider, da stets mehr als eine Variante eines Quality-Gate-Teilkonzeptes angeboten werden kann. Verschiedene Basis-Gate-Netzwerke sind möglich, die durch Operationen modifiziert werden dürfen (vgl. Abschnitt 6.9). Jedoch sind hierbei nicht beliebige Gate-Netzwerke möglich (vgl. Abschnitt 6.1). Hinsichtlich der Kriterienerstellung sind ein Kriterienkatalog oder eine individuelle Erstellung der Kriterien zulässig. Ebenfalls kann die Systematik von der Projektsituation abhängen. So können in einem wichtigen und sehr neuartigen Projekt, die Kriterien systematisch und individuell erstellt werden. Für ein bekanntes Projekt mittlerer Wichtigkeit können die Kriterien hingegen aus einem Kriterienkatalog entnommen und mit geringem Aufwand operationalisiert, ausgewählt oder ergänzt werden. Alle anderen anpassbaren Teilkonzepte können ebenfalls an die Projektsituation angepasst werden.

Trotz der Möglichkeit zu einer flexiblen und anpassbaren Ausgestaltung, darf eine gewisse Leitlinie nicht aus den Augen verloren werden. Generell sollte daher – wenn immer möglich – auch ein ähnlicher Quality-Gate-Prozess in ähnlichen Projektsituationen verwendet werden. So sollten Kriterien in einem festen Kriterienkatalog gesammelt und in Standard-Projekten wiederverwendet werden. Auf diese Weise wird der Aufwand einer (erneuten) Erstellung eingespart und das Erfahrungswissen aus vergangenen Projekten genutzt. Gate-Netzwerke und Varianten anderer anpassbarer Quality-Gate-Teilkonzepte sollten ebenfalls in ähnlicher Projektsituation auch wieder ähnlich verwendet werden. Hierbei hilft wiederum ein formalisiertes Tailoring-Verfahren.

12.2 Unternehmen mit unzureichend ausgestalteten Quality-Gate-Referenzprozess

Viele Unternehmen haben bereits einen Quality-Gate-Referenzprozess etabliert. Dennoch zeigt der Chaos-Report (vgl. Kapitel 0), dass ein hoher Prozentsatz der Projekte scheitert oder nur unter durch Überziehung von Zeit und Kosten zum Ziel geführt werden. Neben verschiedenen Ursachen hierfür (zum Beispiel Unzulänglichkeiten im Referenzprozess der Software-Entwicklung selbst), kann die Ursache auch in einer unzureichenden Ausgestaltung des Quality-Gate-Referenzprozesses liegen. Um mögliche Unzulänglichkeiten in einem Quality-Gate-Referenzprozess aufzudecken, kann die Qualität des jeweiligen Quality-Gate-Referenzprozesses durch das Gate-Management oder auch durch externe Assessoren bewertet werden. Das Ergebnis ist ein Bewertungsprofil, das mögliche Probleme erkennen lässt. Ferner ist es häufig notwendig, die Qualität der eigenen Prozesse (und damit auch die Qualität des etablierten Quality-Gate-Referenzprozesses) potentiellen Auftraggebern aufzuzeigen. Zur Darlegung der Qualität eines Prozesses können ebenfalls die Bewertungsschemata der bekannten Prozessreifegradmodelle CMMI und SPICE genutzt (vgl. Abschnitt 4.7).

Für eine Bewertung ist die Art und Weise, wie ein Quality-Gate-Teilkonzept ausgestaltet ist, zunächst unerheblich. Vielmehr ist von Relevanz, wie deutlich und klar die Ausgestaltung jedes einzelnen Quality-Gate-Teilkonzeptes ist. Jedes ausgestaltbare Teilkonzept lässt sich auf einer dreistufigen ordinalen Skala bewerten:

- **Voll ausgestaltet (●):** Der Quality-Gate-Referenzprozess nimmt zur Ausgestaltung des Teilkonzeptes klar und unmissverständlich Stellung. Dies kann zum Beispiel für eine Rolle eine Profilbeschreibung sein. Für eine Aktivität werden Unteraktivitäten, klare Ein- und Ausgaben und Verantwortlichkeiten definiert. Dokumente werden strukturell beschrieben. Vollausgestaltete Teilkonzepte sind häufig optimiert, sollten jedoch im Zweifelsfall ebenfalls mit dem Framework hinsichtlich ihrer Ausgestaltung abgeglichen werden. Ein voll ausgestaltetes Teilkonzept kann durch die Art und Weise der Ausgestaltung immer noch problembehaftet sein.
- **Teilweise ausgestaltet (⊙):** Die Ausgestaltung ist nicht deutlich formuliert. Beispielsweise sind zwar Rollen definiert und den Aktivitäten zugeordnet, aber eine Profilbeschreibung fehlt oder Aktivitäten sind nur sehr abstrakt beschrieben. Meistens existiert jedoch ein intuitives Verständnis darüber, wie Prozessschritte durchzuführen sind, welche Artefakte wie, wann, und in welcher Form vorliegen müssen und wie Rollen zu besetzen sind. Die Ausgestaltung wurde jedoch nicht immer (eindeutig) formalisiert und verfestigt. Teilweise ausgestaltete Teilkonzepte sind unter Umständen ein Indiz für Unklarheiten im Quality-Gate-Referenzprozess, die bei der Ausführung oder Anpassung des Quality-Gate-Referenzprozesses zu Schwierigkeiten führen können und eine uneinheitliche Verwendung begünstigen. Sie können insbesondere bei der grundlegenden Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* zu Schwierigkeiten führen. Teilweise ausgestaltete Teilkonzepte sind nicht in jeder Situation nachteilig. So kann ein Quality-Gate-Referenzprozess bewusst abstrakt ausgestaltet sein, um beispielsweise eine breite Verwendung in möglichst vielen Unternehmensbereichen zu ermöglichen.
- **Nicht ausgestaltet (○):** Der Quality-Gate-Referenzprozess nimmt zur Ausgestaltung des betreffenden Teilkonzeptes keine Stellung. Drei Ursachen sind hierfür möglich:
 - Die Ausgestaltung wurde vergessen oder war dem Prozessmanagement des Quality-Gate-Referenzprozesses nicht bekannt.
 - Die Ausgestaltung wurde nicht beschrieben, weil es für jeden Unternehmensbereich vollständig anders ausgestaltet werden muss.
 - Das Quality-Gate-Teilkonzept wurde absichtlich nicht ausgestaltet, weil es dem Prozessmanagement nicht wichtig erschien.

Nicht ausgestaltete Teilkonzepte können ein Hinweis für ein Defizit im betrachteten Quality-Gate-Referenzprozess sein, es sei denn, die Ausgestaltung muss für jeden Unternehmensbereich vollständig anders sein.

Unzulänglichkeiten (als Folge der unzureichenden Ausgestaltung) und ihre resultierenden Risiken sind in der folgenden Tabelle aufgeführt.

Unzulänglichkeit	Beschreibung	Mögliche resultierende Probleme
Ungeklärte Gate-Netzwerke	Hierbei ist nicht klar, welche Quality Gates existieren und in welcher Abfolge sie durchlaufen werden müssen.	Es werden unterschiedliche Gate-Netzwerke in unterschiedlichen Projekten verwendet; Qualitätsniveau und Vergleichbarkeit sinken möglicherweise.
Ungeklärte Rollen	Eine Rolle wird von den falschen Personen (aufgrund falscher oder fehlender Profilbeschreibung) besetzt oder fehlt gänzlich.	Je nach Verantwortung der Rolle werden Ergebnisse falsch begutachtet, eine falsche Entscheidung getroffen oder falsche Kriterien erstellt, ergänzt, ausgewählt oder operationalisiert. Ferner wird unter Umständen ein Protokoll nicht oder nur unzureichend erstellt oder der Ablauf der Aktivitäten gestaltet sich zäh (fehlender oder falsch besetzte Rolle Gate-Verantwortlicher).
Ungeklärte Aktivitäten	Das Gate-Review oder die Kriterienerstellung sind nur unzureichend ausgestaltet. Damit ist unklar, durch welche Aktivitäten die Entscheidung zustande kommt und wie die Kriterien erstellt, ergänzt, ausgewählt oder operationalisiert werden.	Ungeeignete Kriterien werden erstellt oder Ergebnisse uneinheitlich bewertet. Qualitätsniveau und Vergleichbarkeit sinken möglicherweise.
Ungeklärtes Protokoll	Der Inhalt des Protokolls ist unklar.	Entscheidungen, Bewertung der Kriterien und Maßnahmen sind später nicht mehr nachvollziehbar.
Ungeklärte Anpassung und ungeklärtes Gate-Management	Hierbei liegen die Unklarheiten darin, welche Elemente auswählbar sind und wie das Tailoring-Verfahren und das Projektmodell ausgestaltet sind. Ein Gate-Management ist nicht etabliert.	Es wird ein ungeeigneter Quality-Gate-Prozess verwendet. Quality-Gate-Prozesse werden uneinheitlich verwendet. Das Qualitätsniveau sinkt. Der Quality-Gate-Referenzprozess wird unter Umständen nicht verbessert.
Ungeklärte Kriterien	Es ist unklar, welcher Art die Kriterien sind. Insbesondere ist unklar, ob generell nicht qualitätsbezogene Kriterien zulässig sind.	Ein Quality-Gate-Prozess wird nicht mehr als Mittel der Qualitätssicherung angesehen, da nicht qualitätsbezogene Kriterien überwiegen. Ferner kann die Möglichkeit, nicht qualitätsbezogene Kriterien gleichzeitig im Gate-Review zu prüfen, ungenutzt bleiben.
Ungeklärte Entscheidungsunterstützung und ungeklärte mögliche Entscheidungen	Es ist unklar, inwieweit Gewichtung und Bewertung der Kriterien zur Entscheidung führen. Weiterhin ist unklar, welche Entscheidungen generell im Quality Gate getroffen werden können.	Entscheidungen werden nicht einheitlich getroffen. Eventuell werden nicht vorgesehene Entscheidungen getroffen. Es können Entscheidungen, die eigentlich getroffen werden können, nicht getroffen. Es ist unklar, wer welche Entscheidung treffen darf. Als Folge kann die Effektivität sinken (vgl. Tabelle 43) oder die Möglichkeit zum Projektvergleichbarkeit kann erschwert werden.

Tabelle 46: Unzulänglichkeiten und resultierende Risiken

Die folgende Tabelle zeigt beispielhaft Bewertungsprofile für die drei relevanten Quality-Gate-Ansätze dieser Arbeit (siehe rechteste drei Spalten): den Quality-Gate-Referenzprozess nach Pfeifer, den Quality-Gate-Referenzprozess des V-Modells XT und den Quality-Gate-Referenzprozess des Stage-Gate-Prozesses.

Paket	Ausgestaltbares Teilkonzept	Quality-Gate-Referenzprozess nach Pfeifer	Quality-Gate-Referenzprozess des V-Modells XT	Quality-Gate-Referenzprozess des Stage-Gate-Prozesses
Struktur	Gate-Netzwerk	●	●	●
	Inhalt			
Inhalt	Kriterienerstellung	●	⊙	⊙
	Ersteller	●	●	⊙
	Kriterien	⊙	○	●
Review	Gate-Review	⊙	●	⊙
	Gate-Verantwortlicher	○	●	⊙
	Gutachter	●	●	●
	Projektvertreter	⊙	●	⊙
	Protokollant	○	●	○
	Protokoll	⊙	●	●
Steuerung	Mögliche Entscheidungen	●	●	●
	Gatekeeper	●	●	●
	Entscheidungsunterstützung	⊙	○	⊙
Anpassung	Gate-Management	○	⊙	⊙
	Prozess-Tailorer	○	●	○
	Tailoring-Verfahren	○	●	⊙
	Auswählbare Elemente	○	●	●
	Projektmodell	○	●	⊙
Legende				
● voll ausgestaltet ⊙ teilweise ausgestaltet ○ nicht ausgestaltet				

Tabelle 47: Bewertungsprofile für die relevanten Quality-Gate-Referenzprozesse

Die drei Bewertungsprofile aus Tabelle 47 lassen verschiedene Problempunkte erkennen. So folgt zwar der Quality-Gate-Referenzprozess nach Pfeifer der Strategie *Quality Gates als flexible Qualitätsstrategie*, weist aber dennoch unter anderem erhebliche Lücken bei den Teilkonzepten auf, die sich mit der Anpassung beschäftigen. Insbesondere ist unklar, welche Elemente auswählbar sind. Der Quality-Gate-Referenzprozess des V-Modells XT lässt hingegen offen, welcher Art die Kriterien sind und in welcher Weise die Entscheidung unterstützt werden kann. Probleme können sich hier ebenso durch eine teilweise Ausgestaltung der Kriterienerstellung ergeben. Der Quality-Gate-Referenzprozess des Stage-Gate-Prozesses zeigt bei den wichtigsten Quality-Gate-Teilkonzepten nur wenige Schwächen. Probleme können sich dadurch ergeben, dass die Rolle des Prozess-Tailorers nicht ausgestaltet ist. Ebenso können unter anderem die teilweise ausgestalteten Teilkonzepte *Kriterienerstellung* und *Ersteller* zu Problemen führen. Bei allen drei Ansätzen ist ferner erkennbar, dass das Gate-Management unzureichend ausgestaltet ist.

Auf Grundlage eines Bewertungsprofils kann das Gate-Management nun die resultierenden Risiken identifizieren. Das Gate-Management kann nun entweder alle unzureichend ausgestalteten Quality-Gate-Teilkonzepte oder lediglich eine Teilmenge dieser auf Basis des Frameworks ausgestalten. Ferner ist es auch sinnvoll, die bereits voll ausgestalteten Teilkonzepte gegen das Framework zu prüfen, um mögliche Unterschiede in der Art und Weise der Ausgestaltung zu identifizieren.

Ferner ist es, wie oben bereits angedeutet, möglich, die Qualität eines Quality-Gate-Referenzprozesses mit einem Bewertungsprofil einem potentiellen Auftraggeber gegenüber aufzuzeigen.

12.3 Unternehmen mit Quality-Gate-Referenzprozess mit Verbesserungspotential

Unternehmen, die eine Verbesserung ihres Quality-Gate-Referenzprozesses anstreben, verfügen schon über einen nahezu optimalen und vollständig ausgestalteten Quality-Gate-Referenzprozess oder haben nur wenige Teilkonzepte absichtlich nur teilweise bzw. nicht ausgestaltet. Unternehmen können dann ihren Quality-Gate-Referenzprozess hinsichtlich der Effizienz und Effektivität verbessern.

Möglichkeiten zur Verbesserung umfassen für Unternehmen in dieser Ausgangslage gemäß Kapitel 11 die folgenden Punkte:

- Umsetzung eines Erfahrungskreislaufes speziell für die Optimierung des Kriterienkatalogs oder weiterer Quality-Gate-Teilkonzepte.
- Vermeidung einer Sitzung durch die Umsetzung der vollständigen Einsatzstrategie mittels des Softwarewerkzeuges NetQGate.
- Einsparung von Rollen durch den Einsatz von NetQGate (Protokollant und Gate-Verantwortlicher) bei vollständiger Einsatzstrategie.
- Nutzung von NetQGate als Dokumentenmanagementsystem bei hybrider und vollständiger Einsatzstrategie.
- Nutzung von NetQGate als Planungswerkzeug für Quality-Gate-Prozesse.
- Unterstützung der Entscheidungsfindung durch das Softwarewerkzeug RiskQG.
- Unterstützung der Erstellung und Operationalisierung (insbesondere der Gewichtung) von Kriterien durch das Softwarewerkzeug RiskQG.

Eine Schwierigkeit, die bei der Verwendung von NetQGate auftritt, ist die mangelnde Flexibilität des realisierten Workflows. Eine generische Lösung gestaltet sich nach Hedberg [Hedberg '04] schwierig. Daher muss jedes Unternehmen eine individuelle Lösung entwickeln, falls der Workflow abweicht. Hinsichtlich der ausschließlichen Nutzung als Dokumentmanagement-System (nur hybride Einsatzstrategie) kann NetQGate ohne größere Änderungen genutzt werden.

13 Fallstudien

Im Folgenden werden anhand dreier Fallstudien die Mächtigkeit und der Nutzen des Frameworks aufgezeigt. Die ersten beiden Fallstudien zeigen den Einsatz von Quality Gates in studentischen Softwareprojekten im Lehrumfeld des Fachgebiets Software Engineering der Gottfried Wilhelm Leibniz Universität Hannover. Die dritte Fallstudie zeigt, wie das Tailoring-Verfahren des V-Modells XT durch ein Fuzzy-Inferenz-System realisiert werden kann.

Die Betrachtung von Softwareprojekten im genannten Lehrumfeld kann jedoch nur in leicht eingeschränkter Weise als Beleg für den Nutzen verwendet werden, da sich die Softwareprojekte von „realen“ Projekten im industriellen Umfeld unterscheiden. Die folgende Tabelle 48 zeigt Unterschiede und Parallelen und ihre jeweiligen Auswirkungen auf.

Parallelen	Unterschiede
<ul style="list-style-type: none"> - Die Einstufung als Projekt ist (bis auf die monetären Ressourcen) gemäß Abschnitt 2.2.2 gerechtfertigt. - Die Projekte folgen jeweils einem strukturierten und realistischen Entwicklungsprozess. Die notwendigen Gate-Netzwerke sind daher realistisch. - Ausreichend viele Projekte werden zeitlich parallel durchgeführt. Daher kann auch das Multiprojektmanagement untersucht werden. - Es ist ein externer (nachempfunder) Kunde vorhanden, der an den Ergebnissen des Projektes interessiert ist. - Die personellen Ressourcen sind eingeschränkt, so dass ein Quality-Gate-Referenzprozess nicht beliebig aufwändig ausgestaltet werden kann. Dies ist mit Hinblick auf kleinere Softwareunternehmen eine durchaus realistische Ausgangslage. 	<ul style="list-style-type: none"> - Die Projektdauer ist deutlich kürzer (~3,5 Monate) als die vieler realer Projekte. Folglich muss ein Projekt in der Regel auch deutlich weniger Quality Gates passieren. - Es existieren zwei Sichten: die Lehr- und die Praxissicht. Ein Projekt kann aus Lehrsicht erfolgreich und aus Praxissicht gescheitert sein. Folglich werden Projekte im universitären Lehrumfeld selten abgebrochen. Entscheidungen fallen häufig milder aus als in der Praxis. - Monetäre Ressourcen werden nicht wirklich aufgewendet. Projekte werden daher weniger ernst genommen und selten abgebrochen. - Es existiert keine ausgeprägte Qualitäts- und Projektmanagement-Hierarchie. Diese Hierarchie ist künstlich zu schaffen und ist sehr flach. - Studenten sind weniger erfahren und nehmen Softwareprojekte im universitären Umfeld weniger ernst. Daher müssen Quality Gates häufiger wiederholt und Ergebnisse nachgeliefert werden. - Ein Projekt wird durchgehend von einer Gruppe bearbeitet. Folglich wechseln die internen Kunden nicht. Kriterien können daher nicht von internen Kunden kommen.

Tabelle 48: Vergleich zwischen Projekten im universitären Lernumfeld und in Softunternehmen

13.1 Quality-Gate-Referenzprozess in studentischen Softwareprojekten

Im Fachgebiet Software Engineering der Gottfried Wilhelm Leibniz Universität Hannover wurde über rund vier Jahre hinweg in 36 Softwareprojekten vorrangig unter Beteiligung von Bachelor-Studenten der Informatik ein Quality-Gate-Referenzprozess eingesetzt [Lübke et al. '06, Lübke et al. '04]. Im Rahmen von 16 dieser Projekte wurde ebenfalls das Softwarewerkzeug *NetQGate* [Flohr '06] einge-

setzt. Um die Projekte so realistisch wie möglich gestalten zu können, wurde das fiktive Softwareunternehmen *FunGate* geschaffen. Hierzu wurde eine fiktive Projekt- und Qualitätsmanagement-Hierarchie aufgebaut. Die Hierarchie orientierte sich dabei an der Projektorganisation aus Abschnitt 3.1.1 und 3.3.2. Dementsprechend existierte eine Geschäftsleitung, ein Projektmanager und ein Qualitätsmanager und für jedes Projekt je ein Projektleiter und Qualitätsbeauftragter. Jedes Projekt hat weiterhin einen externen Kunden, der seine Rolle so realistisch wie möglich spielen sollte. Dies wurde vorrangig dadurch erreicht, dass der jeweilige Kunde die zu entwickelnde Software auch wirklich später nutzen wollte und daher ein reales Interesse am Projekterfolg hatte.

Durch Quality Gates sollte vor allem ohne großen Aufwand ein identisches Qualitätsniveau in allen studentischen Softwareprojekten sichergestellt und eine Möglichkeit zum Eingriff für das Qualitätsmanagement geschaffen werden. Folglich wurde der Quality-Gate-Referenzprozess auf Basis der grundlegenden Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* ausgestaltet.

Aufgrund der eingeschränkten personellen Ressourcen wurden jedoch alle Rollen (mit Ausnahme der Projektvertreter und Gatekeeper) durch den Qualitätsmanager wahrgenommen. Die Rolle des Gatekeepers wird gleichzeitig sowohl durch den Qualitätsmanager als auch durch die Geschäftsleitung besetzt. Die Besetzung der Rollen stellt jedoch keine unrealistische Ausgangslage bei eingeschränkten personellen Ressourcen dar (vgl. Tabelle 48).

Ein Quality-Gate-Referenzprozess wird seit dem Jahr 2004 eingesetzt und gepflegt. Allerdings wurden erst durch die Etablierung eines Gate-Managements im Wintersemester 2006/07 entscheidende Änderungen in der Ausgestaltung vorgenommen. Darüber hinaus kommt seit dem Wintersemester 2006/07 das Softwarewerkzeug NetQGate zum Einsatz (vgl. Abschnitt 11.5).

Tabelle 49 zeigt anhand von Bewertungsprofilen die Entwicklung der Qualität der Ausgestaltung seit 2004. Dabei ist ebenfalls die Planung für das Wintersemester 2008/09 zu sehen. Erkennbar ist, dass fast alle identifizierten Defizite in der Ausgestaltung nach und nach behoben wurden.

Pa- ket	Ausgestaltbares Teilkonzept	Jahr 2004	Jahr 2006/07	Jahr 2007/08	Jahr 2008/09 (in Planung)
Struk- tur	Gate-Netzwerk	●	●	●	●
	Inhalt				
Inhalt	Kriterienerstellung	●	●	●	●
	Ersteller	⊙	●	●	●
	Kriterien	⊙	⊙	⊙	⊙
Review	Gate-Review	⊙	⊙	⊙	●
	Gate-Verantwortlicher	○	●	●	●
	Gutachter	⊙	●	●	●
	Projektvertreter	⊙	●	●	●
	Protokollant	⊙	●	●	●
	Protokoll	●	●	●	●
Steuerung	Mögliche Entscheidungen	⊙	⊙	⊙	●
	Gatekeeper	⊙	●	●	●
	Entscheidungs- unterstützung	⊙	⊙	⊙	●
Anpassung	Gate-Management	⊙	●	●	●
	Prozess-Tailorer	○	●	●	●
	Tailoring-Verfahren	⊙	⊙	⊙	●
	Auswählbare Elemente	⊙	⊙	⊙	●
	Projektmodell	⊙	⊙	⊙	●
Legende					
● voll ausgestattet ⊙ teilweise ausgestattet ○ nicht ausgestattet					

Tabelle 49: Bewertungsprofile des Quality-Gate-Referenzprozesses in verschiedenen Jahren

In den folgenden Abschnitten wird die Ausgestaltung des Quality-Gate-Referenzprozesses im Wintersemester 2007/08 beschrieben. Gegebenenfalls wird dabei auf geplante Änderungen der Ausgestaltung für das Wintersemester 2008/09 eingegangen.

13.1.1 Ausgestaltung der strukturellen Konzepte

Alle Projekte folgten einem wasserfallartigen Vorgehen, das sich wie darstellt:

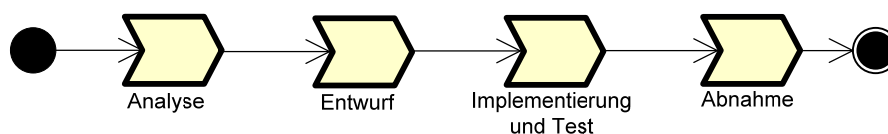


Abbildung 111: Aufbau des Entwicklungsprozesses

Für diesen Entwicklungsprozess wurde das folgende Gate-Netzwerk festgelegt:

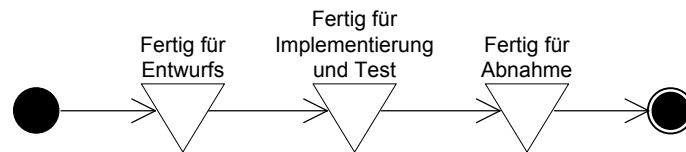


Abbildung 112: Verwendetes Gate-Netzwerk

Für die Wahl dieses Gate-Netzwerkes sprachen verschiedene Gründe:

- Alle Projekte erstreckten sich über einen Zeitraum von 3,5 Monaten, daher war keine höhere Anzahl an Quality Gates möglich und sinnvoll.
- Die parallele Durchführung von bis zu neun Projekten bei gleichzeitig beschränkten personellen Ressourcen ließ keine höhere Anzahl an Quality Gates zu. Zusammengerechnet mussten alle Projekte eines Jahrgangs innerhalb von 3,5 Monaten 24 bis 27 Quality Gates passieren (vgl. Tabelle 50).
- Der Entwicklungsprozess war nur – bis auf ein bis zwei vorgegebene technische Reviews – abstrakt auf Aktivitätsebene definiert. Weitere Unteraktivitäten wurden nicht vorgegeben. Somit konnten hier auch keine Quality Gates ergebnisorientiert platziert werden.

Tabelle 50 illustriert wie hoch der geschätzte Aufwand durch den Einsatz von Quality Gates ist. Im vorliegenden Fall reicht es, den Aufwand für die Gate-Reviews zu betrachten, da für die Kriterienerstellung selbst nur einmalig Aufwand angefallen ist. Dabei ist nicht nur durch die ursprünglich eingeplanten Gate-Reviews (Anzahl der Projekte \times 3) zeitlicher Aufwand entstanden, sondern auch durch notwendige Nachprüfungen im Rahmen der bedingten Projektfortsetzung und durch wiederholte Quality Gates. Weiterhin haben verschiedene Varianten des Gate-Reviews (mit und ohne Einsatz des NetQGate-Systems) und eine verschieden hohe Anzahl von Sitzungsteilnehmer zu unterschiedlich hohen Aufwänden in den verschiedenen Jahren geführt.

Um ausschließlich den Einfluss von NetQGate vergleichen zu können, wurde der zeitliche Aufwand pro Projekt in verschiedenen Jahren ebenfalls für zwei Sitzungsteilnehmer berechnet, obwohl real drei bzw. fünf Sitzungsteilnehmer involviert waren. Es zeigt sich, dass allein durch den Einsatz von NetQGate der Aufwand auf bis zu 57% im Vergleich zum Jahr 2004 reduziert werden konnte. Der Einsatz von NetQGate und eine gleichzeitige Verringerung der Anzahl der Sitzungsteilnehmer reduzieren den zeitlichen Aufwand pro Projekt sogar auf rund 28%.

Insbesondere fallen bei der Betrachtung Schwankungen in der Anzahl der bedingten Projektfortsetzungen auf. Sie können wie folgt erklärt werden:

- Die Schwankungen sind zufällig bedingt, da die betrachtete Stichprobengröße dreier Zeiträume nicht groß genug ist.
- Bedingte Projektfortsetzungen werden häufig schon bei kleineren Mängeln ausgesprochen. Manchmal bewertet ein Gutachter ein Kriterium – trotz leichter „Nichterfüllung“ – als erfüllt ein anderer Gutachter jedoch nicht. Die Schwankungen können daher von einer unterschiedlichen personellen Besetzung der Gutachter-Rollen herrühren.

Aspekt	Jahr 2004	Jahr 2006/07	Jahr 2007/08
Anzahl Projekte	9	8	8
ursprünglich eingeplante Quality Gates	27 [▼]	16 [◆] + 8 [▲]	16 [◆] + 8 [▲]
Anzahl Teilnehmer an der Sitzung des Gate-Reviews (andere Rollen + Projektvertreter)	3 + 2	2 + 1	1 + 1
Anzahl bedingter Projektfortsetzungen	19 [■]	9 [●]	15 [●]
Anzahl wiederholter Quality Gates	4 [▼]	—*	—*
Aufwand in Vielfachen der Sitzung bei realer Rollenverteilung pro Projekt / Aufwand im Vergleich zum Jahr 2004 in %	27,57 / 100%	9,93 / 36,02%	7,6 / 27,56%
Aufwand in Vielfachen der Sitzung bei Rollenverteilung 1 + 1 pro Projekt / Aufwand im Vergleich zum Jahr 2004 in %	12,86 / 100%	7,3 / 56,76%	7,6 / 59,09%
Resultierender zeitlicher Gesamtaufwand bei Rollenverteilung real / 1 + 1 pro Projekt in Stunden (bei Sitzungslänge 20 Minuten)	9,2 h / 4,3 h	3,3 h / 2,4 h	2,5 h / 2,5 h
Legende			
▼ Gate-Review folgt formaler Inspektion (ohne NetQGate)			
◆ Gate-Review folgt formaler Inspektion (hybride Einsatzstrategie)			
▲ Gate-Review folgt formaler Inspektion (vollständige Einsatzstrategie)			
■ Bedingte Projektfortsetzung ohne Unterstützung durch NetQGate			
● Bedingte Projektfortsetzung unterstützt durch NetQGate			
* Es wurde kein Gebrauch von der Entscheidung „Wiederholung des Quality Gates“ gemacht.			

Tabelle 50: Übersicht über den geschätzten Aufwand für die Quality Gates

Den Schätzungen aus Tabelle 50 liegt der geschätzte zeitliche Aufwand für verschiedene Varianten des Gate-Reviews bzw. für die Entscheidungen „bedingte Projektfortsetzung“ und „Wiederholung des Quality Gates“ zu Grunde (vgl. Tabelle 51). Dieser Aufwand setzt sich gemäß Abschnitt 8.1.2 aus den Aufwänden der Organisations-, Prüfungs- und Abschlussaktivitäten zusammen. Der Aufwand ist dabei relativ zum Aufwand der Prüfungsaktivitäten der formalen Inspektion für den Qualitätsmanager angegeben. Diese Prüfungsaktivitäten entsprechen einem Faktor von 1. Ein Faktor von 1 bedeutet einen Aufwand von 20 Minuten. Den Schätzungen liegen die folgenden Fakten und Annahmen zu Grunde:

Fakten:

- Die formale Inspektion fand jeweils ohne individuelle Vorbereitung statt. Leistungsdaten wurden nicht erhoben.
- Die Rolle Projektvertreter wurde durch eine Person besetzt. Alle weiteren Rollen im Gate-Review (Gutachter, Gatekeeper, Protokollant und Gate-Verantwortlicher) wurden zusammengefasst und durch den Qualitätsmanager wahrgenommen. Die Rolle des Gatekeepers für weitreichende Entscheidungen wurde durch die Geschäftsleitung besetzt – diese Besetzung ist jedoch für die Betrachtungen des Aufwands unerheblich, da von weitreichenden Entscheidungen nahezu kein Gebrauch gemacht wurde.
- Bei der bedingten Projektfortsetzung musste kein Projektvertreter anreisen. Etwaige Kommunikation erfolgte entweder über NetQGate oder per E-Mail.
- Der Qualitätsmanager sitzt dauerhaft am Ort der Sitzung und muss daher nicht an- und abreisen.

Annahmen:

- Die Projektvertreter müssen an- und abreisen. Die hierfür aufgewendete Zeit entspricht gemäß Abschnitt 8.1.2 im Mittel ungefähr der Länge einer Sitzung.
- NetQGate halbiert den Aufwand für die Organisationsaktivitäten, da Ergebnisse und weitere Informationen zentral vorliegen. Der Aufwand für die Abschlussaktivitäten (ohne Nachverfolgung) beim Einsatz von NetQGate ist unerheblich, da die Verteilung und Archivierung des Protokolls automatisch geschieht.
- Bei den Prüfungsaktivitäten unterstützt NetQGate die Erstellung des Protokolls. Weiterhin sind alle Ergebnisse und Kriterien zentral verfügbar und der Zugriff sofort möglich. Die Aufwandsreduktion kann mit 20% angenommen werden.

		Aufwandsfaktoren für Projektvertreter			Aufwandsfaktoren für Projektvertreter			Ge- samt
Variante des Gate-Reviews bzw. Entscheidung		Org.	Prüf.	Ab.	Org.	Prüf.	Ab.	
ohne NetQGate	Formale Inspektion	0,1	2	0	0,2	1	0,1	3,4
	Bedingte Projektfortsetzung	0,1	0,1	0	0,1	0,25	0,1	0,65
	Wiederholung des Quality Gates	0,1	1,75	0	0,2	0,75	0,1	2,9
mit NetQGate	Formale Inspektion (Hybride Einsatzstrategie)	0,05	1,8	0	0,1	0,8	0	2,75
	Formale Inspektion (Vollständige Einsatzstrategie)	0,05	0,4	0	0,1	0,8	0	1,35
	Bedingte Projektfortsetzung	0,05	0,1	0	0,05	0,2	0	0,4
	Wiederholung des Quality Gates (Hybride Einsatzstrategie)	0,05	1,6	0	0,1	0,6	0	2,35
	Wiederholung des Quality Gates (Vollständige Einsatzstrategie)	0,05	0,6	0	0,1	0,6	0	1,05
Legende								
Org. Organisationsaktivitäten								
Prüf. Prüfungsaktivitäten (inklusive An- und Abreisezeit für Sitzung)								
Ab. Abschlussaktivitäten (ohne Nachverfolgung) Nachverfolgung ist separat als bedingte Projektfortsetzung aufgeführt.								

Tabelle 51: Geschätzter relativer zeitlicher Aufwand für Varianten des Gate-Reviews

13.1.2 Ausgestaltung der inhaltlichen Konzepte

Der Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* folgend wurde ein fester Kriterienkatalog aufgestellt, der für jedes der drei Quality Gates die zu prüfenden Kriterien enthält. Die Kriterien wurden unaufwändig durch den Qualitätsmanager über ein erfahrungsbasiertes Vorgehen erstellt. Infolgedessen weist der Kriterienkatalog mit hoher Wahrscheinlichkeit Lücken auf. Durch die fortwährende Pflege wird diese Lücke jedoch zukünftig geschlossen werden können. Der Kriterienkatalog wird seit der erstmaligen Ausgestaltung gepflegt und enthält momentan 62 Kriterien. Die Kriterien stellten dabei Forderungen an verschiedene Dokumente (vgl. Abbildung 113).

Die erstellten Kriterien sind in verschiedene Blöcke unterteilt. So existiert ein Block, der die Kriterien enthält, die für alle Ergebnisse gelten. Weiterhin existiert für jedes Ergebnis ein eigener Block. Die Kriterien sind momentan weder auswählbar noch ergänzbar. Dies stellt mitunter ein Problem dar, da nicht immer alle Kriterien anwendbar sind. Ferner konnte immer wieder festgestellt werden, dass für

bestimmte Projekte weitere Kriterien sinnvoll gewesen wären. Die Operationalisierung der Kriterien findet im Gate-Review statt. Dies führt jedoch dazu, dass Kriterien unterschiedlich von den Gutachtern operationalisiert werden. Hierdurch wird die Vergleichbarkeit hinsichtlich des qualitativen Fortschritts – im Moment jedoch vermutlich nur leicht – eingeschränkt.

Ein vorhandenes Problem – wenngleich kein gravierendes – stellt die teilweise Ausgestaltung des Quality-Gate-Teilkonzeptes *Kriterien* dar. Es besteht zwar ein Konsens darüber, dass vorrangig qualitätsorientierte Kriterien geprüft werden sollen. Allerdings sind weitere Kriterien weder ausgeschlossen noch ausdrücklich zugelassen und benannt. Im universitären Lehrumfeld dürften die meisten sekundären Kriterien ohnehin nur eine untergeordnete Rolle spielen, da monetäre Ressourcen keine Rollen spielen. Eine weitere Ausgestaltung des Quality-Gate-Teilkonzeptes *Kriterien* erscheint daher nicht notwendig.

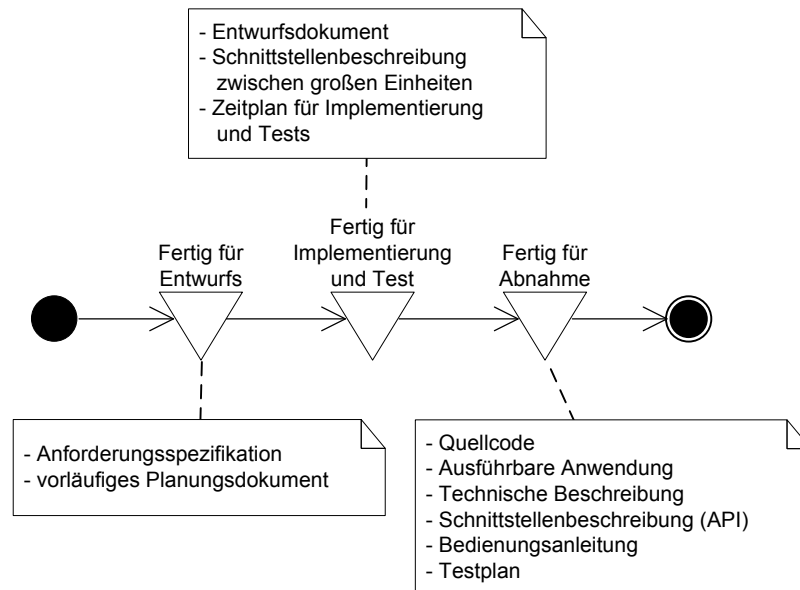


Abbildung 113: Geforderte Dokumente aufgeschlüsselt nach Quality Gates

13.1.3 Ausgestaltung der Reviewkonzepte

Für die ersten beiden Quality Gates *Fertig für Entwurf* und *Fertig für Implementierung und Test* folgt das Gate-Review momentan dem Vorgehen der formalen Inspektion. Hierbei wird das Softwarewerkzeug NetQGate im Rahmen der hybriden Einsatzstrategie eingesetzt. Für das dritte Quality Gate *Fertig für Abnahmephase* wird NetQGate in der vollständigen Einsatzstrategie angewendet. Die Aufteilung ist in dieser Form gewählt, um in den ersten beiden Quality Gates die Wichtigkeit von Quality Gates für das jeweilige Projektteam aufzuzeigen (vgl. hierzu Auswirkungen des Einsatzes von NetQGate im Abschnitt 11.5.6).

Die verwendete formale Inspektion ist unaufwändig ausgestaltet. Es fehlt eine individuelle Vorbereitung. Leistungsdaten werden ebenfalls nicht erhoben, da sie nur relevant sind, wenn eine individuelle Vorbereitung stattgefunden hat. Die formale Inspektion weist eine Sitzung auf, um die Relevanz eines Quality Gates im Projektverlauf zu vermitteln. Jede Sitzung wurde jedoch stets auf 20 Minuten gestrafft, um eine personelle Überlastung zu vermeiden. Ein Projekt wurde in der Sitzung durch den Qualitätsbeauftragten des jeweiligen Projektes vertreten. Der Qualitätsmanager übernahm alle verbleibenden notwendigen Rollen.

Abbildung 114, Abbildung 115 und Abbildung 116 zeigen den Ablauf der formalen Inspektion. Die Aktivitätsdiagramme stellen dabei ein deskriptives Modell der formalen Inspektion dar, da der Ablauf des Gate-Reviews im Zeitraum 2007/08 noch nicht verfestigt war. Dennoch existierte eine relativ konkrete Vorstellung über den Ablauf eines Gate-Reviews, wodurch die Bewertung des Quality-Gate-Teilkonzept Gate-Reviews als teilweise ausgestaltet gerechtfertigt ist.

Während der Durchführung der formalen Inspektion traten wiederholt Probleme auf, die in einer zukünftigen Ausgestaltung des Gate-Reviews berücksichtigt werden müssen:

- Die Aktivität *Formale Eingangsprüfung* wurde nicht genau genug definiert. Hierdurch fanden Sitzungen statt, obwohl nicht alle Ergebnisse vorlagen oder schon bekannt war, dass wichtige Kriterien nicht erfüllt wurden. Zukünftig sind daher klare Eingangskriterien zu definieren. Eine mögliche Ausgestaltung könnte vorsehen, dass die Sitzung (und damit alle Aktivitäten nach der formalen Eingangsprüfung) nicht durchgeführt werden dürfen, wenn ein Ergebnis fehlt, das für ein rot gewichtetes Kriterium benötigt wird.
- Kriterien wurden grundsätzlich während der Sitzung operationalisiert. Hierdurch wurden die Kriterien häufig etwas anders interpretiert. Weiterhin wurden Messungen aus Zeitmangel nur oberflächlich durchgeführt. Eine frühere und einheitlichere Operationalisierung und der systematische Einsatz von softwaregestützten Messungen zur effektiveren Durchführung des Gate-Reviews sind daher zu empfehlen. Hierzu sind verschiedene Softwarewerkzeuge zu identifizieren, die die Messung jeweils erleichtern.

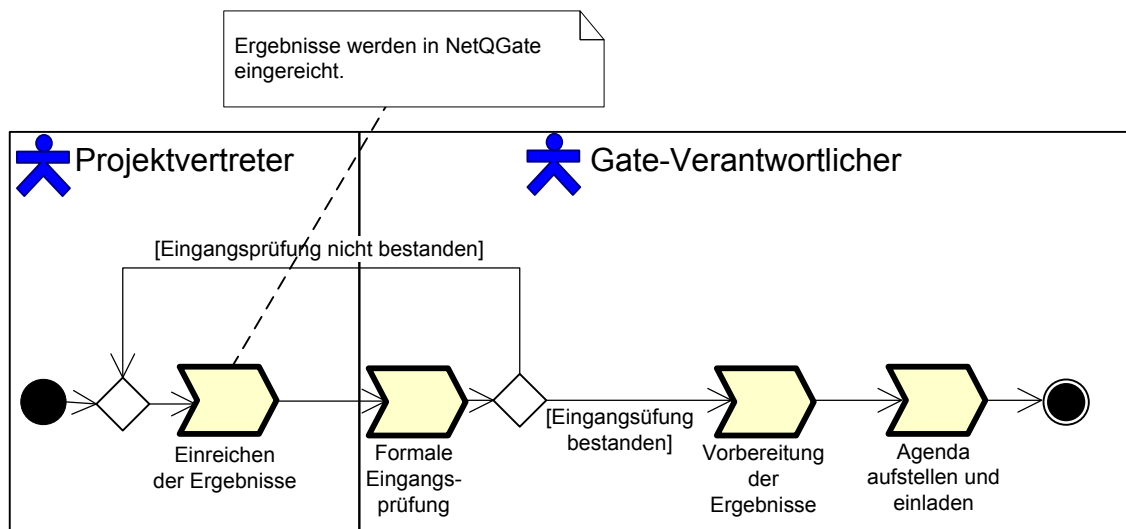


Abbildung 114: Organisationsaktivitäten und zuständige Verantwortliche

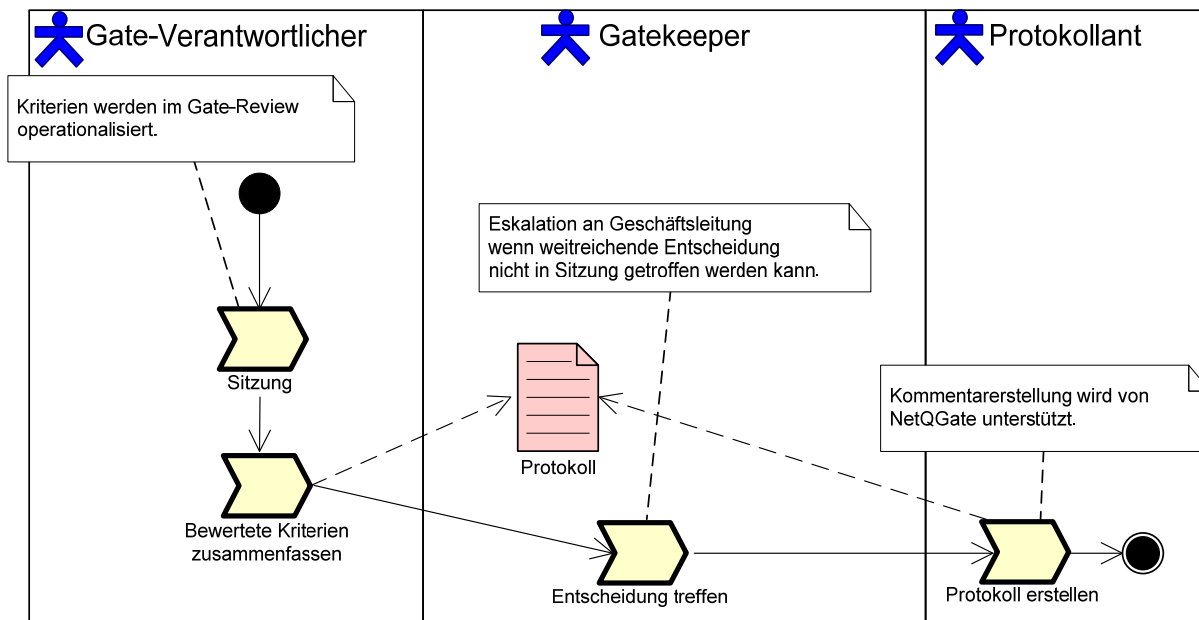


Abbildung 115: Prüfungsaktivitäten und zuständige Verantwortliche

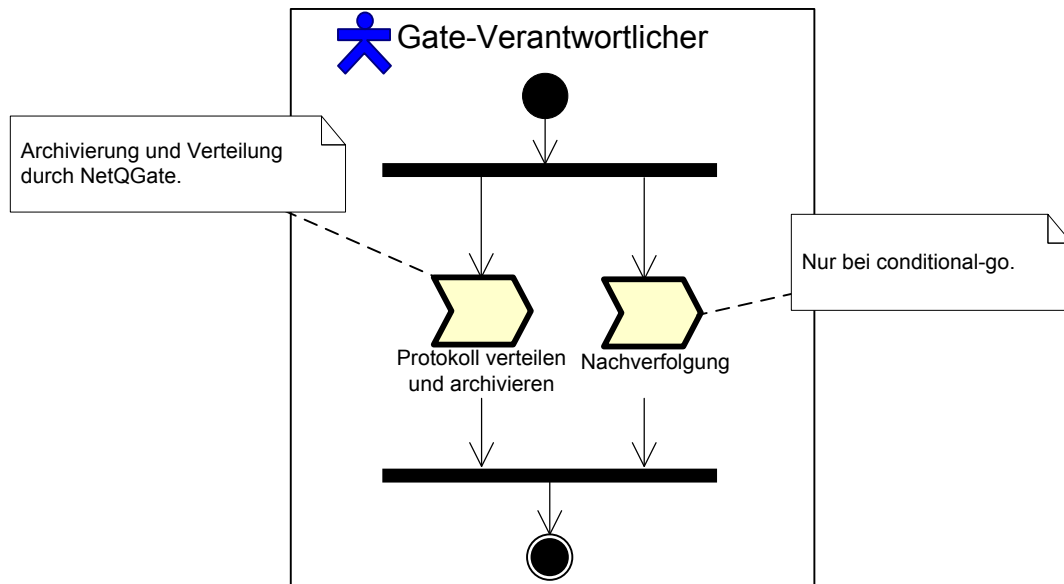


Abbildung 116: Abschlussaktivitäten und zuständige Verantwortliche

Hinsichtlich der vollständigen Einsatzstrategie wurde exakt der Workflow aus Abbildung 108 verwendet.

13.1.4 Ausgestaltung der Steuerungskonzepte

Im Moment ist nicht genau festgelegt, welche Entscheidungen generell getroffen werden dürfen. Ferner ist auch nicht genau festgelegt, welche Entscheidungen durch den Qualitätsmanager und welche Entscheidungen nur durch die Geschäftsleitung getroffen werden dürfen. Im praktischen Einsatz wurden bislang alle Entscheidungen mit Ausnahme der *Zurückstellung des Projektes* getroffen. Daher besteht eine recht klare (jedoch nicht schriftlich fixierte) Vorstellung darüber, welche Entscheidungen grundsätzlich getroffen werden können. Die teilweise Ausgestaltung ist dafür verantwortlich, dass von der Möglichkeit zur Wiederholung eines Quality Gates in den Jahren 2006/2007 und 2007/2008 kein Gebrauch gemacht wurde.

Für das Wintersemester 2008/09 ist geplant, die Entscheidungen *uneingeschränkte Projektfortsetzung*, *bedingte Projektfortsetzung*, *Wiederholung des Quality Gates* und *Projektabbruch* verbindlich schriftlich zu fixieren. Des Weiteren ist zu regeln, welche Rolle welche Entscheidungen treffen dürfen. Hierbei wird verbindlich festgelegt, dass die möglichen Entscheidungen mit Ausnahme eines Projektabbruchs durch den Qualitätsmanager getroffen werden können. Lediglich die Geschäftsleitung (Gatekeeper mit weitreichender Entscheidungsbefugnis) darf ein Projekt abbrechen.

Eine formalisierte und verbindliche Entscheidungsunterstützung existiert momentan nicht. Auch hier ist ein Grundverständnis vorhanden, welche Kriterien erfüllt sein müssen, damit eine bestimmte Entscheidung getroffen werden kann. Dennoch wurden Entscheidungen nicht einheitlich getroffen, obwohl Projekte eine ähnliche Bewertung im Gate-Review erhielten.

Um diese Problematik zu entschärfen, werden zukünftig alle Kriterien entweder mit *rot* oder *gelb* gewichtet. Die Gewichtungen werden im Kriterienkatalog hinterlegt. Darauf aufbauend wird zukünftig die folgende verbindliche und formalisierte Entscheidungsunterstützung λ eingesetzt:

- λ liefert *go* zurück, wenn alle *rot* oder *gelb* gewichteten Kriterien erfüllt sind.
- λ liefert *conditional-go* zurück, wenn mindestens ein *gelb* gewichtetes Kriterium nicht erfüllt ist, aber alle *rot* gewichteten Kriterien erfüllt sind.
- λ liefert *repeat-gate* zurück, wenn mindestens ein *rot* gewichtetes Kriterien nicht erfüllt ist, aber mehr als die Hälfte der *rot* gewichteten Kriterien erfüllt sind.
- λ liefert *kill* zurück, wenn mehr als die Hälfte der *rot* gewichteten Kriterien nicht erfüllt sind.

Ein von der Entscheidungshilfe zurückgelieferter Projektabbruch ist eine überaus drastische Maßnahme. Sie wird daher nicht als verbindlich, sondern im Gegensatz zu den anderen Entscheidungen nur als Empfehlung definiert.

Die Gewichtung kann mittels NetQGate direkt und dauerhaft an die Kriterien geheftet werden (vgl. Abschnitt 11.5.3). Abbildung 117 (Bereich *Guidelines*) zeigt beispielhaft die Verfestigung einer Gewichtung für ein Kriterium in NetQGate.

Abbildung 117: Verfestigung der Gewichtung eines Kriteriums

13.1.5 Ausgestaltung der Anpassungskonzepte

Der betrachtete Quality-Gate-Referenzprozess folgt der grundlegenden Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie*. Daher erscheint eine Möglichkeit zur Anpassung des Quality-Gate-Referenzprozesses an verschiedene Projektsituationen von untergeordneter Bedeutung. Sie ist gemäß Tabelle 40 ohnehin nur für die Quality-Gate-Teilkonzepte *Checkliste* und *Protokoll* möglich. Da das Protokoll bei der einheitlichen Verwendung von NetQGate als Nebenprodukt ohne größeren Aufwand entsteht, ist nur die Checkliste als anpassbares Quality-Gate-Teilkonzept interessant.

Die momentane Ausgestaltung erlaubt weder die Auswahl noch die Ergänzung von Kriterien. Jedoch treten immer wieder Kriterien auf, die nur bedingt oder gar nicht zu bestimmten Projektsituationen passen. Generell ist festzustellen, dass der Großteil der Kriterien zu allen Projektsituationen passt. Die Planung für das Wintersemester 2008/09 sieht daher vor, die Auswahl zuzulassen. Die Bestimmung der geeigneten Kriterien soll über ein Tailoring-Verfahren auf Basis eines Fuzzy-Inferenz-Systems erfolgen. Daher sind ebenfalls ein Projektmodell und die auswählbaren Elemente (in dieser Situation nur die Checkliste) auszugestalten.

So ist zum Beispiel ein Kriterium, das für alle Methoden eines Quellcodes einen McCabe-Wert ≤ 10 fordert, insbesondere vom Standpunkt zukünftiger Wartungsarbeiten her, sinnvoll. Im Kontext eines Softwareprojektes, dessen Ziel die Entwicklung von eingebetteter Software ist, kann die erforderliche Aufteilung des Quellcodes in kleinere Methoden jedoch zu einem höheren (aber meist unerwünschten) Speicherverbrauch führen.

Nach Tabelle 42 beeinflussen fast alle Projektmerkmale die Checkliste. Konkret haben alle Projektmerkmale mit Ausnahme des Merkmals *Stabilität der Anforderungen* einen Einfluss. Ein Projektmodell enthält daher maximal alle Projektmerkmale bis auf das Projektmerkmal *Stabilität der Anforderungen*. Die Projektmerkmalswerte für jedes Projektmerkmal wurden anhand bereits abgeschlossener Projekte ermittelt. Tabelle 52 zeigt das maximale Projektmodell für die studentischen Softwareprojekte.

Qualitativ/ Quantitativ	Kategorie	Projektmerkmale
Quantitativ (nominale Skalen)	Mission des Projektes	Neuentwicklung und Weiterentwicklung
	Projektgegenstand	<u>Eingebettete Software</u> , <u>Komplexes System</u> , <u>andere Software</u>
	Konkreter Einsatzzweck	Spiele-Entwicklung, Spiel, Projektmanagement-Tool, Verwaltungssoftware Universität, Medizinisches Instrument, Verwaltungssoftware, Robotersteuerung, Modellierungstool, Webapplikation, Maschinensteuerung, Server, Desktop-Applikation, File-Editor, Qualitätsmanagement-Tool, Workflow-Werkzeug, Simulation
	Benötigte Technologien und Methoden	Programmiersprachen: Java, PHP Zielarchitekturen: Framework, <u>Client-Server</u> , eingebettete Software, Plug-in, <u>SOA</u> , MVC Werkzeuge: Eclipse Methoden: Grafik-, <u>Netzwerk</u> -, Datenbank- und Webprogrammierung, Protokoll- und Workflow-Entwicklung weitere benötigte Technologien: Applet, Drahtlose Kommunikation, Web-Framework, J2EE, XML
	Qualitäts-Anforderungen	Effizienz, Flexibilität, Integrität, Korrektheit, Portabilität, Testbarkeit, Wartbarkeit, Wiederverwendbarkeit, Verwendbarkeit, Zuverlässigkeit
Qualitativ (ordinale Skalen) oder Quantitativ	Neuartigkeit der Technologien und Methoden	bekannt, größtenteils bekannt, größtenteils unbekannt, unbekannt
	Benötigte Ressourcen und Umfang	Umfang: einfach, mittel, komplex, sehr komplex
	Strategische Relevanz für das Unternehmen/Abteilung	unwichtig, wichtig, sehr wichtig
	Risiko des Projektes (Wahrscheinlichkeit des Scheiterns des Projektes)	gering, möglich, hoch, sehr hoch
	Potenzielles Schadensausmaß im Falle eines Softwarefehlers	keiner, gering, finanzieller Schaden, Verletzung von Menschen, ein Menschenleben, viele Menschenleben

Tabelle 52: Maximales Projektmodell für die studentischen Softwareprojekte

Das so aufgestellte Projektmodell ist als maximal anzusehen. Es müssen daher nur für die Projektmerkmalswerte Zugehörigkeitsfunktionen aufgestellt werden, für die auch eine Regel (also eine Erfahrung) vorhanden ist. In Tabelle 52 sind diejenigen Projektmerkmalswerte unterstrichen, für die eine Zugehörigkeitsfunktion zu bilden ist.

Da aus der Vergangenheit noch nicht ausreichend viele Erfahrungen zu Kriterien vorliegen, erfolgt die Entwicklung des Regelwerks des Tailoring-Verfahrens nach der *Minimalen Anpassungsstrategie*. Die Ausgestaltung erfolgt aus diesem Grund nach und nach. Erste Regeln können im Moment jedoch schon erstellt werden. Dabei zeigt sich, dass nur ein Bruchteil der Projektmerkmalswerte in Regeln benutzt wird (vgl. auch Tabelle 52). Momentan liegen die folgenden Erfahrungen vor:

- Wenn eingebettete Software entwickelt wird, dann ist die Einhaltung eines McCabe-Wertes ≤ 10 zum Quality Gate *Fertig für Abnahme* verzichtbar.
- Wenn eingebettete Software entwickelt wird, dann ist die Vorlage von Mockups oder Prototypen der grafischen Schnittstelle (GUI) zum Quality Gate *Fertig für Implementierungsphase* verzichtbar.
- Wenn eingebettete Software entwickelt wird, dann sind Abschätzungen zum Speicherverbrauch während der Laufzeit unverzichtbar und zum Quality Gate *Fertig für Implementierung und Test* vorzulegen.
- Wenn die Architekturen *SOA* oder *Client-Server* verwendet werden, dann sind Angaben zur physischen Verteilung der einzelnen Softwarekomponenten unverzichtbar und zum Quality Gate *Fertig für Implementierung und Test* vorzulegen.
- Wenn über ein Netzwerk kommuniziert werden muss oder die Architekturen *SOA* oder *Client-Server* verwendet werden, dann sind Netzwerkprotokolle mit entsprechenden Abläufen und Nachrichtenformaten unverzichtbar und zum Quality Gate *Fertig für Implementierung und Test* vorzulegen.
- Wenn über ein Netzwerk kommuniziert werden muss oder die Architekturen *SOA* oder *Client-Server* verwendet werden, dann ist eine Beschreibung der Funktionsverteilung auf verschiedene Rechner oder Systemteile und eine Beschreibung der Kommunikation unter ihnen unverzichtbar und zum Quality Gate *Fertig für Implementierungsphase* vorzulegen.
- Wenn über ein Netzwerk kommuniziert werden muss oder die Architekturen *SOA* oder *Client-Server* verwendet werden, dann ist die Beschreibung der physischen Verteilung der einzelnen Softwarekomponenten unverzichtbar und zum Quality Gate *Fertig für Implementierung und Test* vorzulegen.
- Wenn das Programm funktional komplex oder sehr komplex ist und über eine grafische Schnittstelle verfügt, dann ist der Entwurf nach dem MVC-Entwurfsmuster aufzuteilen und zum Quality Gate *Fertig für Implementierung und Test* vorzulegen.
- Wenn das Programm einfache oder mittlere funktionale Komplexität hat und über eine grafische Schnittstelle verfügt, dann ist es empfehlenswert, den Entwurf nach dem MVC-Entwurfsmuster aufzuteilen und zum Quality Gate *Fertig für Implementierung und Test* vorzulegen.

Die aus den Erfahrungen resultierenden Zugehörigkeitsfunktionen sind im Anhang C zu finden. Insgesamt besteht das notwendige Fuzzy-Inferenz-System aus 38 Zugehörigkeitsfunktionen und 17 Regeln. Es sind mehr als die acht obigen Regeln notwendig, da jeweils auch mindestens eine „umgekehrte“ Regeln benötigt wird, um sicherzustellen, dass das entsprechende Kriterium im umgekehrten Fall ausgewählt bzw. abgewählt wird.

14 der 38 Zugehörigkeitsfunktionen beziehen sich auf Projektmerkmalswerte der Eingangsgrößen. Lediglich 4 der 14 Zugehörigkeitsfunktionen sind eigentliche Fuzzy-Funktionen. Dies bedeutet, dass im Moment noch wenig von den Möglichkeiten des Fuzzy-Konzeptes profitiert wird. Da sich das Tailoring-Verfahren im Moment noch im Aufbau befindet, werden höchstwahrscheinlich weitere eigentliche Fuzzy-Funktionen notwendig sein.

Hinsichtlich seiner Verbindlichkeit muss das realisierte Tailoring-Verfahren zunächst als unverbindlich ausgestaltet werden, da Regeln und Zugehörigkeitsfunktionen noch nicht optimiert und vollständig sind.

13.1.6 Erfahrungen mit NetQGate als Teil des Erfahrungskreislaufes

Seit der Einführung von NetQGate im Wintersemester 2006/2007 wurde die Möglichkeit, Kriterien mit einer Erfahrung zu versehen, vielfach genutzt. Diese Erfahrungen konnten durch das Gate-Management in dreierlei Weise für Verbesserungen des Quality-Gate-Referenzprozesses genutzt werden:

- Kriterien des Kriterienkatalogs wurden abgeändert, entfernt oder mit anderen Kriterien zusammengefasst. Zwei Kriterien wurden dann zusammengefasst, wenn sie subjektiv betrachtet eine hohe Ähnlichkeit aufwiesen.
- Angaben zur Operationalisierung wurden ergänzt, um die Anwendung eines Kriteriums zu erleichtern oder zu vereinheitlichen.
- Weitere Regeln konnten dem Regelwerk des Tailoring-Verfahrens hinzugefügt werden.

Im Rahmen der Verbesserung wurden nur solche Erfahrungen berücksichtigt, die einen nichtleeren Beobachtungsteil aufwiesen. Die Grundannahme war hierbei, dass nur solche Erfahrungen ernst gemeint waren. Tabelle 53 zeigt den quantitativen Einfluss der Erfahrungen aus den Jahren 2006/07 und 2007/08 auf die drei genannten Punkte.

	Geänderte, entfernte oder zusammengefasste Kriterien	Ergänzung von Angaben zur Operationalisierung	Hinzunahme von Tailoring-Regeln
Anzahl	11	7	3
Im Verhältnis zur Anzahl der anfangs 62 Kriterien in %	18%	11%	5%

Tabelle 53: Einfluss der Erfahrungen auf Kriterien, Operationalisierung und Regelwerk

Bei der Verwendung von NetQGate zur Aktivierung von Erfahrungen im Erfahrungskreislauf traten zwei Probleme auf:

- Viele Projektvertreter (realisiert durch Qualitätsbeauftragten) haben die Möglichkeit zur Hinterlegung von Erfahrungen mit der eigentlichen Bewertung der Kriterien hinsichtlich ihrer Erfüllung verwechselt. Infolgedessen haben sie ihr eigenes Projekt anhand der Kriterien bewertet. Hieraus kann der Schluss gezogen werden, dass das Erfahrungssystem von NetQGate weniger intuitiv zu bedienen ist als erwartet und daher eine spezielle Einweisung notwendig ist.
- Die Möglichkeit, eine ausführliche Beobachtung mit Kontext zu einer Erfahrung in NetQGate angeben zu können, wurde durch die Projektvertreter nur spärlich genutzt. Meistens wurden sehr kurze Sätze wie beispielsweise „wirkt doppelt“ oder „trifft nicht zu“ angegeben.

Hinsichtlich zukünftiger Softwareprojekte ist zu erwarten, dass das Erfahrungssystem weniger genutzt wird, da der Kriterienkatalog und das Regelwerk des Tailoring-Verfahrens weniger Defizite aufweisen und somit weniger ungeeignete Kriterien angewendet werden. Eine Validierung oder Falsifizierung dieser Hypothese kann jedoch erst in den folgenden Semestern erfolgen.

13.1.7 Untersuchung der Auswirkungen des Quality-Gate-Referenzprozesses

In diesem Abschnitt wird gezeigt, dass die Reduzierung der Sitzungsteilnehmer und der Einsatz von NetQGate keinen erkennbar negativen Einfluss auf den Projekterfolg genommen haben. Hierzu wird jeweils die Anzahl der erfolgreichen Projekte verschiedener Jahre miteinander verglichen. Wichtig zu erwähnen ist, dass die Kriterien über die Jahre hinweg in etwa gleich geblieben sind. Spätere Jahre weisen die gleichen Kriterien und wenige weitere Kriterien auf. Hiermit ist eine Vergleichbarkeit der jeweiligen Erfolgsquoten gewährleistet.

Tabelle 54 zeigt einen Überblick über die Anzahl der Projekterfolge in verschiedenen Jahren. Es zeigt sich, dass die Erfolgsquote – trotz Veränderungen in der Ausgestaltung im Quality-Gate-Referenzprozess – in etwa konstant bleibt. Schwankungen der Erfolgsquote sind, aufgrund der zu niedrigen Stichprobengröße, höchstwahrscheinlich zufällig bedingt.

Aspekt	Jahr 2004	Jahr 2006/07	Jahr 2007/08
Erfolgreich (Es wird dazu eingesetzt, wozu es gedacht war.)	5	6	5
Eher erfolgreich (Es könnte dazu eingesetzt werden, wozu es gedacht war.)	2	1	1
Eher nicht erfolgreich (Es ist einsetzbar, aber die Alternativen sind besser.)	0	0	1
Nicht erfolgreich (Es ist höchstens stark eingeschränkt einsetzbar.)	2	1	1
Erfolgsquote in %	77,8%	87,5%	75%

Tabelle 54: Überblick über Projekterfolge in verschiedenen Jahren

Eine vermutete negative Auswirkung der Etablierung der vollständigen Einsatzstrategie im Quality Gate *Fertig für Abnahmephase* zeigt sich im Wintersemester 2007/2008. Durch den Wegfall der Sitzung, wurde der Quality-Gate-Prozess vermutlich durch ein Projektteam nicht ernst genommen, so dass die Erfüllung der Kriterien insgesamt dreimal im Rahmen bedingter Projektfortsetzungen hinausgezögert wurde. Infolgedessen wurde dieses Projekt letztlich abgebrochen. Es lässt sich folgern, dass in solchen Fällen rechtzeitig eine *Wiederholung des Quality Gates* mit einer dann realen Sitzung stattfinden sollte.

13.1.8 Abschließende Betrachtung

Diese Fallstudie hat einen Quality-Gate-Referenzprozess gezeigt, der der grundlegenden Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* folgt. Durch die Nutzung von Bewertungsprofilen durch das Gate-Management konnten Defizite in der Ausgestaltung identifiziert und letztlich behoben werden. Dabei ist jedoch festzustellen, dass die derzeitige Ausgestaltung immer noch leichte Defizite aufweist.

Offensichtlich kann NetQGate in der vollständigen Einsatzstrategie genutzt werden. Bei wiederholter Nichterfüllung der Kriterien eines Quality Gates sollte jedoch die Entscheidung „Wiederholung des Quality Gates“ getroffen werden und eine reale Sitzung (gleicher Ort, gleiche Zeit) stattfinden.

Weiterhin zeigt sich, dass der Aufwand für ein Gate-Review durch die verstärkte Anwendung der vollständigen Einsatzstrategie geschätzt auf unter 60% gedrückt werden kann. Bei gleichzeitig erheblicher Reduzierung der Teilnehmeranzahl in der Sitzung eines Gate-Reviews, kann der Aufwand geschätzt auf unter 30% gedrückt werden. Erfreulicherweise wirken sich der Einsatz von NetQGate und die gleichzeitige Reduzierung der Sitzungsteilnehmer nicht erkennbar negativ auf die Erfolgsquote aus.

Obwohl sich das Erfahrungssystem in NetQGate als nicht intuitiv genug herausgestellt hat, konnten dennoch ausreichend viele verwertbare Erfahrungen gewonnen werden. Resultierend konnte eine ganze Reihe von Kriterien und Angaben zu ihrer Operationalisierung verbessert werden. Gleichzeitig konnten jedoch nur sehr wenige Regeln aus diesen Erfahrungen abgeleitet werden.

Für das Fuzzy-Inferenz-System konnten verschiedene Regeln aufgestellt werden. Da die Erstellung der Regeln der *Minimalen Anpassungsstrategie* folgt, sind im Moment nur wenige Regeln vorhanden. Ferner ist der Anteil eigentlicher Fuzzy-Funktionen bei den Projektmerkmalswerten der Eingangsgrößen im Moment noch gering.

13.2 Einsatz von Quality Gates in der Projektarbeit PWS

Im Fachgebiet Software Engineering der Gottfried Wilhelm Leibniz Universität Hannover wurde in der Projektarbeit „Entwicklung einer Webservice-basierten Anwendung“ (PWS) für Master-Studenten der Informatik im Wintersemester 2007/08 ein Quality-Gate-Prozess verwendet. Die Einführung von Quality Gates war notwendig, da bei der erstmaligen Durchführung der Projektarbeit im Wintersemester 2006/07 erhebliche qualitative Mängel bei der erstellten Software festgestellt wurden. Letztendlich erlangte die Software niemals ein einsatzfähiges Stadium. In dieser Fallstudie wird der verwendete Quality-Gate-Prozess vorgestellt. Der Quality-Gate-Prozess wurde zunächst nur für diese spezielle Projektsituation konzipiert. Es erscheint jedoch möglich, hieraus einen Quality-Gate-Referenzprozess zu konstruieren, der für zukünftige Projektarbeiten genutzt werden kann. Da jede zukünftige Projektarbeit PWS höchstwahrscheinlich auf Iterationsebene einen anderen Entwicklungsprozess aufweist, ist der Quality-Gate-Referenzprozess gemäß der grundlegenden Strategie *Quality Gates als flexible Qualitätsstrategie* auszugestalten.

Die Projektarbeit PWS hatte zwei Ziele:

1. Die Umsetzung einer Entwicklungsumgebung für Use Cases (UCDE) [Knauss '07] als serviceorientierte Architektur
2. und die Realisierung eines Change-Management-Systems für Use Cases.

Beide Ziele sollten jeweils durch ein Teilprojekt (TP A und TP B) umgesetzt werden und ihre Ergebnisse im weiteren Projektverlauf zusammengeführt werden. Der Projektarbeit wurde ein iterativer Entwicklungsprozess zugrunde gelegt, der dem Grundmuster des agilen Vorgehens gleicht. Auf Phasenebene existierte lediglich eine Release-Phase:



Abbildung 118: Aufbau des Entwicklungsprozesses auf Phasenebene

Schlüsselt man die Release-Phase auf, so ist die Unterteilung des Projektes in zwei Teilprojekte erkennbar, deren Entwicklungsprozess jeweils in drei Iterationen unterteilt war (vgl. Abbildung 119). Die Ergebnisse der Teilprojekte wurden im Rahmen einer Integration (Iteration 3) zusammengeführt.

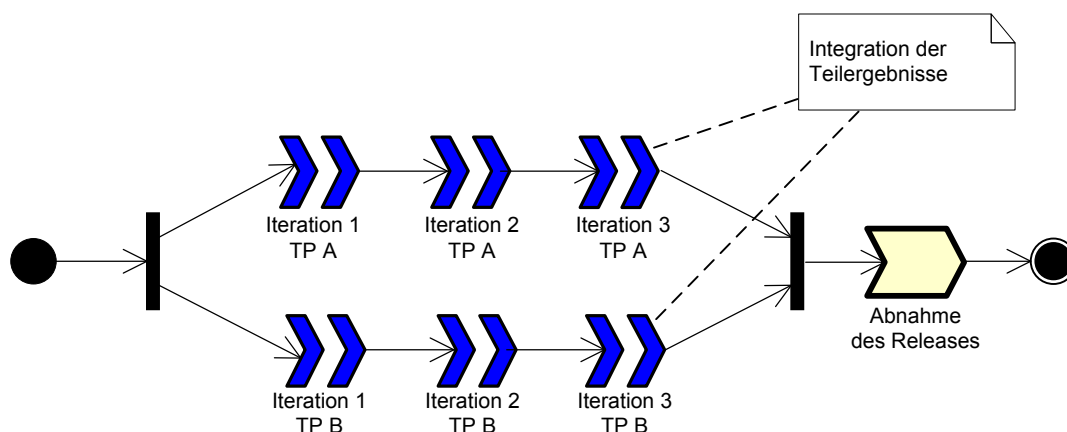


Abbildung 119: Aufbau der Release-Phase

Jede Iteration hatte eine Länge von vier Wochen und war auf Aktivitätsebene gleich strukturiert (vgl. Abbildung 120).

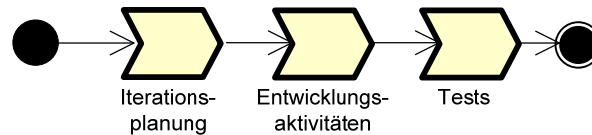


Abbildung 120: Aufbau einer Iteration

Für die Projektarbeit wurde eine fiktive Projekt- und Qualitätsmanagement-Hierarchie geschaffen. Jedes Teilprojekt besaß jeweils einen Projektleiter und einen Qualitätsbeauftragten, die durch verschiedene Personen besetzt werden mussten. Eine Hierarchieebene höher existierten jeweils ein Projektmanager und ein Qualitätsmanager. Eine explizite Geschäftsleitung war nicht vorhanden.

13.2.1 Ausgestaltung der strukturellen Konzepte

Aufgrund des iterativen Vorgehens und der Notwendigkeit, rechtzeitig und häufig steuernd in den Entwicklungsprozess eingreifen zu können, wurde ein Gate-Netzwerk konzipiert, das ein Quality Gate zwischen jeder Iteration vorsah (vgl. Abbildung 121). Um die Qualität der Integration der Ergebnisse zu prüfen, wurde ein weiteres Quality Gate vor der Abnahme des Releases platziert. Das resultierende Gate-Netzwerk orientiert sich hierbei an den Gate-Netzwerken bei parallel verlaufenden Handlungssträngen, da sich jedes Teilprojekt mit der Entwicklung von Software beschäftigt.

Insgesamt existierten drei verschiedene Quality Gates im Gate-Netzwerk, wobei das Quality Gate *Fertig für nächste Iteration* zweimal pro Handlungsstrang durchlaufen werden musste. Insgesamt musste das Gesamtprojekt damit fünf Quality Gates durchlaufen.

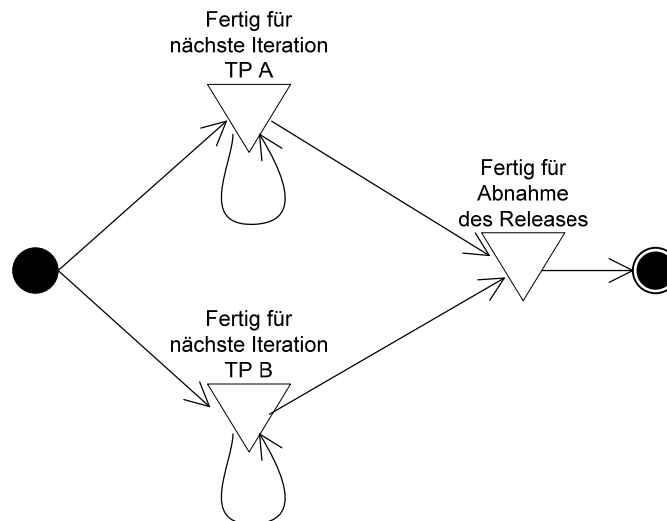


Abbildung 121: Verwendetes Gate-Netzwerk in der Projektarbeit PWS

13.2.2 Ausgestaltung der inhaltlichen Konzepte

Kriterien wurden über eine Mischung aus zielorientiertem und erfahrungsbasiertem Vorgehen erstellt, da zum einen bereits aus dem Vorjahr eine Vorstellung über die notwendigen Kriterien vorhanden war und zum anderen die Befürchtung bestand, dass die Abdeckung der Qualitätsziele durch die Kriterien nicht ausreichend hoch genug war. Die Kriterienerstellung wurde durch den Qualitätsmanager durchgeführt und die so identifizierten Kriterien mit dem Projektmanager abgestimmt.

Die Kriterien wurden zu Beginn der Durchführungsphase erfahrungsbasiert für alle fünf Quality Gates festgelegt und durch die GQM-Methode ergänzt. Abbildung 122 zeigt das erste Ergebnis des erfahrungsbasierten Vorgehens bei der Kriterienerstellung. Horizontal sind die drei Quality Gates erkennbar, die jedes Teilprojekt jeweils durchlaufen musste. Die erste Spalte definiert nach Ergebnissen geordnete Kennzahlen. Die drei weiteren Spalten definieren jeweils Sollwerte und Prädikate für jedes Quality Gate.

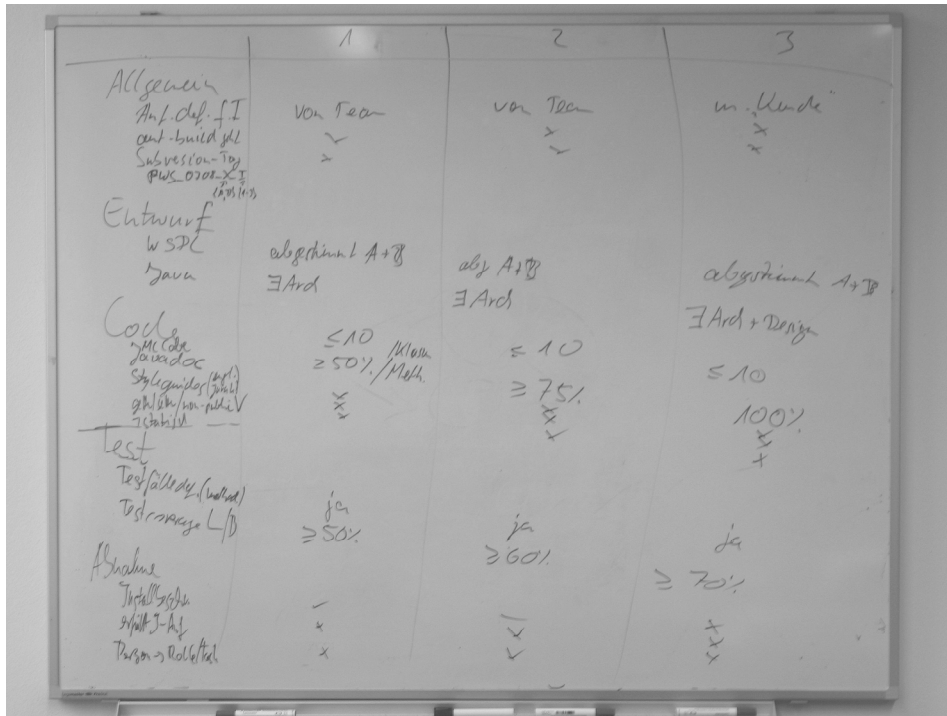


Abbildung 122: Erstes Ergebnis des erfahrungsbasierten Vorgehens bei der Kriterienerstellung

Die zu erreichenden Qualitätsziele betrafen die Korrektheit, die Test- und Wartbarkeit, die Verständlichkeit, die Strukturiertheit (insbesondere hinsichtlich der Objektorientierung) und die Zugänglichkeit (insbesondere hinsichtlich der Installierbarkeit). Der resultierende vollständige GQM-Baum mit Zielen, Teilzielen, Fragen und Maßen ist in Abbildung 123 zu sehen. Mittels der GQM-Methode konnten insgesamt fünf neue Maße identifiziert werden (fett umrandete Kästen in Abbildung 123). Für diese Maße wurden Sollwerte und Prädikate definiert, um die notwendigen Kriterien zu erhalten. Anhang D fasst alle erstellten Kriterien zusammen.

Aufgrund des iterativen Vorgehens wurden die gleichen Kriterien für jedes Quality Gate vorgegeben. Die Sollwerte einiger dieser Kriterien wurden in späteren Quality Gates jedoch verschärft. So war beispielsweise zunächst eine Testüberdeckung der Zeilen und Verzweigungen des Quellcodes von mindestens 50% gefordert. Dieser Sollwerte wurde in den zwei weiteren Quality Gates zunächst auf 60% und dann auf 70% angehoben.

Die Kriterien stellten Forderungen an verschiedene Ergebnisse. Diese Ergebnisse sind in der folgenden Tabelle zusammengefasst.

Geforderte Ergebnisse
Anforderungsspezifikation für Iteration
Ant-Build-File
Quellcode als Zip-Archiv
WSDL-Beschreibung
Entwurfsdokument
Richtlinien zur Bezeichnung
Dokument mit Testfällen

Tabelle 55: Durch Kriterien geforderte Ergebnisse

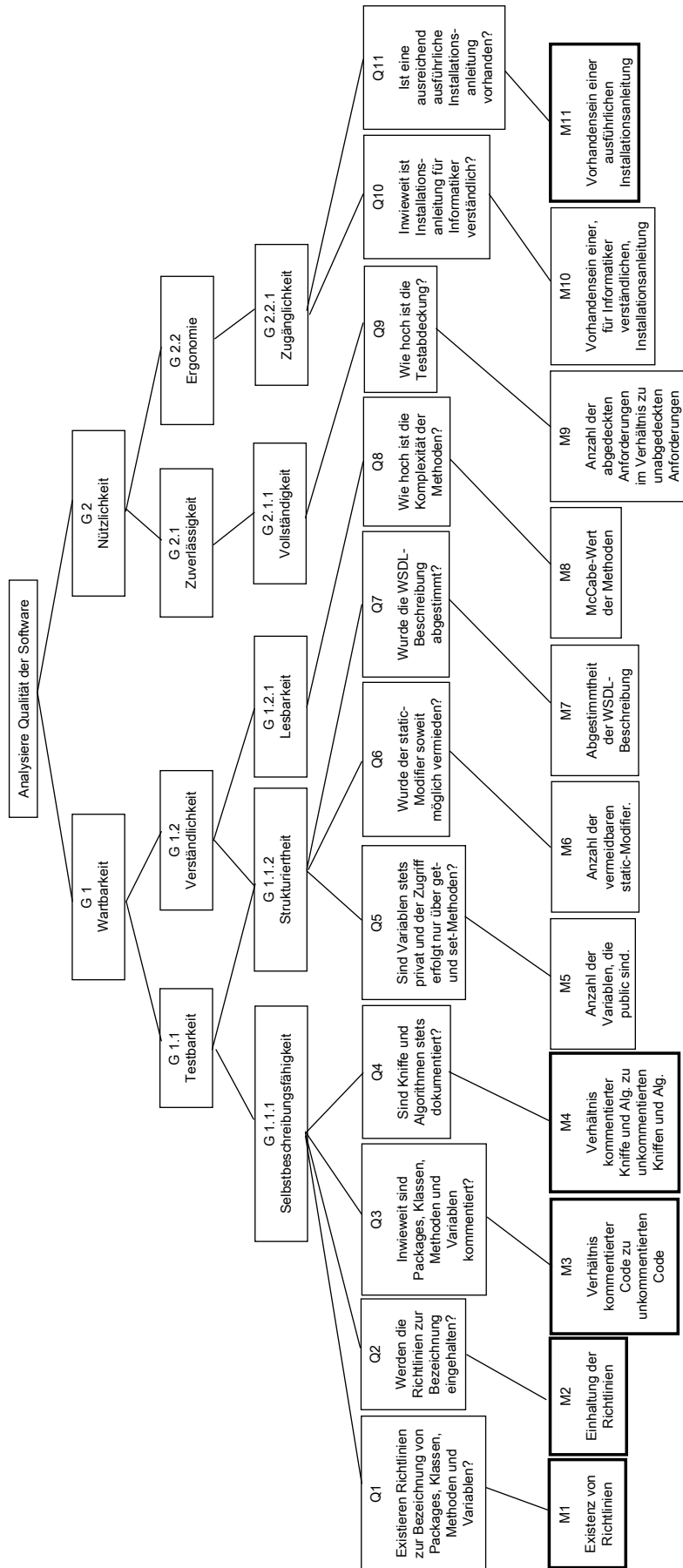


Abbildung 123: Vollständiger QQM-Baum zur Kriterienerstellung

Werden erfahrungsbasiertes und zielorientiertes Vorgehen bei der Kriterienerstellung kombiniert, so stellt sich die Frage nach dem Zugewinn an Kriterien durch das zielorientierte Vorgehen. Weiterhin ist von Interesse, ob ein zielorientiertes Vorgehen (neben dem möglichen Zugewinn) alle Kriterien liefert, die auch ein erfahrungsbasiertes Vorgehen liefern würde.

Die folgende Tabelle zeigt, wie viele der Kriterien jeweils nur durch das erfahrungsbasierte bzw. nur das zielorientierte Vorgehen identifiziert werden konnten. Weiterhin zeigt sie, wie viele Kriterien durch beide Vorgehensweisen identifiziert werden konnten.

	nur erfahrungsbasiert	zielorientiert und erfahrungsbasiert	nur zielorientiert
Anzahl der Kriterien	9 bereinigt 1	6	5
In % zur Gesamtzahl von 20 Kriterien	45%	30%	25%
In % zur bereinigten Gesamtzahl	8,33%	50%	41,67%

Tabelle 56: Anzahl identifizierter Kriterien durch verschiedene Vorgehensweisen

Es zeigt sich, dass 5 der 20 Kriterien (also 25%) durch die Verwendung der GQM-Methode ergänzt werden konnten. Gleichzeitig kann festgestellt werden, dass die GQM-Methode nicht alle Kriterien liefert, die die erfahrungsbasierte Vorgehensweise liefert. Diese Feststellung ist allerdings mit Vorsicht zu betrachten, da 8 dieser 9 Kriterien sich auf das Vorhandensein bestimmter Ergebnisse beziehen oder die Erleichterung der Messung betreffen. Diese Kriterien sind entweder trivial (z.B. wenn das Ergebnis bereits Messobjekt eines anderen Kriteriums ist und somit trivialerweise vorliegen muss) oder nicht qualitätsorientiert. Dementsprechend konnte nur ein Kriterium ausschließlich durch das erfahrungsbasierte Vorgehen identifiziert werden. Die dritte Zeile der Tabelle stellt die, auf diese Weise, „bereinigten“ Prozentwerte dar.

Die Differenz zum zielorientiertem Vorgehen lässt sich wie folgt erklären:

- Das fehlende Kriterium „Die Java-Code-Konventionen wurden eingehalten?“ wurde nicht durch die GQM-Methode identifiziert, da der zugehörige Qualitätsfaktor nicht mit einer negativen Ausgangshypothese im Abstraction Sheet verbunden ist: „Die Codekonventionen werden zumeist eingehalten“. Daher ist eine Prüfung im Quality Gate nicht notwendig und das Kriterium damit unnötig.
- Das zweite Kriterium „Die Architektur ist im Entwurfsdokument vollständig fixiert?“ wurde durch einen handwerklichen Fehler nicht identifiziert. Das Kriterium hätte über den Zielpfad Wartbarkeit → Verständlichkeit → Lesbarkeit identifiziert werden müssen.

13.2.3 Ausgestaltung der Reviewkonzepte

Das Gate-Review wurde in der Form einer formalen Inspektion ohne individuelle Vorbereitung durchgeführt (vgl. Abbildung 114, Abbildung 115 und Abbildung 116). Dabei wurde stets NetQGate in der hybriden Einsatzstrategie eingesetzt. Folglich fand zu allen fünf Quality Gates eine Sitzung statt. Die Sitzungslänge betrug jeweils etwa 30 Minuten.

Die folgende Tabelle zeigt die Ausgestaltung der einzelnen Rollen im Gate-Review.

Rolle im Gate-Review	Besetzt durch
Gate-Verantwortlicher	Qualitätsmanager
Gutachter	Qualitätsmanager
Projektvertreter	Qualitätsbeauftragter des Projektes
Protokollant	Qualitätsmanager
Gatekeeper (für nicht weitreichende Entscheidungen)	Qualitätsmanager und Projektmanager
Gatekeeper (für weitreichenden Entscheidungen)	Nur teilweise ausgestaltet

Tabelle 57: Ausgestaltung der Rollen im Gate-Review

Die Operationalisierung der Kriterien (insbesondere der Messung) fand im Gate-Review selbst statt. Dies stellte sich als Problem heraus, da die notwendigen Software-Messwerkzeuge zur Bestimmung der Istwerte nicht auf den benutzten Rechnern vorlagen und entsprechend nachinstalliert werden mussten. Hierfür wurde bereits ein Großteil der Sitzungszeit benötigt. Teilweise war ebenfalls unklar, wie die Messungen durchgeführt werden sollten. Idealerweise hätte zuvor ein Messplan aufgestellt werden müssen. Dieser hätte sich ebenfalls durch die GQM-Methode gewinnen lassen können.

Einige Messungen hätten aufgrund des notwendigen Aufwands im Vorfeld der Sitzung stattfinden müssen. Im Vorfeld hätte auch die notwendige (IT-)Infrastruktur aufgesetzt werden können, um die Messungen effizient durchführen zu können. Insbesondere Messungen am Quellcode gestalteten sich schwierig, wenn kein Softwarewerkzeug vorhanden war, das die Messung erleichterte. So ist es ohne Software-Unterstützung insbesondere schwierig festzustellen, wie gut der Quellcode dokumentiert ist und ob Konventionen der objektorientierten Programmierung eingehalten wurden.

Eine Ausgestaltung des Gate-Reviews als formale Inspektion mit individueller Vorbereitung und Sitzung wäre für die vorliegende Projektarbeit geeigneter gewesen. Die Sitzung (inkl. Anreisezeit) hätte sich damit gemäß Abschnitt 8.1.2 auf geschätzt 60% des zeitlichen Aufwands reduziert. Gleichzeitig hätte die Prüfung der Kriterien im Rahmen der individuellen Vorbereitung genauer ausfallen können.

13.2.4 Ausgestaltung der Steuerungskonzepte

Es waren drei Möglichkeiten zur Entscheidung vorgegeben: die uneingeschränkte Projektfortsetzung, die bedingte Projektfortsetzung und der Projektabbruch. Hierbei galt der Projektabbruch jedoch von Anfang an als Entscheidungsmöglichkeit, die aufgrund des Lehrveranstaltungscharakters nur in extremen Situationen getroffen werden konnte. Auf die Möglichkeit zur Wiederholung eines Quality Gates wurde verzichtet, da der zeitliche und organisatorische Aufwand hierfür zu hoch erschien. Stattdessen wurde die bedingte Projektfortsetzung gewählt, die die Möglichkeit einer nebenläufigen Begutachtung und Entscheidung bietet und gleichzeitig zeitlich und organisatorisch weniger aufwändig ist.

Des Weiteren wurde eine Entscheidungsunterstützung λ definiert, die davon ausging, dass jedes Kriterium gleich gewichtet (Strategie „Gleichgewichtung“) ist:

$$\lambda = (g(k), b) = \begin{cases} \text{go}, & \text{falls alle Kriterien erfüllt sind} \\ \text{cond.-go}, & \text{falls } \geq 1 \text{ Kriterium nicht erfüllt ist/sind.} \end{cases}$$

Die Menge der Empfehlungen wurde unmittelbar an die möglichen Entscheidungen geknüpft:

$$E = \{\text{go}, \text{cond.-go}\}$$

Der Projektabbruch wurde hingegen aus oben genanntem Grund nicht näher in der Entscheidungsunterstützung formalisiert.

13.2.5 Untersuchung der Auswirkungen des Quality-Gate-Referenzprozesses

Tabelle 58 zeigt die aufgetretenen Abweichungen der Istwerte zu den Sollwerten, die in den verschiedenen Quality Gates ermittelt wurden. Da stets Abweichungen auftraten, wurde das Projekt immer nur bedingt fortgesetzt werden.

Erfreulicherweise konnten durch die vorhandenen Quality Gates Mängel in den beiden Teilprojekten entdeckt werden, die wahrscheinlich erst später für das Qualitätsmanagement sichtbar gewesen wären. Die meisten Abweichungen traten bei projektgefährdenden Kriterien auf (fett hervorgehoben in Tabelle 58). Durch die Gatekeeper (für nicht weitreichende Entscheidungen) wurde daher stets im Rahmen der bedingten Projektfortsetzung die Erfüllung der Kriterien gefordert. In der nebenläufigen Prüfung konnte erfreulicherweise stets die Erfüllung dieser Kriterien festgestellt werden.

Teilprojekt	Fertig für Iteration 2	Fertig für Iteration 3	Fertig für Abnahme des Releases
TP A	<ul style="list-style-type: none"> - Die WSDL-Beschreibung wurde nur teilweise abgestimmt. - Das Programm erfüllt die Anforderungsspezifikation der Iteration nicht. 	<ul style="list-style-type: none"> - Es liegt keine getaggte Programmversion in Subversion vor. - < 75% der Klassen sind kommentiert. 	<ul style="list-style-type: none"> - Es liegt keine getaggte Programmversion in Subversion vor. - Ein McCabe-Wert ≤ 10 ist nicht überall erfüllt. - Es wurden nicht alle Anforderungen aus der Anforderungsspezifikation umgesetzt.
TP B	<ul style="list-style-type: none"> - Die Programmversion ist mit einem falschen Tag in Subversion versehen worden. - Die WSDL-Beschreibung wurde nur teilweise abgestimmt. - Das Programm erfüllt die Anforderungsspezifikation der Iteration nicht. 	<ul style="list-style-type: none"> - Es liegt keine getaggte Programmversion in Subversion vor. - Die Überdeckung der Zeilen und Verzweigungen ist < 60% (Konnte nicht geprüft werden, da Ant-Build-File fehlerhaft) 	<ul style="list-style-type: none"> - Es wurden nicht alle Anforderungen aus der Anforderungsspezifikation umgesetzt.

Tabelle 58: Abweichungen von Sollwerten in den einzelnen Quality Gates

13.2.6 Abschließende Betrachtung

In dieser Fallstudie wurde ein Quality-Gate-Prozess für eine sehr spezielle Projektsituation betrachtet. Jedoch besteht die Möglichkeit, den Quality-Gate-Prozess zukünftig zu einem Quality-Gate-Referenzprozess mit verschiedenen Gate-Netzwerken und einem festen Kriterienkatalog mit Auswahl und Ergänzung auszubauen. Dieser Quality-Gate-Referenzprozess kann dann für verschiedene zukünftige Projektsituationen der Lehrveranstaltung *Projektarbeit PWS* angepasst werden. Da ein Quality-Gate-Referenzprozess im Moment noch nicht vorliegt, konnte eine Ausgestaltung der Anpassungskonzepte innerhalb dieser Fallstudie nicht untersucht werden.

Da zukünftige Projektarbeiten hinsichtlich ihrer Projektsituation sehr ähnlich sein dürften, können die im Rahmen dieser Projektarbeit erstellten Kriterien wahrscheinlich weitergenutzt werden, auch wenn bei der Erstellung der Kriterien nicht explizit darauf hingearbeitet wurde. Die Weiternutzung dürfte allerdings nicht für die Gate-Netzwerke gelten, da die verwendeten Prozesse auf Iterationsebene höchstwahrscheinlich stets verschieden sind. Da Kriterien im Rahmen der bedingten Projektfortsetzung immer erfüllt wurden, ist die Menge der möglichen Entscheidungen bereits stabil ausgestaltet.

Die Fallstudie zeigt weiterhin, dass eine Kombination aus erfahrungsbasiertem und zielorientiertem Vorgehen bei der Kriterienerstellung nicht sinnvoll ist, da bereits 92% aller notwendigen Kriterien durch die GQM-Methode gewonnen werden konnten. Allerdings konnten 58% der Kriterien durch das erfahrungsbasierte Vorgehen identifiziert werden. Dieser Wert erscheint umso besser, wenn man berücksichtigt, dass das erfahrungsbasierte Vorgehen sehr viel weniger Aufwand erforderte. Leider ist festzustellen, dass das erfahrungsbasierte Vorgehen viele unnötige, aber vor allem triviale Kriterien liefert.

Des Weiteren ist festzustellen, dass Messungen gut vorbereitet werden müssen, um die Sitzung effizient durchführen zu können. Ist der Messvorgang noch nicht stabil und reif – wie in diesem Fall – so stellt ein Gate-Review mit individueller Vorbereitung eine bessere Ausgestaltung des Gate-Reviews dar.

13.3 Abbildung des V-Modell-XT-Tailoring-Verfahrens in Fuzzy-Logik

Das Tailoring-Verfahren des V-Modells XT lässt sich vollständig durch ein Fuzzy-Inferenz-System (vgl. Abschnitt 10.3.2) modellieren. Die für das Anpassen des V-Modells XT notwendigen sechs Projektmerkmale sind in Tabelle 20 dargestellt (mit * und ** gekennzeichnete Projektmerkmale). Die Projektmerkmale besitzen jeweils zwei bis sechs (relevante) Merkmalswerte.

Die Anzahl der Stellgrößen ist weitaus höher. Für das Quality-Gate-Teilkonzept *Gate-Netzwerk* existieren 10 Stellgrößen. Insgesamt existieren jedoch nur 10 gültige Varianten für das Gate-Netzwerk – sie stellen die zehn möglichen Gate-Netzwerke dar. Weiterhin sind die Quality Gates auswählbar. Hierfür sind 18 Stellgrößen notwendig, die sich ebenfalls zu 10 gültigen Varianten kombinieren lassen. Für das anpassbare Quality-Gate-Teilkonzept *Checkliste* existieren 54 Stellgrößen, die sich zu insgesamt 20 Varianten kombinieren lassen. Diese 54 Stellgrößen entsprechen Kriterien, die jeweils das Vorhandensein eines bestimmten Produktes zu einem bestimmten Quality Gate verlangen.

Da das Tailoring-Verfahren des V-Modells XT nur boolesche Regeln zur Anpassung benötigt, sind lediglich Singleton-Funktionen zum Aufbau des Fuzzy-Inferenz-Systems notwendig. Insgesamt sind für die Darstellung der Projektmerkmalswerte 19 Zugehörigkeitsfunktionen notwendig. Für die Darstellung der linguistischen Werte der Stellgrößen werden 164 Zugehörigkeitsfunktionen benötigt.

Wie bereits festgestellt, kann das Tailoring-Verfahren des V-Modells XT komplett mit Singleton-Funktionen realisiert werden – ihre Umsetzung gestaltet sich somit sehr einfach. Alle benötigten Zugehörigkeitsfunktionen werden im Anhang E vorgestellt.

Da es zu umfangreich wäre, alle Regeln in dieser Arbeit zu beschreiben, werden nachfolgend exemplarisch nur die beiden Regeln gezeigt, die relevant für die Auswahl der agilen Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer (Projektmerkmalswert β_1) sind:

1. „Agile Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer“ ja:

$$\underbrace{(\mu_{\alpha_{1,1}} \vee \mu_{\alpha_{1,2}} \vee \mu_{\alpha_{1,3}} \vee \mu_{\alpha_{1,4}})}_A \wedge \underbrace{(\mu_{\alpha_{2,3}} \vee \mu_{\alpha_{2,4}})}_B \wedge \underbrace{(\mu_{\alpha_{3,1}} \vee \mu_{\alpha_{3,3}})}_G \wedge \underbrace{\mu_{\alpha_{5,1}}}_\Delta \Rightarrow \mu_{\beta_{1,1}}$$

2. „Agile Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer“ nein:

$$\underbrace{(\mu_{\alpha_{2,1}} \vee \mu_{\alpha_{2,2}} \vee \mu_{\alpha_{2,5}} \vee \mu_{\alpha_{2,6}})}_B \wedge \underbrace{\mu_{\alpha_{3,2}}}_G \wedge \underbrace{\mu_{\alpha_{5,2}}}_\Delta \Rightarrow \mu_{\beta_{1,2}}$$

Die folgende Tabelle begründet die einzelnen Teilausdrücke in den Prämissen.

Projektmerkmal	Umgangssprachliche Beschreibung	Teilausdruck in Prämisse
Projektgegenstand	Die agile Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer kann unabhängig vom Projektgegenstand gewählt werden.	A
Projektrolle	Die agile Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer ist nur wählbar bei der Kombination Auftraggeber/Auftragnehmer.	B
Systemlebenszyklusausschnitt	Die agile Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer kann nur bei Entwicklung oder Weiterentwicklung und Migration gewählt werden.	Γ
Fertigprodukte	Es besteht keine Beziehung zur agilen Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer.	—
Hohe Realisierungsrisiken	Die agile Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer ist zu wählen, wenn die Realisierungsrisiken hoch sind.	Δ
Safety und Security	Es besteht keine Beziehung zur agilen Projektdurchführungsstrategie für Auftraggeber/Auftragnehmer.	—

Tabelle 59: Relevante Regeln für agile Projektdurchführungsstrategie

Basierend auf der beschriebenen Ausgestaltung des Fuzzy-Inferenz-System ermöglicht die folgende Filterfunktion die Identifikation der notwendigen auswählbaren Elemente:

$$\sigma : [0;1] \rightarrow \{\text{rot, grün}\}$$

$$\sigma(x) = \begin{cases} \text{rot,} & \text{für } 0 \leq x < 1 \\ \text{grün,} & \text{für } x = 1. \end{cases}$$

Abbildung 124 zeigt die Empfehlung des Quality-Gate-Tailoring-Assistent QGT für die obige Regel bei folgender Projektsituation (vgl. auch Abschnitt 4.5.3):

- **Projektgegenstand:** Softwaresystem
- **Projektrolle:** Auftraggeber und Auftragnehmer ohne Unterauftragnehmern
- **Systemlebenszyklusausschnitt:** Entwicklung
- **Fertigprodukte:** nein
- **Hohe Realisierungsrisiken:** ja

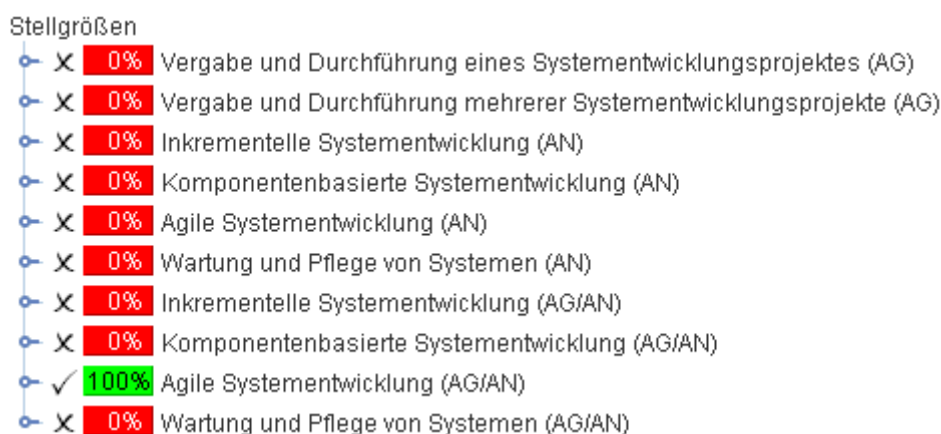


Abbildung 124: Empfehlung des Tailoring-Assistenten QGT

14 Zusammenfassung und Ausblick

Dieses Kapitel schließt die Arbeit mit einer Zusammenfassung und einem Ausblick ab. Ferner wird darauf eingegangen, wie die identifizierten Problempunkte durch das Framework adressiert werden.

14.1 Zusammenfassung

Zielsetzung dieser Arbeit war es, eine Anleitung zu entwickeln, mit der das Prozessmanagement eines Unternehmens Quality Gates etablieren kann. Diese Anleitung wurde als Framework aufgebaut, auf dessen Grundlage das Prozessmanagement mit vertretbarem Aufwand einen geeigneten Quality-Gate-Referenzprozess ausgestalten kann.

Diese Ausgestaltung wird – neben der spezifischen Ausgangssituation eines Unternehmens – im Wesentlichen durch die Wahl einer grundlegenden Strategie beeinflusst. Hierbei sind zwei Strategien möglich: *Quality Gates als einheitliche Qualitäts-Leitlinie* und *Quality Gates als flexible Qualitätsstrategie*. Erstere Strategie setzt darauf, Quality Gates möglichst einheitlich in allen Projekten einzusetzen, um Projekte im Sinne des Multiprojektmanagements miteinander vergleichen zu können. Die zweite Strategie setzt darauf, Quality Gates möglichst flexibel einsetzen zu können, um Qualitätsanforderungen prüfen zu können, die speziell an die Ergebnisse bestimmter Projekte gestellt werden. Die Verwendung von Quality Gates als Fortschrittsmaß ist in diesem Fall weniger wichtig.

Ein Unternehmen kann sich in drei Ausgangssituationen befinden:

- Es ist kein Quality-Gate-Referenzprozess vorhanden. Das Unternehmen muss eine grundlegende Strategie wählen (vgl. Abschnitt 12.1) und einen Quality-Gate-Referenzprozess von Grund auf ausgestalten.
- Es ist bereits ein Quality-Gate-Referenzprozess vorhanden, der jedoch problembehaftet ist. Ein Unternehmen in dieser Ausgangslage muss die Ausgestaltung seines Quality-Gate-Referenzprozesses bewerten und die Ausgestaltung entsprechend ändern (vgl. Abschnitt 12.2).
- Es ist bereits ein Quality-Gate-Referenzprozess vorhanden, der nur noch wenige Probleme verursacht. In diesem Fall kann die Effektivität und Effizienz durch den Einsatz von systematischen Erfahrungskreisläufen und Softwarewerkzeugen gesteigert werden (vgl. Abschnitt 12.3).

Um das Framework entwickeln zu können, wurden Quality-Gate-Referenzprozesse verschiedener Domänen betrachtet. Dabei konnten nur zum Teil Quality-Gate-Referenzprozesse in der Software-Entwicklung betrachtet werden, da sie gegenwärtig (noch) kein Forschungsgegenstand im Software Engineering sind. Gleichzeitig kann jedoch festgestellt werden, dass viele Unternehmen, die Software entwickeln, bereits Quality Gates nutzen. Aus diesem Grund wurde eine empirische Erhebung unter diesen Unternehmen durchgeführt. Durch die empirische Erhebung konnten drei wichtige Quality-Gate-Referenzprozesse identifiziert werden: der Quality-Gate-Referenzprozess nach Pfeifer, der Quality-Gate-Referenzprozess des V-Modells XT und der Quality-Gate-Referenzprozess des Stage-Gate-Prozesses. Durch die empirische Erhebung und die Untersuchung der drei wichtigen Quality-Gate-Referenzprozesse konnten verschiedene Problempunkte aufgedeckt werden, die bei der Verwendung von Quality Gates auftreten. Für diese Problempunkte bietet das Framework verschiedene Lösungsmöglichkeiten an (vgl. Abschnitt 14.2).

Basierend auf diesen Untersuchungen ist es möglich, ein Quality Gate eindeutig von den Konzepten des (projektbezogenen) Meilensteins und des technischen Reviews abgrenzen zu können. Dies ist wichtig, da häufig diese beiden Konzepte mit dem Konzept des Quality Gates gleichgesetzt werden. Erst durch diese klare Abgrenzung ist es dem Prozessmanagement eines Unternehmens möglich zu bestimmen, ob es bereits Quality Gates (eventuell unter einer anderen Bezeichnung) nutzt oder nicht.

Weiterhin wurde in dieser Arbeit ein abstrahierendes Domänenmodell erstellt, das die Teilkonzepte nennt, die mit Quality Gates in Beziehung stehen. Es schafft für Wissenschaftler und Praxisanwender einen Begriffsrahmen mit einer festen Terminologie für den wissenschaftlichen Austausch. Dem Prozessmanagement eines Unternehmens liefert das abstrahierende Domänenmodell einen Überblick über die Teilkonzepte, die ausgestaltet werden müssen, um einen vollständigen Quality-Gate-Referenzprozess zu erhalten.

Das abstrahierende Domänenmodell teilt die Teilkonzepte auf verschiedene, thematisch zusammenhängende Pakete auf. Diese Pakete beschäftigten sich mit strukturellen Konzepten, inhaltlichen Konzepten, Reviewkonzepten, Steuerungskonzepten und Anpassungskonzepten.

Die nachfolgenden Betrachtungen fassen kurz die Kernergebnisse zusammen, die für die Ausgestaltung der Teilkonzepte der einzelnen Pakete gelten.

- **Strukturelle Konzepte**

Innerhalb der Untersuchung der strukturellen Konzepte wurde das Teilkonzept *Gate-Netzwerk* als ausgestaltbares Teilkonzept identifiziert. Ein Gate-Netzwerk ermöglicht es, eine Anzahl, eine Auswahl und eine Anordnung von Quality Gates für einen Entwicklungsprozess festlegen zu können. Es existieren verschiedene Anforderungen an Gate-Netzwerke. Beispielsweise sollten Gate-Netzwerke für risikoreichere oder wichtigere Projekte auch eine höhere Anzahl an Quality Gates aufweisen. Es können Gate-Netzwerke für verschiedene Vorgehensweisen definiert werden. Dabei stellt es kein Problem dar, Gate-Netzwerke für iteratives, inkrementelles und agiles Vorgehen zu definieren und so Quality Gates auch bei nicht wasserfallartigem Vorgehen verwenden zu können. Um die Anzahl der Gate-Netzwerke möglichst gering zu erhalten, lassen sich bestimmte Gate-Netzwerke (sog. Basis-Gate-Netzwerke) durch Operationen manipulieren. Auf diese Weise erhält ein Prozess-Tailorer ein Gate-Netzwerk, das besser für ein bestimmtes Projekt geeignet ist als das ursprüngliche Basis-Gate-Netzwerk.

- **Inhaltliche Konzepte**

Durch eine Ausgestaltung der inhaltlichen Konzepte wird geklärt, wie, wann und durch wen was für Kriterien erstellt werden. Die Kriterienerstellung stellt das wichtigste Teilkonzept der inhaltlichen Konzepte dar. Die Kriterienerstellung kann hinsichtlich der Dimensionen Zeitpunkt der Erstellung, Individualität der Kriterien und Systematik der Erstellung unterschieden werden. Kriterien lassen sich entweder zum Projektstart, in der Planungsphase oder in der Durchführungsphase erstellen. Eine spätere Erstellung ermöglicht eine bessere Ausrichtung auf die Ergebnisse des Projektes, während eine frühere Erstellung mehr Eingriffsmöglichkeiten für das Management ermöglicht. Kriterien lassen sich in einem Kriterienkatalog fixieren, der es ermöglicht, die Kriterien in anderen Projekten wiederverwenden zu können. Die Ausgestaltung eines Kriterienkatalogs ist zwingend bei Befolgung der Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* erforderlich. Ferner ist es möglich, Kriterien individuell für jedes Projekt zu bestimmen. Die Operationalisierung, Ergänzung und Auswahl von Kriterien ermöglicht es jedoch auch bei festen Kriterienkatalog flexibler auf die Bedürfnisse eines Projektes eingehen zu können. Die Kriterienerstellung kann zielorientiert, erfahrungsbasiert oder intuitiv verlaufen.

Da Quality Gates qualitätsfokussierte Entscheidungspunkte sind, beziehen sich die erstellten Kriterien primär auf Qualitäts-Anforderungen, die an die Ergebnisse eines Projektes gestellt werden. Andere Kriterien sind möglich, können jedoch ein Quality Gate zum unfokussierten Entscheidungspunkt ausarten lassen.

Ein weiterer wichtiger Punkt ist die Gewichtung der Kriterien. Durch eine Gewichtung der Kriterien kann festgelegt werden, welche Kriterien den Kriterienerstellern besonders wichtig sind. Diese Gewichtung kann und sollte idealerweise für eine spätere Entscheidung herangezogen werden.

- **Reviewkonzepte**

Eine Orientierung des Gate-Reviews an den Varianten von technischen Reviews ist möglich. Sie lassen sich sogar nahezu identisch wiederverwenden. Für jedes Projekt ist zu klären, wie

aufwändig die notwendigen Gate-Review zu gestalten sind. Dazu muss festgelegt werden, ob eine Sitzung oder eine individuelle Vorbereitung jeweils notwendig sind. Ebenso ist zu klären, ob Gatekeeper mit weitreichender Entscheidungsbefugnis an der Sitzung teilnehmen sollten. Hinsichtlich des zeitlichen Aufwands wurde eine Schätzung der einzelnen Varianten des Gate-Reviews vorgenommen. Dabei zeigt sich, dass die weniger aufwändigen Varianten deutlich weniger Zeit zur Durchführung benötigen.

Im Rahmen der Untersuchung der Reviewkonzepte konnte (neben der Untersuchung der verschiedenen notwendigen Rollen) auch der Inhalt des Protokolls erarbeitet werden, das in einem Gate-Review entsteht.

- **Steuerungskonzepte**

Es lassen sich generell fünf verschiedene Entscheidungen unterscheiden, die in einem Quality Gate getroffen werden können. Es ist jeweils festzulegen, welche Rolle welche Entscheidungen treffen dürfen. Insbesondere ist festzulegen, wer weitreichende Entscheidungen (wie den Abbruch) treffen darf.

Ein wichtiges Teilkonzept ist die Entscheidungsunterstützung. Eine Entscheidungsunterstützung liefert auf Grundlage der Gewichtung und dem Erfüllungsgrad der Kriterien eine Empfehlung über die Fortsetzung eines Projektes. Eine Entscheidungsunterstützung ermöglicht es, falls sie verbindlich ausgestaltet und formalisiert wurde, Entscheidungen projektübergreifend vergleichbar treffen zu können. Eine Entscheidungsunterstützung ist daher sorgfältig zu konzipieren, um Fehlentscheidungen zu vermeiden.

- **Anpassungskonzepte**

Hierbei wurde untersucht, wie ein Unternehmen das Teilkonzept Gate-Management ausgestalten kann. Dabei wurde aufgezeigt, welche Ausgestaltungsmöglichkeiten Unternehmen haben, die nur wenige Ressourcen hierfür aufwenden können oder wollen. Hinsichtlich der Ausgestaltung kann festgestellt werden, dass das Vorhandensein eines Gate-Managements wichtiger ist, als dessen aufwändige Ausgestaltung.

Das wichtigste Teilkonzept der Anpassungskonzepte ist das Tailoring-Verfahren. Ein ausgestaltetes Tailoring-Verfahren erleichtert es einem Prozess-Tailorer, den optimalen Quality-Gate-Prozess für eine Projektsituation bestimmen zu können. Ein Tailoring-Verfahren kann verbindlich und formalisiert ausgestaltet sein. Eine informelle Ausgestaltung ist ebenfalls möglich. Die Realisierung eines Tailoring-Verfahrens auf Basis der Fuzzy-Logik stellt eine gute Möglichkeit dar, die Ergebnisse des Tailoring-Verfahrens nachvollziehbar zu gestalten.

Die Realisierung auf Basis der Fuzzy-Logik stellt verschiedene Herausforderungen an das Gate-Management. So ist insbesondere eine Vielzahl so genannter Zugehörigkeitsfunktionen zu erstellen. Das Gate-Management kann durch verschiedene Strategien hierauf antworten. Diese Strategien unterscheiden sich durch ihren anfänglichen Aufwand und in der anfänglich erzielten Reife des Tailoring-Verfahrens.

Die Formalisierung eines Tailoring-Verfahrens erfordert es, die auswählbaren Elemente eines Quality-Gate-Referenzprozesses festzulegen. Welche Elemente auswählbar sind, hängt im Wesentlichen von der Auswahl der Strategie ab. Die Strategie *Quality Gates als einheitliche Qualitäts-Leitlinie* lässt hierbei weniger veränderliche Elemente zu. Insbesondere ist das Teilkonzept *Gate-Netzwerk* in diesem Fall nicht anpassbar. Weiterhin wurde untersucht, welche Projektmerkmale ein Projektmodell enthalten kann. Hierbei wurde auch untersucht, inwieweit Projektmerkmale und auswählbare Elemente quantitativ und qualitativ bei der Anpassung miteinander in Beziehung stehen.

Im weiteren Verlauf der Arbeit wurde untersucht, wie ein Unternehmen seinen bereits bestehenden Quality-Gate-Referenzprozess verbessern kann. Hierbei erweist sich vor allem der Einsatz von Erfahrungskreisläufen und Softwarewerkzeugen als sinnvoll. Softwarewerkzeuge können in verschiedener Weise unterstützend wirken. Insbesondere wurde die Unterstützung des Gate-Reviews durch das Softwarewerkzeug *NetQGate* untersucht. Hierbei erweist sich die Unterstützung als sinnvoll. Gleichzeitig gestaltet sich die Unterstützung verschiedener möglicher Abläufe eines Gate-Reviews als Her-

ausforderung. Darüber hinaus kann auch die Unterstützung der Anpassung und der Operationalisierung sinnvoll unterstützt werden.

Nutzen und Mächtigkeit des Frameworks wurden in drei verschiedenen Fallstudien untersucht. Zwei von ihnen betrachteten den Einsatz von Quality Gates in studentischen Projekten im universitären Umfeld. In der letzten Fallstudie wurde das Tailoring-Verfahren des V-Modells XT mittels Fuzzy-Logik umgesetzt.

In der ersten Fallstudie wurde ein bereits bestehender Quality-Gate-Referenzprozess bewertet und über einen längeren Zeitraum verbessert. Hierdurch konnten verschiedene Defizite aufgedeckt und ausgeräumt werden. Ebenfalls wurde NetQGate in verschiedener Weise zur Unterstützung des Gate-Reviews eingesetzt. Dabei zeigt sich, dass sich allein durch den Einsatz des Softwarewerkzeuges NetQGate der zeitliche Aufwand geschätzt auf unter 60% drücken lässt. Eine gleichzeitige Reduzierung der Personenanzahl im Gate-Review ermöglicht eine geschätzte Reduktion des zeitlichen Aufwands auf unter 30%. Weiterhin konnte damit begonnen werden, ein auf Fuzzy-Logik basierendes Tailoring-Verfahren aufzusetzen. Dieses Tailoring-Verfahren befindet sich im Moment noch im Aufbau. Dementsprechend ist der Anteil von Regeln, die von der Fuzzy-Logik Gebrauch machen, noch gering.

Innerhalb der zweiten Studie wurde kein anpassbarer Quality-Gate-Referenzprozess untersucht, sondern ein Quality-Gate-Prozess, der für eine ganz spezielle Projektsituation konstruiert. Mit gewisser Wahrscheinlichkeit kann hieraus jedoch ein Quality-Gate-Referenzprozess entwickelt werden. In dieser Fallstudie konnte festgestellt werden, dass durch den Einsatz der GQM-Methode 92% der notwendigen Kriterien erstellt werden konnten. Durch ein erfahrungsbasiertes Vorgehen war es lediglich möglich, rund 58% der notwendigen Kriterien zu erstellen.

Durch die dritte Fallstudie konnte gezeigt werden, dass es mühelos möglich ist, das Tailoring-Verfahren des V-Modells XT mit Hilfe der Fuzzy-Logik zu realisieren. Hierbei kommen jedoch ausschließlich uneigentliche Fuzzy-Funktionen zum Einsatz, da das Tailoring-Verfahren des V-Modells XT auf boolescher Logik beruht.

14.2 Behandlung der Problempunkte durch das Framework

In diesem Abschnitt wird aufgezeigt, wie die in dieser Arbeit identifizierten Problempunkte mittels des Frameworks aufgelöst werden können. Die folgenden vier Tabellen zeigen jeweils, wie die identifizierten Problempunkte aufgelöst werden können. Die vier Tabellen sind hierbei geordnet nach den verschiedenen Problempunkten:

- Problempunkte bei Kriterien und Ergebnissen,
- Problempunkte beim Gate-Review,
- Problempunkte bei der Steuerung
- und Problempunkte im Zusammenspiel mit verschiedenen Entwicklungsprozessen.

Problempunkt	Durch das Framework angebotene Lösung
Projektunabhängig definierte Kriterien sind abstrakt. Die notwendige Operationalisierung vorläuft häufig subjektiv und unsystematisch.	Operationalisierung durch GQM-Methode und Qualitätsmodelle. Subjektivität als Stärke nutzen.
Formale Kriterien für alle zu prüfenden Aspekte zu finden ist schwierig.	Kriterien abstrakter definieren. Kriterien für jedes Projekt operationalisieren. GQM-Methode und Qualitätsmodelle zur Kriterienerstellung nutzen.
Die Lieferung von Kennzahlwerten in der richtigen Detaillierungsstufe und Vollständigkeit ist schwierig.	Kriterienerstellung systematisch zwischen Lieferanten und (internen) Kunden durchführen.
Abläufe, die der Erstellung von Kriterien dienen, fehlen meistens.	Kriterienerstellung zielorientiert oder erfahrungsbasiert durchführen.

Tabelle 60: Problempunkte bei Kriterien und Ergebnissen und angebotene Lösungen

Problempunkt	Durch das Framework angebotene Lösung
Die Prüfungsaktivitäten im Gate-Review werden häufig nicht systematisch durchgeführt.	Orientierung an formaler Inspektion oder Walkthrough.
Die unterschiedliche Durchführung von Gate-Reviews führt zu unterschiedlichen Entscheidungen bei vergleichbarer Ausgangslage.	Eine Variante des Gate-Reviews verbindlich für alle Projekte festlegen. Notwendige Review-Rollen fest auf Rollen im Unternehmen abbilden. Entscheidungsunterstützung verbindlich gestalten und formalisieren.
Beim Erreichen eines Quality Gates wird die Entwicklung häufig angehalten.	Conditional-go als Umsetzung des Fuzzy-Konzeptes ermöglichen.

Tabelle 61: Problempunkte beim Gate-Review und angebotene Lösungen

Problempunkt	Durch das Framework angebotene Lösung
Projekte werden häufig in den späten Phasen der Entwicklung nicht mehr abgebrochen.	Entscheidungsunterstützung verbindlich gestalten und formalisieren. Kriterien gewichten. Gewichtung ggf. im Kriterienkatalog fixieren. Kriterien späterer Quality Gates stärker gewichten.
Es ist möglich, dass ein Projekt zu früh abgebrochen wird, obwohl es vielversprechend ist.	Entscheidungsunterstützung verbindlich gestalten und formalisieren. Kriterien gewichten. Gewichtung ggf. im Kriterienkatalog fixieren. Kriterien früherer Quality Gates schwächer gewichten.
Es ist nicht immer offensichtlich, wer über den weiteren Verlauf eines Projektes entscheiden darf	Abbildung Gatekeeper auf Rollen im Unternehmen fixieren.
Abläufe, die der Entscheidungsfindung und zur Herleitung von notwendigen Maßnahmen dienen, fehlen häufig oder sind nur abstrakt beschrieben.	Entscheidungsunterstützung verbindlich gestalten und formalisieren. Kriterien gewichten.

Tabelle 62: Problempunkte bei der Steuerung und angebotene Lösungen

Problempunkt	Durch das Framework angebotene Lösung
Quality Gates werden bei inkrementellem, iterativem und agilem Vorgehen weniger häufig eingesetzt.	Strukturelle Konzepte bieten Gate-Netzwerke für unterschiedliches Vorgehen an.
Die Synchronisation von projektspezifischen Meilensteinen und Quality Gates ist problembehaftet.	Ergebnisse und Kriterien in der Planungs- oder Durchführungsphase festlegen. Quality Gates mit bestimmten projektspezifischen Meilensteinen zusammenlegen.
Ein formalisiertes Tailoring-Verfahren ist selten vorhanden. Quality Gates werden als Konsequenz uneinheitlich eingesetzt	Tailoring-Verfahren als Fuzzy-Inferenz-System ausgestalten. Ergebnis des Tailoring-Verfahrens ist verbindlich.

Tabelle 63: Problempunkte im Zusammenspiel mit verschiedenen Entwicklungsprozessen und angebotene Lösungen

14.3 Ausblick

Innerhalb dieser Arbeit wurden Quality-Gate-Referenzprozesse tief untersucht. Hieraus konnte ein Framework entwickelt werden, auf dessen Grundlage das Prozessmanagement eines Unternehmens einen geeigneten Quality-Gate-Referenzprozess erstellen kann. Verschiedene wichtige Forschungsfragen bleiben dennoch offen. Sie können jedoch wahrscheinlich durch weitere Untersuchungen geklärt werden. Die Forschungsfragen betreffen die folgenden Punkte:

- **Aufwandsschätzungen:** Der zeitliche Aufwand verschiedener Varianten des Gate-Reviews wurde lediglich aus Erfahrung geschätzt. Die Angaben zum Aufwand sind daher nicht statistisch valide, stellen jedoch eine gute Ausgangsbasis dar und können daher als Hypothesen für weitere empirische Untersuchungen in der freien Wirtschaft genutzt werden.

Für die Kriterienerstellung wurden keinerlei Untersuchungen hinsichtlich des zeitlichen Aufwands durchgeführt. Folglich besteht hier Bedarf für weitere empirische Untersuchungen.
- **Identifikation und Untersuchung weiterer Strategien:** In dieser Arbeit wurden zwei grundlegende Strategien betrachtet: *Quality Gates als einheitliche Qualitäts-Leitlinie* und *Quality Gates als flexible Qualitätsstrategie*. Denkbar sind Zwischenstufen dieser beiden Strategien, die es zulassen, nur bestimmte Untermengen der Menge aller Quality-Gate-Teilkonzepte anpassbar zu gestalten. Ebenfalls wurden Quality Gates nicht als ereignisorientiertes Konzept untersucht. Hierbei werden Quality Gates nicht an Prozessschritte gebunden, sondern müssen in periodisch gleichen zeitlichen Abständen passiert werden. Diese Herangehensweise ermöglicht wahrscheinlich weitere grundlegende Strategien.
- **Definition und Bewertung sekundärer Kriterien:** Innerhalb dieser Arbeit wurde nur untersucht, wie qualitätsorientierte Kriterien erstellt werden können. Die Erstellung anderer – so genannter sekundärer Kriterien – wurde nicht untersucht. Hierzu sind weitere Untersuchungen notwendig. Hinsichtlich der Bewertung können Verfahren eingesetzt werden, die ein Projekt aus verschiedenen Perspektiven betrachten. Ein solches Verfahren stellt beispielsweise das Balanced-Scorecard-Konzept dar. Hawlitzky [Hawlitzky '02] gibt einen Überblick über die Nutzbarkeit des Balanced-Scorecard-Konzepts und weiterer Verfahren zur Bewertung von Quality-Gate-Kriterien.
- **Identifikation von Leistungsmaßen für Gate-Reviews:** Es wurde nicht untersucht, durch welche Maße die Leistung von Gate-Reviews bewertet werden. Eine Orientierung an den Leistungsmaßen für technische Reviews erscheint sinnvoll, ist jedoch wahrscheinlich nicht gänzlich zulässig. Geeignete Leistungsmaße sind daher durch weitere Untersuchungen zu identifizieren.
- **Erarbeitung eines flexiblen Workflows für Gate-Reviews:** Das vorgestellte Softwarewerkzeug NetQGate zur Unterstützung von Gate-Reviews bildet im Moment einen recht starren

Workflow ab. Der Einsatz bei anders ausgestalteten Varianten des Gate-Reviews ist daher schwierig. Ebenso ist es nach Hedberg [Hedberg '04] schwierig, eine generische Lösung zu konstruieren. Daher ist mit gewisser Wahrscheinlichkeit jeweils eine eigenständige Lösung zu konstruieren. Weitere Untersuchungen sind notwendig, um dies zu bestätigen.

- **Weitere Maßnahmen zur Verbesserung des Erfahrungskreislaufs:** Innerhalb dieser Arbeit wurde einseitig die Verbesserung von Kriterien durch softwareunterstützte Erfahrungskreisläufe betrachtet. Dies ist ausreichend, da optimale Kriterien entscheidend für effektive Quality Gates sind. Dennoch sind weitere Untersuchungen notwendig, in denen die systematische erfahrungsbasierte Verbesserung anderer Teilkonzepte betrachtet wird.
- **Verknüpfung mit konstruktiven Maßnahmen der Qualitätssicherung:** Für Entwickler besteht die Möglichkeit, Softwarewerkzeuge einzusetzen, die die Einhaltung von Kriterien bereits bei der Erstellung der Produkte überwachen und bei Abweichung warnen. Jedoch lassen sich nicht alle Kriterien durch Softwarewerkzeuge überwachen. Eine Möglichkeit besteht darin, Kriterien, die durch Softwarewerkzeuge überwacht werden können, nicht mehr in einem Quality Gate zu prüfen. Alle anderen Kriterien müssen hingegen in einem Quality Gate geprüft werden. Auf diese Weise kann der Aufwand für die Prüfungsaktivitäten im Gate-Review reduziert werden. Ein erster Schritt zur Untersuchung einer derartigen Aufteilung wurde für die Anforderungsanalyse bereits getan [Knauss et al. '07].

Viele der Ergebnisse dieser Dissertation können als Ausgangspunkte oder als Hypothesen für weitere Untersuchungen verwendet werden. Insbesondere sind die zeitlichen Aufwandsschätzungen für die Varianten des Gate-Reviews und für die verschiedenen Systematiken der Kriterienerstellung weiter empirisch zu untersuchen. Ferner sind weitere grundlegender Strategien zu identifizieren. Gerade diese Ergebnisse dürften für Software-Unternehmen hochinteressant sein.

Literaturverzeichnis

[Assmann et al. '02]

D. Assmann, R. Kalmar und T. Punter. *Messen und Bewerten von Webapplikationen mit der Goal/Question/Metric Methode*. Fraunhofer Institut für Experimentelles Software Engineering (IESE), 2002.

Link: <http://publica.fraunhofer.de/eprints/urn:nbn:de:0011-n-176425.pdf>

Letzter Zugriff: 17. März 2008

[Ballman et al. '94]

K. Ballman und L. G. Votta. *Organizational congestion in large-scale software development*. In: Third International Conference on the Software Process. Reston, USA. IEEE CS Press, S. 124-134, 1994.

[Basili et al. '94a]

V. Basili, G. Caldiera und H. D. Rombach. *Experience factory*. In: *Encyclopedia of Software Engineering*. John Wiley & Sons, Auflage: 469-476, 1994a.

[Basili '93]

V. R. Basili. *The Maturing of the Quality Improvement Paradigm in the SEL and Experience Factory Fundamentals*. In: Eighteenth Software Engineering Workshop (SEL). NASA/Goddard Space Flight Center, Maryland, USA. 1993.

[Basili et al. '94b]

V. R. Basili, G. Caldiera und D. H. Rombach. *Goal question metric paradigm*. In: *Encyclopedia of Software Engineering*. John Wiley & Sons, Auflage: 528 - 532, 1994b.

[Basili et al. '96]

V. R. Basili, G. Caldiera, F. Lanubile und F. Shull. *Studies on reading techniques*. In: 21. Annual Software Engineering Workshop. Goddard Space Flight Center, Maryland, USA. S. 59-65, 1996.

[Beck '00]

K. Beck. *Extreme Programming Explained*. Addison-Wesley, 2000.

[Behl '00]

S. Behl. *Software Risk Management*. Universität Stuttgart, 2000.

Link: <http://www.iste.uni-stuttgart.de/ps/Lehre/stupro0001/Seminar/Stefan.Behl.ps.gz>

Letzter Zugriff: 17. März 2008

[Bernards '05]

M. Bernards. *Modulare Prüfplanung*. Doktorarbeit. Fakultät für Maschinenwesen, RWTH Aachen, 2005.

[Boehm et al. '78]

B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. J. MacLeod und M. J. Merrit. *Characteristics of software quality*. North-Holland Publishing Co., 1978.

[Boehm '88]

B. W. Boehm. *A Spiral Model of Software Development and Enhancement*. In: IEEE Computer. Band 21(5): S. 61-72, 1988.

[Brooks '95]

F. P. Brooks. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, Auflage: Anniversary Edition, 1995.

[CGISS '02]

CGISS 2002 Malcolm Baldrige Application Summary. Motorola Inc. Commercial, Government and Industrial Solutions Sector (CGISS), 2002.

Link: http://www.quality.nist.gov/PDF_files/MotorolaCGISS_Application_Summary.pdf

Letzter Zugriff: 11. April 2007

[CHAOSReport '94]

The CHAOS Report 1994. The Standish Group, 1994.

[CHAOSReport '06]

The CHAOS Report 2006. The Standish Group, 2006.

[Charvat '03]

J. P. Charvat. *How to use quality gates to guide IT projects*. BUILDER.COM - Beyond the Code, 2003.

Link: http://builder.com.com/5100-6315_14-1061893.html

Letzter Zugriff: 28. März 2007

[Ciolkowski et al. '03]

M. Ciolkowski, O. Laitenberger und S. Biffel. *Software Reviews: The State of the Practice*. In: IEEE Software. Band 20(6): S. 46-51, 2003.

[CMMI '06]

CMMI for Development Version 1.2. Carnegie Mellon Software Engineering Institute (SEI), 2006.

Link: http://www.sei.cmu.edu/publications/documents/06_reports/06tr008.html

Letzter Zugriff: 17. August 2007

[Cooper '98a]

R. G. Cooper. *Third-generation new-product processes*. In: Journal of Product Innovation Management. Band 11(1): S. 3-14, 1998a.

[Cooper '98b]

R. G. Cooper. *Product Leadership: Creating and Launching Superior New Product*. Perseus Books, 1998b.

[Cooper '01]

R. G. Cooper. *Winning At New Products: Accelerating the Process from Idea to Launch*. Perseus Books Group, 2001.

[Cornish '05]

P. Cornish. *Optimizing Cost, Risk & Time-to-Market with a Gate Review Process*. PD-TRAK: NPD Project & Portfolio Management System, 2005.

Link: <http://www.pd-trak.com/gatereview.htm>

Letzter Zugriff: 12. Januar 2007

[Crosby '79]

P. B. Crosby. *Quality is Free*. McGraw-Hill, 1979.

[DeMarco '82]

T. DeMarco. *Controlling Software Projects*. Yourdon Press, 1982.

[Dunham '99]

D. J. Dunham. *Gate Meetings: Speed Bump versus Acceleration Ramp*. Product Development and Management Association (PDMA), 1999.

Link: <http://www.pdma.org/visions/jul99/dunham.html>

Letzter Zugriff: 20. Dezember 2007

- [Ebert '06]
C. Ebert. *Understanding the Product Life Cycle: Four Key Requirements Engineering Techniques*. In: IEEE Software. Band 23(3): S. 19-25, 2006.
- [Fagan '76]
M. E. Fagan. *Design and Code Inspections to Reduce Errors in Program Development*. In: IBM Systems Journal. Band 15(3): S. 182-211, 1976.
- [Fährnich '04]
K.-P. Fährnich. *Software Management*. Vorlesung. Institut für Informatik, Anwendungsspezifische Informationssysteme, Universität Leipzig, 2004.
- [Fauth et al. '99]
G. Fauth, W. Winkelbauer, T. Pfeifer und T. Prefi. *Den Anlauf im Griff - Quality Gates in der Produktion sichern Marktqualität*. In: QZ Qualität und Zuverlässigkeit. Band 1999(6): S. 756-760, 1999.
- [Fischer et al. '98]
W. Fischer, J. Geschke, T. Kordes und H. Multhaupt. *Projektziele setzen - und erreichen: Das Projektmanagement während der Entwicklung*. In: ATZ Automobiltechnische Zeitschrift. Band 100(5): S. 164-171, 1998.
- [Flohr '06]
T. Flohr. *NetQGate - Tool Support for Quality Gate Processes*. In: 9th International Conference on Quality Engineering in Software Technology (CONQUEST). Berlin. dpunkt.verlag, S. 261-273, 2006.
- [Frühauf et al. '04]
K. Frühauf, J. Ludewig und H. Sandmayr. *Software Prüfung. Eine Fibel*. Verlag der Fachvereine (vdf), Auflage: 5, 2004.
- [Gilb et al. '93]
T. Gilb, D. Graham und S. Finzi. *Software Inspection*. Addison-Wesley, 1993.
- [Glinz '05]
M. Glinz. *Software Engineering: eine Einführung*. Vorlesung. Institut für Informatik, 2005.
- [Greive '05]
J. Greive. *Entwurf und Implementierung eines Systems zur netzbasierten Durchführung von Quality-Gates*. Bachelorarbeit. Fachgebiet Software Engineering, Gottfried Wilhelm Leibniz Universität Hannover, 2005.
- [Griffin '97]
A. Griffin. *Drivers of NPD Success: The 1997 PDMA Report*. Product Development and Management Association (PDMA), 1997.
- [Hawlitzky '02]
N. Hawlitzky. *Integriertes Qualitätscontrolling von Unternehmensprozessen - Gestaltung eines Quality Gate-Konzeptes*. TCW Transfer-Centrum, 2002.
- [Hedberg '04]
H. Hedberg. *Introducing the Next Generation of Software Inspection Tools*. In: *Product Focused Software Process Improvement*. Springer, Auflage: 234-247, 2004.

- [Hindel et al. '06]
B. Hindel, K. Hörmann, M. Müller und S. Jürgen. *Basiswissen Software-Projektmanagement*. dpunkt.verlag, 2006.
- [Hörmann et al. '06]
K. Hörmann, L. Dittmann, B. Hindel und M. Müller. *SPICE in der Praxis. Interpretationshilfe für Anwender und Assessoren*. dpunkt Verlag, 2006.
- [Houdek '06]
F. Houdek. *Management von Softwareprojekten*. Vorlesung. Institut für Programmiermethodik und Compilerbau, Universität Ulm, 2006.
- [Huff '96]
K. E. Huff. *Software Process Modeling*. In: *Software Process*. John Wiley & Sons, Auflage: 97-106, 1996.
- [Hughes et al. '02]
B. Hughes und M. Cotterell. *Software project management*. McGraw-Hill, Auflage: 3, 2002.
- [Humphrey '89]
W. S. Humphrey. *Managing the Software Process*. Addison Wesley, 1989.
- [Humphrey '92]
W. S. Humphrey. *Introduction to Software Process Improvement*. Software Engineering Institute (SEI), Carnegie Mellon University, 1992.
Link: <http://www.sei.cmu.edu/pub/documents/92.reports/pdf/tr07.92.pdf>
Letzter Zugriff: 18. März 2008
- [IDEF0 '93]
Integration Definition for Function Modeling (IDEF0). 1993.
Link: <http://www.idef.com/idef0.html>
Letzter Zugriff: 26. Juli 2007
- [IEEEESG '90]
IEEE Standard Glossary of Software Engineering Terminology (IEEE Std. 610.12-1990). Institute of Electrical and Electronics Engineers (IEEE), 1990.
- [Ittner '05]
J. Ittner. *Software-Assisted Tailoring of Process Descriptions*. Doktorarbeit. Technische Fakultät, Universität Erlangen-Nürnberg, 2005.
- [Jones et al. '99]
O. Jones und G. Stevens. *Evaluating failure in the innovation process: the micropolitics of new product development*. In: *R&D Management*. Band 29(2): S. 167-178, 1999.
- [Kamiske et al. '05]
G. F. Kamiske und J.-P. Brauer. *Qualitätsmanagement von A bis Z - Erläuterungen moderner Begriffe des Qualitätsmanagement*. Carl Hanser Verlag, 2005.
- [Kaplan et al. '92]
R. Kaplan und D. Norton. *The Balanced Scorecard Measures That Drive Performance*. In: *Harvard Business Review*. Band 1992(Jan-Feb): S. 71-80, 1992.
- [Karlström et al. '05]
D. Karlström und P. Runeson. *Combining Agile Methods with Stage-Gate Project Management*. In: *IEEE Software*. Band 22(3): S. 43-49, 2005.

- [Knauss '07]
E. Knauss. *Einsatz computergestützter Kritiken für Anforderungen*. In: GI Softwaretechnik-Trends. Band 27(1): S. 27-28, 2007.
- [Knauss et al. '07]
E. Knauss und T. Flohr. *Managing Requirement Engineering Processes by Adapted Quality Gateways and critique-based RE-Tools*. In: Workshop on Measuring Requirements for Project and Product Success at 19th International Conference on Advanced Information Systems Engineering. Palma de Mallorca, Spain. S. 40-53, 2007.
- [Kruchten '99]
P. Kruchten. *Der Rational Unified Process - Eine Einführung*. Addison-Wesley, Auflage: 2, 1999.
- [Leszak et al. '02]
M. Leszak und W. Kammerer. *QDB - a Flexible Environment for Process and Quality Management*. In: CONQUEST 2002. Nuernberg, Germany. 2002.
- [Leszak '04]
M. Leszak. *Process Modeling and Quality Control for Embedded Telecommunication Systems*. In: IEEE Workshop on Process Modeling and Simulation (ProSim). Edinburgh, UK. 2004.
- [Liggesmeyer '02]
P. Liggesmeyer. *Software-Qualität. Testen, Analysieren und Verifizieren von Software*. Spektrum Akademischer Verlag, 2002.
- [Liro '04]
T. Liro. *Konzept und Realisierung eines Werkzeugs zur Erhebung, Verwaltung und Bewertung von Erfahrungen*. Bachelorarbeit. Fachgebiet Software Engineering, Gottfried Wilhelm Leibniz Universität Hannover, 2004.
- [Lübke et al. '04]
D. Lübke, T. Flohr und K. Schneider. *Serious Insights through Playful Software-Projects*. In: 11th European Conference: European Software Process Improvement Initiative (EuroSPI). Trondheim, Norway. Springer, S. 57-68, 2004.
- [Lübke et al. '06]
D. Lübke und T. Flohr. *Simulated Software Project Driven by Quality Gates*. In: Electronics World. Band 2006(1840): S. 38-42, 2006.
- [Macdonald et al. '96]
F. Macdonald, J. Miller, A. Brooks, M. Roper und M. Wood. *Automating the software inspection process*. In: Automated Software Engineering: An International Journal. Band 3(3-4): S. 193-283, 1996.
- [Macdonald '98]
F. Macdonald. *Computer Supported Software Inspection*. Doktorarbeit. Department of Computer Science, University of Strathclyde, 1998.
- [McCabe '76]
T. J. McCabe. *A Complexity Measure*. In: IEEE Transactions on Software Engineering. Band 2(4): S. 308-320, 1976.
- [Mulder '97]
L. Mulder. *The importance of a common project management method in the corporate environment*. In: R&D Management. Band 27(3): S. 189-196, 1997.

[Müller et al. '99]

G. Müller und P. Reindl. *Der BMW DMU-Prozess mit Entwicklungspartnern*. In: Virtuelle Produktentstehung in der Fahrzeugtechnik. Berlin. VDI-Verlag, S. 135-143, 1999.

[Oeltjenbruns '02]

H. Oeltjenbruns. *Projektsteuerung mit Quality Gates – Erfahrungsbericht eines Baureihenleiters der DaimlerChrysler AG*. 2002.

Link: http://www2.imab.tu-clausthal.de/files/anlagenprojektierung/aktuelles/2_Vortrag%20Oeltjenbruns.pdf

Letzter Zugriff: 18. Dezember 2006

[Peucker '07]

C. Peucker. *Ein Anpassungskonzept basierend auf Bausteinen für Quality Gates*. Masterarbeit. Fachgebiet Software Engineering, Gottfried Wilhelm Leibniz Universität Hannover, 2007.

[Pfeifer et al. '03]

T. Pfeifer und R. Schmidt. *Das Quality-Gate-Konzept: Entwicklungsprojekte softwareintensiver Systeme verlässlich planen*. In: Industrie Management. Band 19(5): S. 21-24, 2003.

[Porter et al. '95]

A. Porter und H. Siy. *An experiment to assess the cost-benefits of code inspections in large scale software development*. In: SIGSOFT Software Engineering Notes. Band 20(4): S. 92-103, 1995.

[Projektinsight '08]

Projektinsight - Web-Based Project Management Software. Projektinsight, 2008.

Link: <http://www.projectinsight.net/Tour/ProjectDashboard.aspx>

Letzter Zugriff: 13. Februar 2008

[RDAST '94]

Research, Development, and Acquisition - Systems Transitioning. Department of the Army, U.S. Army Tank-Automotive and Armanent Command Warren, 1994.

Link: <http://www.fleetmgt.army.mil/docs/TACOM-Reg-70-68-Systems-Transitioning.doc>

Letzter Zugriff: 4. Februar 2007

[Richter '06]

B. Richter. *Vorbereitung und Durchführung einer Umfrage zur Untersuchung des ERP-KMU-Dilemmas*. Studienarbeit. Department für Informatik, Abteilung für Wirtschaftsinformatik, Carl von Ossietzky Universität Oldenburg, 2006.

[Robertson et al. '99]

S. Robertson und J. Robertson. *Mastering the Requirements Process*. Addison-Wesley Professional, 1999.

[Royce '98]

W. Royce. *Software Project Management: A Unified Framework*. Addison-Wesley Professional, 1998.

[Ruffler et al. '06]

M. Ruffler und M. Leszak. *Software Quality Assessment – A Tool-Supported Model*. In: International Workshop on Software Metrics and DASMA Software Metrik Kongress (IWSM/Metrikon 2006). Potsdam, Deutschland. Shaker Verlag, S. 51-66, 2006.

[Russel '91]

G. W. Russel. *Experience with inspection in ultralarge-scale developments*. In: IEEE Software. Band 8(1): S. 25-31, 1991.

- [Sapsomboon '99]
B. Sapsomboon. *Software Inspection and Computer Support*. Doktorarbeit. Department of Information Science and Telecommunications, School of Information Sciences, University of Pittsburgh, 1999.
- [Scacchi '04]
W. Scacchi. *Free and Open Source Development Practices in the Game Community*. In: IEEE Software. Band 21(1): S. 59-66, 2004.
- [Scharer '01]
M. Scharer. *Quality Gate-Ansatz mit integriertem Risikomanagement*. Doktorarbeit. Forschungsberichte aus dem Institut für Werkzeugmaschinen und Betriebstechnik der Universität Karlsruhe, Institut für Werkzeugmaschinen und Betriebstechnik der Universität Karlsruhe, 2001.
- [Schneider '00]
K. Schneider. *LIDs: A Light-Weight Approach to Experience Elicitation and Reuse*. In: Product Focused Software Process Improvement (PROFES 2000). Oulo, Finland. Springer, S. 407-424, 2000.
- [Schneider et al. '05]
K. Schneider und D. Lübke. *Systematic Tailoring of Quality Techniques*. In: 3rd World Congress for Software Quality. Munich, Germany. 2005.
- [Schneider '07a]
K. Schneider. *Grundlagen der Softwaretechnik*. Vorlesung. Fachgebiet Software Engineering, 2007a.
- [Schneider '07b]
K. Schneider. *Abenteuer Softwarequalität*. dpunkt.verlag, 2007b.
- [Scholz '06]
D. Scholz. *Entwurf und Implementierung von NetQGate mit J2EE*. Masterarbeit. Fachgebiet Software Engineering, Gottfried Wilhelm Leibniz Universität Hannover, 2006.
- [Schubert et al. '06]
P. Schubert, T. Guiver, R. MacDonald und F. Yu. *Using Quality Measures to Manage Statistical Risks in Business Surveys*. In: European Conference on Quality in Survey Statistics. Cardiff, UK. 2006.
- [Schulz '07]
C. Schulz. *Tailoring von Gate-Netzwerken*. Bachelorarbeit. Fachgebiet Software Engineering, Gottfried Wilhelm Leibniz Universität Hannover, 2007.
- [Solingen et al. '99]
R. v. Solingen und E. Berghout. *The Goal/Question/Metric Method - A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill, 1999.
- [Sommerville '07]
I. Sommerville. *Software Engineering*. Pearson Studium, Auflage: 8, 2007.
- [SPEMSpec '05]
Software Process Engineering Metamodel Specification 1.1. Object Management Group, 2005.
Link: <http://www.omg.org/docs/formal/05-01-06.pdf>
Letzter Zugriff: 5. Dezember 2006

[Stachowiak '73]

H. Stachowiak. *Allgemeine Modelltheorie*. Springer, 1973.

[Stanic '06]

D. Stanic. *Werkzeuggestützte Reviews*. Bachelorarbeit. Fachgebiet Software Engineering, Gottfried Wilhelm Leibniz Universität Hannover, 2006.

[Tasc '06]

Process - Phases. Tasc, 2006.

Link: <http://www.tasc-solutions.com/ObjectMetrix/pages/process.htm>

Letzter Zugriff: 25. Oktober 2006

[TQU '05]

Stage-Gate, der Wegweiser zwischen Idee und Innovation. Transferzentren Qualität im Unternehmen (TQU), 2005.

Link: <http://www.tqu.com/downloads/stagegate.pdf>

Letzter Zugriff: 20. April 2007

[UMLSpec '01]

OMG Unified Modeling Language Specification (Version 1.4.2). Object Management Group (OMG), 2001.

Link: <http://www.omg.org/cgi-bin/apps/doc?formal/05-04-01.pdf>

Letzter Zugriff: 18. März 2008

[VDA '98]

Sicherung der Qualität vor Serieneinsatz: Projektplanung. In: Qualitätsmanagement in der Automobilindustrie. Band 4(3): S. 1-53, 1998.

[VMXT '06]

V-Modell XT (Version 1.2). Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung, 2006.

Link: <http://www.v-modell.iabg.de/>

Letzter Zugriff: 18. März 2008

[VMXT-PA '06]

V-Modell XT Projektassistent (1.2). Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung, 2006.

Link: <http://ftp.uni-kl.de/pub/v-modell-xt/Release-1.2/Werkzeuge/Projektassistent/>

Letzter Zugriff: 30. Januar 2007

[Votta '93]

L. G. Votta. *Does every inspection need a meeting?* In: 1st ACM SIGSOFT symposium on Foundations of software engineering. Los Angeles, USA. ACM Press, S. 107-114, 1993.

[Wallin et al. '02]

C. Wallin, F. Ekdahl und S. Larsson. *Integrating Business and Software Development Models*. In: IEEE Software. Band 19(6): S. 28-33, 2002.

[Wallin et al. '02]

C. Wallin, S. Larsson, F. Ekdahl und I. Crnkovic. *Combining Models for Business Decisions and Software Development*. In: 28th Euromicro Conference (EUROMICRO'02). Dortmund, Germany. 2002.

[Walton '86]

M. Walton. *The Deming management method*. Perigee, 1986.

[Wieggers '02]

K. E. Wieggers. *Peer Reviews in Software: A Practical Guide*. Addison-Wesley, 2002.

[Wikipedia '07]

Erläuterung des Begriffs Multiprojektmanagement. Wikipedia, 2007.

Link: <http://de.wikipedia.org/wiki/Multiprojektmanagement>

Letzter Zugriff: 4. Juni 2007

[Wikipedia '08]

Erläuterung des Begriffs Konzept. Wikipedia, 2008.

Link: <http://de.wikipedia.org/wiki/Konzept>

Letzter Zugriff: 20. März 2008

[Wildemann '03]

H. Wildemann. *Controlling im Total Quality Management*. In: *Qualitätsmanagement im Unternehmen. Grundlagen, Methoden und Werkzeuge, Praxisbeispiele*. Symposion, Auflage: 1-29, 2003.

[Wißler '05]

F. E. Wißler. *Ein Verfahren zur Bewertung technischer Risiken in der Phase der Entwicklung komplexer Serienprodukte*. Doktorarbeit. Fakultät für Maschinenbau, Universität Stuttgart, 2005.

[Wolski et al. '08]

P. Wolski, M. Beck, X. Velasquez Zambrana, A. Schütte und R. Pham. *Softwareprojekt RiskQG*. Softwareprojekt. Fachgebiet Software Engineering, Gottfried Wilhelm Leibniz Universität Hannover, 2008.

[Yourdon '89]

E. Yourdon. *Structured Walkthroughs*. Prentice Hall, 1989.

[Zuser et al. '01]

W. Zuser, S. Biffel, T. Grechenig und M. Köhle. *Software Engineering mit UML und dem Unified Process*. Pearson Studium, 2001.

Anhang A – Fragebogen und Auswertung

FG Software Engineering
Universität Hannover
Welfengarten 1
3. Stock, Flur G

Fragebogen
Quality Gates und
alternative Ansätze



Hinweis: Bitte Ankreuzungen durch doppeltes Klicken auf das jeweilige Kästchen und anschließendes Klicken auf „Aktiviert“ (Standardwert) und „OK“ vornehmen.

1. Welche Aufgabe haben Sie in Ihrem Unternehmen? (Mehrfachnennungen möglich)

Projektmanagement Qualitätsmanagement

Weitere Aufgaben:

2. Werden in Ihrem Unternehmen ein Quality Gates verwendet?

ja nein

Falls bei Ihnen kein solcher Ansatz verwendet wird, durch welchen Ansatz wird bei Ihnen im Unternehmen formal geprüft, ob die Eingangskriterien von Dokumenten für Phasen oder Iterationen erfüllt sind?

- Management-Review Meilenstein-Review
 QA Meilenstein Stage-Gate
 Quality-Gateway (Projekt-)Entscheidungspunkt
 Ein solches Konzept existiert bei uns nicht

Durch keinen der oben genannten Ansätze, sondern durch:

Die folgenden Fragen bitte nur ausfüllen, falls Sie bei Frage 2 einen Ansatz angekreuzt oder genannt haben! Falls nicht, schicken Sie bitte die Fragebögen an uns zurück.

3. Beschreiben Sie kurz die Kernkonzepte Ihres Ansatzes so wie er in Ihrem Unternehmen angewendet wird:

4. Zu welchen Zeitpunkten im Projekt wird Ihr Ansatz eingesetzt und durch was wird er ausgelöst?

5. Beschreiben Sie kurz den Ablauf einer Durchführung Ihres Ansatzes:

6. Welche Dokumente werden bei einer Durchführung Ihres Ansatzes geprüft und was wird geprüft?

7. Welche Rollen sind an der Durchführung Ihres Ansatzes beteiligt und welche Aufgaben und welches Profil haben sie jeweils?

8. Beschreiben Sie kurz das Verhältnis von Meilensteinen zu Ihrem Ansatz:

9. Gibt es Projektmerkmale, nach denen entschieden wird, ob Ihr Ansatz eingesetzt wird oder nicht (zum Beispiel Projektgröße oder Risiko)?

10. Welche alternativen Bezeichnungen gibt es für ihren Ansatz in Ihrem Unternehmen?

11. Welche Ausgänge kann Ihr Ansatz haben? (Mehrfachnennungen möglich)

Projektabbruch

Eintritt in nächste Phase / Iteration

Projekt kann zurückgestellt werden

Erneute Vorlage der geforderten Dokumente nach Überarbeitung

Weitere Ausgänge:

12. Basiert Ihr Ansatz auf einem gängigen Referenzmodell?

ja, und zwar _____

nein, es handelt sich um einen unternehmenseigenen Ansatz.

Falls Sie mit „nein“ geantwortet haben: Ist Ihr Ansatz dokumentiert und formalisiert?

ja

nein

13. Projekten liegt ein Vorgehensmodell zu Grunde. Bei welchen Vorgehensmodellen wird Ihr Ansatz eingesetzt? (Mehrfachnennungen möglich)

- Sequenziell / Wasserfallmodell
- Iterative Modelle

14. Zu welchem Zweck wird Ihr Ansatz in Ihrem Unternehmen eingesetzt? (Mehrfachnennungen möglich)

- Qualitätssicherung
- Projektmanagement
- Risikomanagement / Identifikation von Risiken und Potentialen

Weitere Zwecke und Ergänzungen:

15. In welchen Projekten wird Ihr Ansatz durchgeführt? (Mehrfachnennungen möglich)

- Standardsoftware
- Eigenentwicklungs-Projekte
- Auftraggeber-Projekte mit externen Kunden

Weitere Projekttypen:

16. Welche Arten von Software werden in Ihrem Unternehmen entwickelt und wird Ihr Ansatz dann jeweils eingesetzt oder nicht? (Mehrfachnennungen möglich)

Art der Software	Ansatz wird eingesetzt?
<input type="checkbox"/> Anwendungssoftware	<input type="checkbox"/>
<input type="checkbox"/> Eingebettete Software	<input type="checkbox"/>
<input type="checkbox"/> Verteilte Software / Web-Anwendungen	<input type="checkbox"/>
<input type="checkbox"/> Systemsoftware	<input type="checkbox"/>

Falls Ihr Ansatz nicht bei allen (für sie zutreffenden) Softwaretypen durchgeführt werden, welches sind die Gründe dafür?

17. Ist in Ihrem Unternehmen der Ansatz projektspezifisch oder projektübergreifend definiert? Unterscheidet sich also Ihr Ansatz in der Anzahl der Durchführungen, den Vorgaben, den beteiligten Rollen und weiteren Merkmalen je nach Projekt?

Der Ansatz ist bei uns projektübergreifend immer gleich definiert

Falls Ihr Ansatz bei Ihnen nicht projektübergreifend definiert ist, welche Aussagen sind dann richtig? (Mehrfachnennungen möglich)

Die Anzahl der Durchführungen ist projektspezifisch

Die Vorgaben jeder einzelnen Durchführung sind projektspezifisch

Die beteiligten Rollen sind projektspezifisch

Der Ansatz wird in folgender Weise noch weiter spezifiziert:

18. In welcher Form sind die Prüfkriterien festgehalten?

Checkliste

Die Prüfkriterien sind nur informell festgehalten

Weitere Formen:

19. Wird der Ansatz in Ihrem Unternehmen durch Softwarewerkzeuge zumindest teilweise unterstützt?

ja

nein

Falls Sie mit „ja“ geantwortet haben, um welche Softwareprodukte handelt es sich und welche Aspekte werden damit unterstützt?

Falls Sie mit „nein“ geantwortet haben, können Sie sich vorstellen, dass Ihr Ansatz durch Software teilweise unterstützt werden? Falls ja, wie könnte so eine Unterstützung aussehen?

20. Gibt es Schwachpunkte bei Ihrem Ansatz?

--

21. Wird Ihr Ansatz in Ihrem Unternehmen kontinuierlich verbessert?

ja

nein

22. Gibt es noch Aspekte, die durch obige Fragen bezüglich ihres Ansatzes noch nicht beleuchtet wurden?

--

Vielen Dank, dass Sie sich Zeit zum Ausfüllen dieses Fragebogens genommen haben! Aus Erfahrung wissen wir, dass sich aus gegebenen Antworten immer noch Fragen ergeben können. Aus diesem Grund möchten wir Sie fragen, ob Sie für ein telefonisches Interview oder vielleicht per E-Mail für Rückfragen zur Verfügung stehen? Falls ja, tragen Sie einfach unten in die entsprechenden Felder Ihren Namen, Ihre Telefonnummer und E-Mail-Adresse ein.

Name:

Telefonnummer:

E-Mail-Adresse:

Name des Ansatzes	Einbettung	Ablauf	Dokumente und Kriterien	Rollen & Aufgaben	Projektmerkmale	Entscheidung	Referenzmodell	Vorgehensmodell	Einsatzzweck	Projekttypen	Software-typen	Anpassung	Checkliste	Software-werkzeug	Verbesserung
Quality Gate	zwischen Phasen, spezieller Meilenstein	K,I,S,A	Dokumente	GM,P,Gu, Gk	Zeit und Geld	G,K,H,R	eigener Ansatz	Wasserfall	P,Q,R	E,K	A	A,K,R	Checkliste	Nein	Ja
Quality Gate	zwischen Phasen, spezieller Meilenstein	K,I,S,Ü	Nachweis von Aktivitäten	GV,Gu,Gk	*	G,H	eigener Ansatz	Iterativ	Q,R	E,K	E,V	A	Checkliste	ja, alle Aspekte	ja
Quality Gate	zwischen Phasen, Iterationen, spezieller Meilenstein	A,Ü	Nachweis von Aktivitäten	GV, Gu	immer	G,K,H	ja, V-Modell	Iterativ	P,Q,R	S,E,K	E	A	Checkliste	ja, DVB	ja
Meilenstein-Review, identisch mit vorgegeben Meilenstein	durch Vorgehensmodell vorgegeben	*	Dokumente	P,Gu,Gk	immer	G	eigener Ansatz	Wasserfall	P	E	E	Ansatz immer gleich	Checkliste	ja, PMW	*
Quality Gate	zwischen Phasen, spezieller Meilenstein	A	Nachweis von Aktivitäten	P, Gk	immer	G,R	eigener Ansatz	Wasserfall	Q,P,R	*	E	K	Checkliste	ja, DVB	ja
Quality Gate	zwischen Phasen, spezieller Meilenstein	*	Nachweis von Aktivitäten	*	nur bei wichtigen Projekten	G,K,H,R	eigener Ansatz	*	*	*	*	A	*	ja, DVB	*
Stage-Gate	zwischen Phasen, spezieller Meilenstein	I,S	Nachweis von Aktivitäten, BG	P,Gu,Gk	nach Projektgröße	G,K,H,R	ja, Stage-Gate	Wasserfall, Iterativ	Q	S,E	A,E,V,S	A,K	Checkliste	nein	ja
Name anonymisiert	zwischen Phasen, spezieller Meilenstein	I,S	Dokumente, BG	GV,Gu,Gk	nach Projektgröße	G,K	ja, Stage-Gate	Wasserfall, Iterativ	Q,R	*	*	A,K,R	Checkliste	*	*
Name anonymisiert	zwischen Phasen, spezieller Meilenstein	V	Dokumente oder Phasen-report	GV,P,Gu,Gk	*	G	ja, Stage-Gate	Wasserfall	P,Q,R	*	*	*	Checkliste	*	*
Name anonymisiert	zwischen Phasen, Iterationen, spezieller Meilenstein	*	Dokumente, BG	GV,P,Gu	*	G	ja, Stage-Gate	*	*	*	*	K	*	*	*
Name anonymisiert	zwischen Phasen, spezieller Meilenstein	*	*	*	*	*	ja, Stage-Gate	*	*	*	*	*	*	*	*

Legende zur Auswertungstabelle der Fragebögen – Teil 1

Kategorie	Abkürzung	Bedeutung
Ablauf	Kr	Erstellung von Kriterien
	K	Kickoff
	I	Individuelle Vorbereitung
	S	Gate-Sitzung
	Ü	Überwachung von Maßnahmen
	A	Verdichtung der Daten auf Ampelskala
Dokumente und Kriterien	BG	Geschäftsziele
Rollen und Aufgaben	GM	Gate-Management
	GV	Gate-Verantwortlicher
	P	Projektvertreter
	Gu	Gutachter
	Gk	Gatekeeper
Ausgänge	G	Go: Uneingeschränkte Projektfortsetzung
	K	Kill: Projektabbruch
	H	Hold: Zurückstellung des Projektes
	R	Repeat-Gate: Wiederholung des Quality Gates
Einsatzzweck	P	Projektmanagement
	Q	Qualitätsmanagement
	R	Risikomanagement
Projekttypen	S	Standardsoftware
	E	Eigenentwicklungen
	K	Projekt mit externen Kunden
Softwaretypen	A	Anwendungssoftware
	E	Eingebettete Software
	V	Verteilte Software/Web-Anwendungen
	S	Systemsoftware
Anpassung	A	Die Anzahl der Gates ist projektspezifisch
	K	Die Kriterien jedes Quality Gates sind projektspezifisch
	R	Die Rollen sind projektspezifisch

Legende zur Auswertungstabelle der Fragebögen – Teil II

Kategorie	Abkürzung	Bedeutung
Software- werkzeug	DVB	Softwarewerkzeug zur Unterstützung der Datensammlung, Datenverdichtung und Erstellung von Berichten
	PMW	Werkzeug zur Unterstützung typischer Aufgaben des Projektmanagements
Allgemein	*	Fehlende oder nicht verwertbare Antwort (im Sinne der Fragestellung)

Anhang B – Mathematische Definition der Zugehörigkeitsfunktionen

Die folgende Tabelle definiert die in Abschnitt 10.3.1 eingeführten Zugehörigkeitsfunktionen als Mathematische Funktionen und nennt jeweils den Einsatzzweck der Zugehörigkeitsfunktion hinsichtlich der Fuzzy-Logik.

Funktion	Mathematische Definition	Zweck
$flramp_{a,b,h}(x)$	$flramp_{a,b,h}(x) = \begin{cases} h, & \text{falls } x \leq a \\ \frac{b-x}{b-a} \cdot h, & \text{falls } a < x \leq b \\ 0, & \text{falls } x > b \end{cases}$	Modellierung von abfallender Zugehörigkeit bei steigendem x und Modellierung eines unendlichen Bereichs mit Zugehörigkeit 0.
$clamp_{a,b,h}(x)$	$clamp_{a,b,h}(x) = \begin{cases} 0, & \text{falls } x \leq a \\ \frac{x-a}{b-a} \cdot h, & \text{falls } a < x \leq b \\ h, & \text{falls } x > b \end{cases}$	Modellierung von steigender Zugehörigkeit bei steigendem x und Modellierung eines unendlichen Bereichs mit Zugehörigkeit h .
$trap_{a,b,c,d,h}(x)$	$trap_{a,b,c,d,h}(x) = \begin{cases} 0, & \text{falls } x \leq a \\ clramp_{a,b,h}(x), & \text{falls } a < x \leq b \\ h, & \text{falls } b < x \leq c \\ flramp_{c,d,h}(x), & \text{falls } c < x \leq d \\ 0 & \text{falls } x > d \end{cases}$	Modellierung von steigender und fallender Zugehörigkeit am linken bzw. rechten Randbereich und Modellierung eines Plateaus, das das höchste Zugehörigkeitsniveau h darstellt.
$peak_{a,b,c,h}(x)$	$peak_{a,b,c,h}(x) = trap_{a,b,b,c,h}(x)$	Modellierung von steigender und fallender Zugehörigkeit am linken bzw. rechten Randbereich und Modellierung eines singulären Wertes, der die höchste Zugehörigkeit h darstellt.
$sgl_{a,h}(x)$	$sgl_{a,h}(x) = \begin{cases} h, & \text{falls } x = a \\ 0, & \text{sonst} \end{cases}$	Darstellung eines einzelnen singulären Wertes, der das höchste Zugehörigkeitsniveau h darstellt.
$crisp_{a,b,h}(x)$	$crisp_{a,b,h}(x) = \begin{cases} h, & \text{falls } a \leq x \leq b \\ 0, & \text{sonst} \end{cases}$	Darstellung eines Wertebereichs, der das höchste Zugehörigkeitsniveau h darstellt.

Anhang C – Umsetzung des Tailoring-Verfahrens für die studentischen Softwareprojekte

Die folgende Auflistung zeigt die Zugehörigkeitsfunktionen geordnet nach Projektmerkmalen, die notwendig sind, um das Tailoring-Verfahren für die studentischen Softwareprojekte umzusetzen. Zunächst werden die Zugehörigkeitsfunktionen für die Projektmerkmale

A_1 = Projektgegenstand, A_2 = Benötigte Technologien und Methoden (Zielarchitektur) und
 A_3 = Benötigte Technologien und Methoden (Methoden)

aufgestellt:

- **Projektgegenstand:**

$$\mu_{\alpha_{1,i}} : X_1 \rightarrow [0;1]$$

$$\mu_{\alpha_{1,i}}(x) = sgl_{w_i,1}(x)$$

mit $i \in \{1, \dots, 3\}$, $(w_1, w_2, w_3) = (\text{Eingebettete Software, Komplexes System, andere Software})$
 und $X_1 = \{\text{Eingebettete Software, Komplexes System, andere Software}\}$.

- **Benötigte Technologien und Methoden (Zielarchitektur):**

$$\mu_{\alpha_{2,i}} : X_2 \rightarrow [0;1]$$

$$\mu_{\alpha_{2,i}}(x) = sgl_{w_i,1}(x)$$

mit $i \in \{1, \dots, 3\}$, $(w_1, w_2, w_3) = (\text{SOA, Client-Server, andere Architektur})$
 und $X_2 = \{\text{SOA, Client-Server, andere Architektur}\}$.

- **Benötigte Technologien und Methoden (Netzwerkprogrammierung):**

$$\mu_{\alpha_{3,i}} : \{\text{ja, nein}\} \rightarrow [0;1]$$

$$\mu_{\alpha_{3,i}}(x) = sgl_{w_i,1}(x)$$

mit $i \in \{1, 2\}$ und $(w_1, w_2) = (\text{ja, nein})$.

- **Benötigte Technologien und Methoden (Oberflächenprogrammierung):**

$$\mu_{\alpha_{4,i}} : \{\text{ja, nein}\} \rightarrow [0;1]$$

$$\mu_{\alpha_{4,i}}(x) = sgl_{w_i,1}(x)$$

mit $i \in \{1, 2\}$ und $(w_1, w_2) = (\text{ja, nein})$.

- **Benötigte Ressourcen und Umfang (Umfang):**

Linguistische Werte: $\alpha_{5,1} = \text{einfach}$, $\alpha_{5,2} = \text{mittel}$, $\alpha_{5,3} = \text{komplex}$ und $\alpha_{5,4} = \text{sehr komplex}$.

$x \in \mathbb{N}^0$ gibt die Anzahl der funktionalen Anforderungen in der Anforderungsspezifikation an.

$$\begin{array}{ll} \mu_{\alpha_{5,1}} : N^0 \rightarrow [0;1] & \mu_{\alpha_{5,2}} : N^0 \rightarrow [0;1] \\ \mu_{\alpha_{5,1}}(x) = flramp_{9,18,1}(y) & \mu_{\alpha_{5,2}}(x) = trap_{9,18,27,39,1}(y) \\ \\ \mu_{\alpha_{5,3}} : N^0 \rightarrow [0;1] & \mu_{\alpha_{5,4}} : N^0 \rightarrow [0;1] \\ \mu_{\alpha_{5,3}}(x) = trap_{27,39,50,58,1}(y) & \mu_{\alpha_{5,4}}(x) = clramp_{50,58,1}(y) \end{array}$$

Die Stellgrößen sind die folgenden Kriterien:

1. „Alle Methoden des Quelltextes weisen einen McCabe-Wert ≤ 10 auf?“
2. „Es liegen Mockups oder Prototypen der grafischen Schnittstelle (GUI) vor?“
3. „Abschätzungen zum Speicherverbrauch während der Laufzeit liegen vor?“
4. „Angaben zur physischen Verteilung der einzelnen Softwarekomponenten liegen vor?“
5. „Netzwerkprotokolle mit entsprechenden Abläufen und Nachrichtenformaten liegen vor?“
6. „Eine Beschreibung der Funktionsverteilung auf verschiedene Rechner oder Systemteile und eine Beschreibung der Kommunikation unter ihnen liegt vor?“
7. „Eine Beschreibung der physischen Verteilung der einzelnen Softwarekomponenten muss liegt vor?“
8. „Der Entwurf ist nach dem MVC-Architekturmuster aufgeteilt?“

Für jedes der Kriterien B_i mit $i \in \{1, \dots, 8\}$ sind die Zugehörigkeitsfunktion für die Projektmerkmalswerte $\beta_{i,1}$ = unverzichtbar, $\beta_{i,2}$ = empfohlen und $\beta_{i,3}$ = verzichtbar wie folgt definiert:

$$\begin{array}{ll} \mu_{\beta_{i,1}} : [0;1] \rightarrow [0;1] & \mu_{\beta_{i,2}} : [0;1] \rightarrow [0;1] \\ \mu_{\beta_{i,1}}(x) = clramp_{0,7,0,8,1}(y) & \mu_{\beta_{i,2}}(x) = trap_{0,15,0,3,0,7,0,8,1}(y) \\ \\ & \mu_{\beta_{i,3}} : [0;1] \rightarrow [0;1] \\ & \mu_{\beta_{i,3}}(x) = flramp_{0,15,0,3,1}(y) \end{array}$$

Die Regeln gemäß Abschnitt 13.1.5 ergeben das folgende Regelwerk:

Kriterium Nr.	Notwendige Regeln
1	$\mu_{\alpha_{1,1}} \Rightarrow \mu_{\beta_{1,3}}, \mu_{\alpha_{1,2}} \vee \mu_{\alpha_{1,3}} \Rightarrow \mu_{\beta_{1,1}}$
2	$\mu_{\alpha_{1,1}} \Rightarrow \mu_{\beta_{2,3}}, \mu_{\alpha_{1,2}} \vee \mu_{\alpha_{1,3}} \Rightarrow \mu_{\beta_{2,1}}$
3	$\mu_{\alpha_{1,1}} \Rightarrow \mu_{\beta_{3,3}}, \mu_{\alpha_{1,2}} \vee \mu_{\alpha_{1,3}} \Rightarrow \mu_{\beta_{3,1}}$
4	$\mu_{\alpha_{2,1}} \vee \mu_{\alpha_{2,2}} \Rightarrow \mu_{\beta_{4,1}}, \mu_{\alpha_{2,3}} \Rightarrow \mu_{\beta_{4,3}}$
5	$\mu_{\alpha_{2,1}} \vee \mu_{\alpha_{2,2}} \vee \mu_{\alpha_{3,1}} \Rightarrow \mu_{\beta_{5,1}}, \mu_{\alpha_{2,3}} \wedge \mu_{\alpha_{3,2}} \Rightarrow \mu_{\beta_{5,3}}$
6	$\mu_{\alpha_{2,1}} \vee \mu_{\alpha_{2,2}} \vee \mu_{\alpha_{3,1}} \Rightarrow \mu_{\beta_{6,1}}, \mu_{\alpha_{2,3}} \wedge \mu_{\alpha_{3,2}} \Rightarrow \mu_{\beta_{6,3}}$
7	$\mu_{\alpha_{2,1}} \vee \mu_{\alpha_{2,2}} \vee \mu_{\alpha_{3,1}} \Rightarrow \mu_{\beta_{7,1}}, \mu_{\alpha_{2,3}} \wedge \mu_{\alpha_{3,2}} \Rightarrow \mu_{\beta_{7,3}}$
8	$(\mu_{\alpha_{5,3}} \vee \mu_{\alpha_{5,4}}) \wedge \mu_{\alpha_{4,1}} \Rightarrow \mu_{\beta_{8,1}}, (\mu_{\alpha_{5,1}} \vee \mu_{\alpha_{5,2}}) \wedge \mu_{\alpha_{4,1}} \Rightarrow \mu_{\beta_{8,2}},$ $\mu_{\alpha_{4,2}} \Rightarrow \mu_{\beta_{8,3}}$

Als Filter wird die folgende Funktion verwendet:

$$\sigma : [0;1] \rightarrow \{\text{rot, gelb, grün}\}$$

$$\sigma(x) = \begin{cases} \text{rot,} & \text{für } 0 \leq x \leq 20 \\ \text{gelb,} & \text{für } 20 < x \leq 80 \\ \text{grün,} & \text{für } 80 < x \leq 100 \end{cases}$$

Anhang D – Erstellte Kriterien für die Projektarbeit PWS

Die folgenden zwei Tabellen zeigen die Kriterien der drei verschiedenen Quality Gates für die Projektarbeit PWS. Fett gedruckte Kriterien sind hinsichtlich ihres Sollwertes veränderlich. Mit * gekennzeichnete Kriterien wurden durch die GQM-Methode (jedoch nicht unbedingt ausschließlich) ermittelt. Mit ** Kriterien wurden nicht durch die GQM-Methode identifiziert.

Kategorie	Fertig für Iteration 2	Fertig für Iteration 3	Fertig für Abnahme des Releases
Allgemeine Anforderungen	Die Anforderungsspezifikation der Iteration liegt vor?	Die Anforderungsspezifikation der Iteration liegt vor?	Die Anforderungsspezifikation der Iteration liegt vor?
	Es liegt ein funktionsfähiges Ant-Build-File vor, mit dem der Quellcode getestet, kompiliert und mit Cobertura vermessen werden kann?	Es liegt ein funktionsfähiges Ant-Build-File vor, mit dem der Quellcode getestet, kompiliert und mit Cobertura vermessen werden kann?	Es liegt ein funktionsfähiges Ant-Build-File vor, mit dem der Quellcode getestet, kompiliert und mit Cobertura vermessen werden kann?
	Der Quellcode liegt mit dem Tag PWS-0708-[A B]-[1] in Subversion vor?	Der Quellcode liegt mit dem Tag PWS-0708-[A B]-[2] in Subversion vor?	Der Quellcode liegt mit dem Tag PWS-0708-[A B]-[3] in Subversion vor?
Allgemeine Entwurf	Die Beschreibung der Webservices in WSDL ist vorhanden?	Die Beschreibung der Webservices in WSDL ist vorhanden?	Die Beschreibung der Webservices in WSDL ist vorhanden?
	Die Architektur ist im Entwurfsdokument vollständig fixiert?*	Die Architektur ist im Entwurfsdokument vollständig fixiert?*	Die Architektur ist im Entwurfsdokument vollständig fixiert?*
	Die Beschreibung der Webservices in WSDL ist zwischen den Teams abgestimmt worden?*	Die Beschreibung der Webservices in WSDL ist zwischen den Teams abgestimmt worden?*	Die Beschreibung der Webservices in WSDL ist zwischen den Teams abgestimmt worden?*
Anforderungen an den Quellcode	Alle Methoden haben einen McCabe-Wert ≤ 10 ?*	Alle Methoden haben einen McCabe-Wert ≤ 10 ?*	Alle Methoden haben einen McCabe-Wert ≤ 10 ?*
	Min. 50% aller Packages, Klassen, Methoden und Variablen sind kommentiert?*	Min. 75% aller Packages, Klassen, Methoden und Variablen sind kommentiert?*	100% aller Packages, Klassen, Methoden und Variablen sind kommentiert?*
	Die Java-Code-Konventionen wurden eingehalten?*	Die Java-Code-Konventionen wurden eingehalten?*	Die Java-Code-Konventionen wurden eingehalten?*
	Alle Instanzvariablen sind nicht öffentlich und der Zugriff erfolgt über get- und set-Methoden?*	Alle Instanzvariablen sind nicht öffentlich und der Zugriff erfolgt über get- und set-Methoden?*	Alle Instanzvariablen sind nicht öffentlich und der Zugriff erfolgt über get- und set-Methoden?*
	Auf den static-Modifier wurde soweit wie möglich verzichtet?*	Auf den static-Modifier wurde soweit wie möglich verzichtet?*	Auf den static-Modifier wurde soweit wie möglich verzichtet?*
	Richtlinien zur Bezeichnung von Packages, Klassen, Methoden und Variablen sind schriftlich fixiert. Bezeichner sind in englischer Sprache und selbstsprechend?*	Richtlinien zur Bezeichnung von Packages, Klassen, Methoden und Variablen sind schriftlich fixiert. Bezeichner sind in englischer Sprache und selbstsprechend?*	Richtlinien zur Bezeichnung von Packages, Klassen, Methoden und Variablen sind schriftlich fixiert. Bezeichner sind in englischer Sprache und selbstsprechend?*
	Die Richtlinien zur Bezeichnung werden befolgt?*	Die Richtlinien zur Bezeichnung werden befolgt?*	Die Richtlinien zur Bezeichnung werden befolgt?*
	Kniffe und Algorithmen im Quellcode sind stets dokumentiert?*	Kniffe und Algorithmen im Quellcode sind stets dokumentiert?*	Kniffe und Algorithmen im Quellcode sind stets dokumentiert?*

Kategorie	Fertig für Iteration 2	Fertig für Iteration 3	Fertig für Abnahme des Releases
Anforderungen an Test und Abnahme	Alle Testfälle sind mit Ein- und Ausgaben versehen, nummeriert und übersichtlich dargestellt?	Alle Testfälle sind mit Ein- und Ausgaben versehen, nummeriert und übersichtlich dargestellt?	Alle Testfälle sind mit Ein- und Ausgaben versehen, nummeriert und übersichtlich dargestellt?
	Ein Namenskürzel bestätigt jeweils, das jeder Testfall erfolgreich läuft?	Ein Namenskürzel bestätigt jeweils, das jeder Testfall erfolgreich läuft?	Ein Namenskürzel bestätigt jeweils, das jeder Testfall erfolgreich läuft?
	Min. 50% der Zeilen und Verzweigungen sind überdeckt?	Min. 60% der Zeilen und Verzweigungen sind überdeckt?	Min. 70% der Zeilen und Verzweigungen sind überdeckt?
	Alle Anforderungen aus der Anforderungsspezifikation sind abgedeckt?*	Alle Anforderungen aus der Anforderungsspezifikation sind abgedeckt?*	Alle Anforderungen aus der Anforderungsspezifikation sind abgedeckt?*
Anforderungen an die Installation	—	—	Eine Installationsbeschreibung liegt vor. Sie beschreibt den aktuellen Installationsvorgang verständlich und ausführlich?*
	—	—	Der beschriebene Installationsvorgang ist mindestens durch eine projektfremde Person praktisch nachvollzogen worden?*

Anhang E – Zugehörigkeitsfunktion für das Tailoring-Verfahren des V-Modells XT

Die folgende Auflistung zeigt die Zugehörigkeitsfunktionen geordnet nach Projektmerkmalen, die notwendig sind, um das Tailoring-Verfahren des V-Modells XT umzusetzen:

- **Projektgegenstand:**

$$\mu_{\alpha_{1,i}} : X_1 \rightarrow [0;1]$$

$$\mu_{\alpha_{1,i}}(x) = sgl_{w_i,1}(x)$$

$$\text{mit } i \in \{1, \dots, 4\},$$

$$(w_1, w_2, w_3, w_4) =$$

(Eingebettetes System, Komplexes System, Softwaresystem, Systemintegration)

und $X_1 = \{\text{Eingebettetes System, Komplexes System, Softwaresystem, Systemintegration}\}$

- **Projektrolle:**

$$\mu_{\alpha_{2,i}} : X_2 \rightarrow [0;1]$$

$$\mu_{\alpha_{2,i}}(x) = sgl_{w_i,1}(x)$$

$$\text{mit } i \in \{1, \dots, 6\},$$

$$(w_1, w_2, w_3, w_4, w_5, w_6) =$$

(Auftraggeber mit einem Auftragnehmer,

Auftraggeber mit mehreren Auftragnehmern,

Auftraggeber und Auftragnehmer mit Unterauftragnehmern,

Auftraggeber und Auftragnehmer ohne Unterauftragnehmer,

Auftragnehmer mit Unterauftragnehmern,

Auftragnehmer ohne Unterauftragnehmer)

$$\text{und } X_2 =$$

\{\text{Auftraggeber mit einem Auftragnehmer,}

Auftraggeber mit mehreren Auftragnehmern,

Auftraggeber und Auftragnehmer mit Unterauftragnehmern,

Auftraggeber und Auftragnehmer ohne Unterauftragnehmer,

Auftragnehmer mit Unterauftragnehmern,

Auftragnehmer ohne Unterauftragnehmer\}

- **Systemlebenszyklusausschnitt:**

$$\mu_{\alpha_{3,i}} : X_3 \rightarrow [0;1]$$

$$\mu_{\alpha_{3,i}}(x) = sgl_{w_i,1}(x)$$

$$\text{mit } i \in \{1, \dots, 3\},$$

$(w_1, w_2, w_3) = (\text{Entwicklung, Wartung und Pflege, Weiterentwicklung und Migration})$

und $X_3 = \{\text{Entwicklung, Wartung und Pflege, Weiterentwicklung und Migration}\}$.

- **Fertigprodukte:**

$$\mu_{\alpha_{4,i}} : \{\text{ja, nein}\} \rightarrow [0;1]$$

$$\mu_{\alpha_{4,i}}(x) = sgl_{w_i,1}(x)$$

mit $i \in \{1, 2\}$ und $(w_1, w_2) = (\text{ja, nein})$.

- **Hohe Realisierungsrisiken:**

$$\mu_{\alpha_{5,i}} : \{\text{ja, nein}\} \rightarrow [0;1]$$

$$\mu_{\alpha_{5,i}}(x) = sgl_{w_i,1}(x)$$

mit $i \in \{1,2\}$ und $(w_1, w_2) = (\text{ja, nein})$.

- **Safety und Security:**

$$\mu_{\alpha_{6,i}} : \{\text{ja, nein}\} \rightarrow [0;1]$$

$$\mu_{\alpha_{6,i}}(x) = sgl_{w_i,1}(x)$$

mit $i \in \{1,2\}$ und $(w_1, w_2) = (\text{ja, nein})$.

Für jede der 82 Stellgrößen ergeben sich jeweils zwei Zugehörigkeitsfunktionen:

$$\mu_{\beta_{i,1}} : [0;1] \rightarrow [0;1]$$

$$\mu_{\beta_{i,1}}(x) = sgl_{1,1}(x)$$

$$\mu_{\beta_{i,2}} : [0;1] \rightarrow [0;1]$$

$$\mu_{\beta_{i,2}}(x) = sgl_{0,1}(x)$$

mit $i \in \{1, \dots, 82\}$.

Die linke Zugehörigkeitsfunktion entspricht dabei dem booleschen Wert wahr, die rechte dem booleschen Wert falsch.

Werdegang

Persönliches

Name	Thomas Flohr
Geburtsdaten	2. Januar 1977 in Burgwedel bei Hannover
Kontaktdaten	Am Haselbusch 27, 30459 Hannover, +49-(0)179-7523163, email@thomas-flohr.com

Wissenschaftlicher Werdegang

1998 – 2004	Studium an der Universität Hannover Studiengang: Mathematik mit Studienrichtung Informatik Anwendungsfach: Wirtschaftsinformatik Abschluss: Diplom-Mathematiker Gesamtnote: „gut“ Titel der Diplomarbeit: „Implementierung von Algorithmen zur Positionierung von Basisstationen in Mobilfunk-Netzen“ Note der Diplomarbeit: „sehr gut“ (1,3) Studienarbeit (Herbst 2001): „Konzeption und Realisierung einer Komponente zur Erfassung und Visualisierung der Mitarbeiteranwesenheit als Teil der IT-Unterstützung der Telekooperation in Unternehmen mit verteilten Organisations-systemen“
1988 – 1997	Integrierte Gesamtschule Mühlenberg in Hannover Abschluss: Allgemeine Hochschulreife mit Note 2,2
1983 – 1988	Wilhelm-Busch-Grundschule in Hannover

Beruflicher Werdegang

2004 – vor. 2009	Wissenschaftlicher Mitarbeiter im Fachgebiet Software Engineering der Gottfried Wilhelm Leibniz Universität Hannover Aufgabenbereiche Durchführung zweier Drittmittelprojekte Betreute Lehrveranstaltungen: <ul style="list-style-type: none">• Laborübung <i>Usability Engineering Labor</i> (2007/08)• <i>Softwareprojekt</i> (2004 und 2006/07)• Laborübung <i>Software-Experiment</i> (2004/05)• Seminar <i>Werkzeuggestützte Softwareprüfungen</i> (2006)• Übung zur Vorlesung <i>Software-Qualität</i> (2005, 2007 und 2008)• Übung zur Vorlesung <i>Grundlagen der Softwaretechnik</i> (2005/06)• Übung zur Vorlesung <i>Anforderungen und Entwurf</i> (2006)
Februar 2001 – März 2001	Studentische Aushilfe bei der IT-P GmbH Hannover Aufgabenbereich Anwendungsprogrammierung
2000 – 2004	Studentische Hilfskraft im Fachbereich Informatik der Gottfried Wilhelm Leibniz Universität Hannover Aufgabenbereiche Übungs-Betreuung Programmieren I (Scheme) und Programmieren II (Java)
1997 – 1998	Zivildienst im Pflegeheim Badenstedt in Hannover