

**Prüfer:** Prof. Dr. rer. nat. Jochen Ludewig  
**Betreuerin:** Dipl. Inform. Patricia Mandl-Striegnitz

**Beginn am:** 15.03.2001  
**Beendet am:** 14.09.2001  
**CR-Klassifikation:** D.2.5, D.2.9, I.6.5, K.6.1, K.6.3, K.6.4

Diplomarbeit Nr. 1931

# **Eine feingranulare SESAM-Variante**

Tilman Hampp

Institut für Informatik  
Universität Stuttgart  
Breitwiesenstr. 20 - 22  
D-70565 Stuttgart

## **Zusammenfassung**

Mit dem SESAM-Projekt (Software Engineering Simulation by Animated Models) der Abteilung Software Engineering an der Universität Stuttgart wird die Ausbildung von Projektleitern um eine Simulationskomponente erweitert. Der Simulator führt dazu Modelle von Software-Projekten aus. Die Modelle enthalten alle für die Ausbildung relevanten Aspekte eines Projekts. Ausnahme ist der Projektleiter, der das simulierte Projekt durchführt.

In der Ausbildung wird das QS-Modell (Qualitätssicherungs-Modell) bereits eingesetzt. Der Schwerpunkt des Modells liegt auf Maßnahmen zur Qualitätssicherung, das Projekt wird von der Analyse bis zur Übergabe an den Kunden nachgebildet. Die Simulation des Modells erfolgt auf Tagesbasis.

Diese Zeitgranularität schränkt die simulierten Effekte ein, da sich Vorgänge innerhalb eines Tages nicht beobachten lassen und der Projektleiter nur tageweise eingreifen kann. Ziel dieser Diplomarbeit ist, die Möglichkeiten, aber auch Probleme einer feineren Zeitgranularität zu untersuchen. Die vermuteten Möglichkeiten und Probleme werden zuerst als Hypothesen formuliert. Um die Hypothesen zu untersuchen, werden einzelne Phasen des QS-Modells auf eine feinere Zeitgranularität von zwei Stunden umgestellt. Diese Variante des QS-Modells wird um feingranulare Effekte erweitert. Der Schwerpunkt liegt dabei auf dem Reviewprozess, der feingranular modelliert wird. Anhand dieser feingranularen Variante werden die aufgestellten Hypothesen überprüft.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Überblick über SESAM	1
1.2	Das QS-Modell	2
1.3	Aufgabenstellung	4
1.4	Überblick über die Arbeit	5
<b>2</b>	<b>Das SESAM-System</b>	<b>6</b>
2.1	Aufbau und Rollen des Systems	6
2.2	SESAM-Modelle	7
<b>3</b>	<b>Hypothesen über den Nutzen und die Probleme einer feineren Zeitgranularität</b>	<b>12</b>
3.1	Die verschiedenen Arten der Granularität	12
3.2	Überblick über die Hypothesen	14
3.3	Auswirkungen auf die Modellierung	15
3.4	Auswirkungen auf die Ausbildung	19
3.5	Technische Aspekte	24
3.6	Auswirkungen auf das QS-Modell	27
<b>4</b>	<b>Umstellung des QS-Modells auf eine feinere Zeitgranularität</b>	<b>29</b>
4.1	Umstellung des QS-Modells	29
4.2	Organisation des Tests	31
4.3	Erzielte Ergebnisse	33
<b>5</b>	<b>Konzeption einer feingranularen Variante</b>	<b>42</b>
5.1	Überblick und Vorgehen	42
5.2	Ausbildungsziele und Abgrenzung des Originals	43
5.3	Effekte und Zusammenhänge der feingranularen Variante	46
5.4	Quantifizierung	60
5.5	Hypothesen zur Modellierung	69
<b>6</b>	<b>Realisierung einer feingranularen Variante</b>	<b>71</b>
6.1	Grundlegende Anforderungen	71
6.2	Grobentwurf	72
6.3	Strukturen der Implementierung	74
6.4	Hypothesen zur Implementierung	79
<b>7</b>	<b>Überprüfung der Ergebnisse</b>	<b>80</b>
7.1	Vorgehen	80
7.2	Projektresultate im Vergleich mit dem QS-Modell	80
7.3	Überprüfung der Hypothesen	83

<b>8</b>	<b>Fazit und Ausblick</b>	<b>91</b>
8.1	Erfahrung mit einer feingranularen Modellvariante	91
8.2	Rückblick auf die Arbeit	92
8.3	Ausblick	95
	<b>Begriffslexikon</b>	<b>97</b>
	<b>Literaturverzeichnis</b>	<b>100</b>

## Abbildungsverzeichnis

Abbildung 1	Projektsimulation mit SESAM	2
Abbildung 2	Aufbau und Rollen des SESAM-Systems	7
Abbildung 3	Komponenten der SESAM-Modelle	10
Abbildung 4	Zeitbegriffe in SESAM-Modellen	10
Abbildung 5	Veränderungen abhängig von der Zeitgranularität	12
Abbildung 6	Gliederung der Hypothesen	14
Abbildung 7	Aufwand und Dauer des Entwurfs	33
Abbildung 8	Phasenaufwand und Kosten im Vergleich	34
Abbildung 9	Umfang der Spezifikation im Vergleich	35
Abbildung 10	Aktion bei grober Zeitgranularität	36
Abbildung 11	Aktion bei feiner Zeitgranularität	37
Abbildung 12	Vergleich der Projektverläufe	38
Abbildung 13	Tagesablauf des Projektleiters	39
Abbildung 14	Ausführungszeit	40
Abbildung 15	Speicherbedarf	41
Abbildung 16	Phasen des technischen Reviews	45
Abbildung 17	Aspekte für die Spezifikation	48
Abbildung 18	Einbettung des Reviews	50
Abbildung 19	Anforderungen und Befunde	51
Abbildung 20	Prüfleistung, Entdeckungsquoten und Umfang	53
Abbildung 21	Fehlerentdeckung mit und ohne Aspekte	54
Abbildung 22	Fehler aus der Vorbereitung	55
Abbildung 23	Empfehlungen des Reviewmodells	58
Abbildung 24	Entdeckungsquote mit und ohne Aspekte	61
Abbildung 25	Fehler in der Vorbereitung und in der Sitzung	62
Abbildung 26	Einfluss des Umfangs auf die Prüfleistung	64
Abbildung 27	Einfluss des Umfangs auf die Entdeckungsquoten	64
Abbildung 28	Einfluss der kritischen Gutachter	65
Abbildung 29	Einfluss diplomatischer Gutachter	65
Abbildung 30	Einfluss des Moderators	66
Abbildung 31	Einfluss der Rolle des Aktuars	66
Abbildung 32	Funktion für die Dauer der Sitzung	66
Abbildung 33	Frustration des Autors	67
Abbildung 34	Grenzwerte für das Spezifikationsreview	68
Abbildung 35	Einfluss der Überstunden auf die Produktivität	69
Abbildung 36	Dateien, Modul und Modell	72
Abbildung 37	Einbettung in das Regelmodell	73
Abbildung 38	Relationsklassen für die Parallelarbeit	74
Abbildung 39	Aufteilung zwischen verschiedenen Tätigkeiten	76
Abbildung 40	Arbeitszeit eines Entwicklers	77
Abbildung 41	Struktur für die Vorbereitung	78

Abbildung 42	Struktur der Reviewsitzung	78
Abbildung 43	Fehler und Verluste bei zwei Gutachtern	81
Abbildung 44	Fehler und Verluste bei drei Gutachtern	81
Abbildung 45	Aufwand im Spezifikationsreview	82
Abbildung 46	Kontrolle der Vorbereitungszeit	83
Abbildung 47	Motivation nach zwei Sitzungen	84
Abbildung 48	Sitzung und Korrektur während der Spezifikation	84
Abbildung 49	Unterschiedliche Vorgehensweisen	85
Abbildung 50	Eingriffe bei feiner Zeitgranularität	86
Abbildung 51	Ein Tag des Projektleiters	87
Abbildung 52	Ausführungszeit	89
Abbildung 53	Speicherbedarf	89
Abbildung 54	Ausführungszeit und Speicherbedarf im Vergleich	90
Abbildung 55	Wichtige Meilensteine und Termine	93

# Kapitel 1

## Einführung

Diese Diplomarbeit beschäftigt sich mit dem SESAM-Projekt. SESAM steht für Software Engineering Simulation by Animated Models. Mit SESAM werden Modelle für die Ausbildung von (angehenden) Projektleitern verwendet. In dieser Diplomarbeit wird von einem bestehenden Modell eine feingranulare Variante erstellt.

Das Einführungskapitel stellt im ersten Abschnitt kurz das SESAM-Projekt vor. Im zweiten Abschnitt wird das Modell, von dem eine Variante erstellt wird, beschrieben. Anschließend wird die Aufgabenstellung beschrieben und ein Überblick über den Aufbau dieses Berichts gegeben.

### 1.1 Überblick über SESAM

Ein Projektleiter, der ein Softwareprojekt durchführen soll, steht vor einer schwierigen Aufgabe. Der Projektleiter benötigt nicht nur theoretische Kenntnisse über Softwareentwicklung, sondern auch Erfahrung. Die Ausbildung zum Projektleiter steht vor mehreren Problemen: Die notwendige Erfahrung lässt sich nicht theoretisch vermitteln, Praktika sind sehr aufwendig. Die Lehren der Theorie erscheinen trivial und werden erst durch eigene Erfahrung der kritischen Situationen und Probleme als Projektleiter klar. Durch "Learning-by-doing" kann der Projektleiter Erfahrungen sammeln, meist aber zu einem hohen Preis.

Mit dem SESAM-System (Ludewig, 1994) wird daher der Ansatz verfolgt, Projektleiter an einem Simulator auszubilden. Vergleichbar mit der Ausbildung von Piloten an einem Flugsimulator lernen die Projektleiter, ein simuliertes Softwareprojekt erfolgreich durchzuführen. Am Simulator werden die typischen Schwierigkeiten und kritischen Situationen nachgebildet, die bei der Durchführung von Projekten auftreten. Als Spieler führt der Projektleiter simulierte Projekte durch und sammelt dabei Erfahrungen. Damit kann die Theorie deutlich gemacht werden, der Projektleiter kann am Simulator Fehler machen und die Folgen erfahren, ohne dass reale Projekte betroffen sind.

Der Simulator führt Modelle von Software-Projekten aus. Ein Modell enthält alle für die Ausbildung relevanten Aspekte eines Projekts, etwa Dokumente wie die Spezifikation und den Systementwurf, Mitarbeiter des Projekts und den Kunden. Die einzige Ausnahme ist der Projektleiter, diese Rolle übernimmt der Spieler. Abbildung 1 (aus Reißing, 1996a) zeigt diesen Zusammenhang. Der Spieler kommuniziert mit dem Projekt über Kommandos und Nachrichten in eingeschränkter natürlicher Sprache.

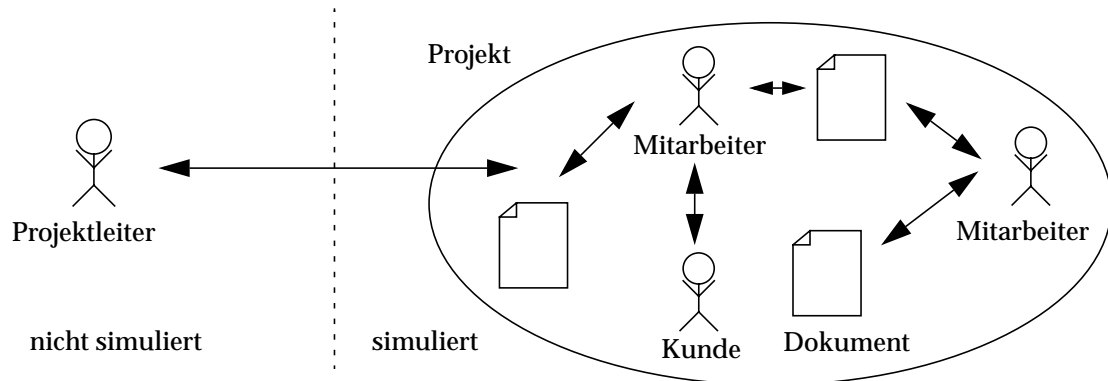


Abb. 1: Projektsimulation mit SESAM

## 1.2 Das QS-Modell

Ein erstes umfassendes Simulationsmodell ist das Qualitätssicherungs-Modell (Drappa, 2000), kurz QS-Modell. Mit diesem Modell konnte gezeigt werden, dass das SESAM-System erfolgreich zur Ausbildung eingesetzt werden kann (Mandl-Striegnitz, 2000). In diesem Abschnitt wird das QS-Modell kurz erklärt.

### 1.2.1 Grundlagen des QS-Modells

Mit dem QS-Modell wird eine bestimmte Projektklasse simuliert, Drappa (2000) grenzt die Projektklasse nach Jones (1996) ein: Es handelt sich um Vertragssoftware, es wird Software für einen Kunden entwickelt. Die zu entwickelnde Applikation ist auf 200 bis 1200 Function Points begrenzt, es werden etwa fünf Entwickler eingesetzt. Die Art der Applikation wird nicht allzu sehr eingeschränkt, das Modell soll vor allem für Informations- und Verwaltungssysteme gültig sein.

Der Schwerpunkt des Modells liegt auf Maßnahmen zur Qualitätssicherung. Der Spieler soll am Simulator erfahren, welche Folgen für das Projekt und die erzielten Resultate eintreten, wenn er diese Maßnahmen einzusetzt oder vernachlässigt. Da sich die Folgen seines Vorgehens erst in den späten Phasen zeigen, wird das Projekt von der Analyse bis zur Übergabe an den Kunden simuliert. Der gesamte Entwicklungsprozess orientiert sich am Wasserfallmodell. Der Spieler kann aber auch einen anderen Ansatz wählen, zum Beispiel das sogenannte "Code-and-Fix"-Vorgehen.

Um ein Projekt durchzuführen, stellt der Spieler Entwickler ein. Er beauftragt dann einen Entwickler mit einer Tätigkeit. Der Spieler kann die Dokumente Spezifikation, Grobentwurf, Feinentwurf, Code und Handbuch erstellen lassen. Der Code wird am Ende des Projekts integriert, anschließend kann der Kunde einen Abnahmetest durchführen. Maßnahmen zur Qualitätssicherung sind die Prüfung eines Dokuments durch Reviews und verschiedene Tests des Codes: Modultest, Integrations- und Systemtest. Nach diesen Maßnahmen kann das geprüfte Dokument korrigiert werden.



Die Entwickler führen die Tätigkeit selbständig durch. Damit ist ausreichend, dass die Simulation auf Tagesbasis erfolgt. Der Spieler kann tageweise in das Projekt eingreifen, der Zustand wird im Modell einmal am Tag aktualisiert, die Zeit in der Simulation kann in ganzen Tagen weitergeschaltet werden.

### 1.2.2 Effekte im QS-Modell

Das Modellverhalten wird durch Annahmen über grundlegende Effekte in Software-Projekten bestimmt (siehe Drappa, 2000):

- Anforderungen des Kunden werden schrittweise top-down von einer höheren Abstraktionsebene auf eine niedrigere Abstraktionsebene übertragen. Sie werden also durch die Phasen des Wasserfallmodells von einem Vorgabedokument in ein Zieldokument übertragen. Dabei werden Fehler eingefügt und Anforderungen unvollständig übertragen.
- Führt ein Entwickler eine Tätigkeit durch, spielen seine Kenntnisse und Erfahrungen eine wesentliche Rolle. Sie wirken sich auf die Dauer und die Ergebnisse der Tätigkeit aus.
- Je mehr Entwickler für eine Tätigkeit eingesetzt werden, desto früher wird die Tätigkeit beendet. Der einzelne Entwickler arbeitet aber langsamer, da der Kommunikationsaufwand größer wird. Damit steigt der Gesamtaufwand.
- Fehler können typisch nur auf der gleichen Abstraktionsebene entdeckt werden, auf der sie eingefügt wurden. Je später ein Fehler entdeckt wird, desto höher sind die Kosten des Fehlers.
- Der Einsatz von Reviews erlaubt, Fehler bereits in den frühen Phasen zu entdecken. Der Korrekturaufwand ist für Reviews geringer als für Tests, da Fehler bereits lokalisiert sind.

### 1.2.3 Reviews im QS-Modell

Am Beispiel von Reviews werden die im QS-Modell realisierten Effekte und die Tätigkeiten des Projektleiters näher beschrieben: Der Spieler kann Reviews für ein Dokument mit zwei oder drei Gutachtern anweisen. Werden die Spezifikation oder das Handbuch geprüft, kann zusätzlich der Kunde eingeladen werden.

Die Prüfung findet dann in verschiedenen Sitzungen statt. Das Dokument wird dafür ohne Eingriffe des Projektleiters in einzelne Arbeitspakete aufgeteilt, jedes Arbeitspaket wird in einer Sitzung geprüft. Die Größe des Arbeitspakets wird automatisch so gewählt, dass die Sitzung nicht länger als zwei Stunden dauert. Die Sitzungen finden immer im Abstand von drei Arbeitstagen statt, damit die Entwickler Zeit für die Vorbereitung der nächsten Sitzung haben.

Abhängig vom Vorgabedokument wird in den geprüften Anforderungen ein Teil der enthaltenen Fehler entdeckt. Verluste, also unvollständig oder nicht im zu prüfenden Dokument enthaltene Anforderungen, werden gefunden. Neben der Qualität des Vorga-

bedokuments und des zu prüfenden Dokuments bestimmt die Anzahl der Gutachter und ihre Erfahrung, wieviel Fehler und Verluste entdeckt werden. Je mehr Gutachter teilnehmen und je erfahrener die Gutachter sind, desto mehr Befunde werden entdeckt. Wird der Kunde eingeladen, findet er zusätzlich Fehler und Verluste. Anforderungen, die ein Gutachter geschrieben hat, werden nicht begutachtet. Wenn eine Sitzung stattgefunden hat, erhält der Projektleiter eine Meldung mit der Anzahl der entdeckten Fehler und den entdeckten Verlusten.

Für die Teilnahme an Reviews verwenden die Gutachter einen Teil ihrer Zeit. Nehmen am Review drei Gutachter teil, muss jeder Gutachter mehr Zeit opfern als bei einem Review mit zwei Gutachtern, da der Kommunikationsaufwand steigt. Vorbereitung und Sitzung sind nicht getrennt modelliert, die nötige Zeit wird auf die zwei Tage vor der Sitzung und den Tag der Sitzung gleichmäßig verteilt. Wird mehr als die Hälfte des Tages für Reviews geopfert, bereitet sich der Gutachter nicht mehr vor.

Die gesamte Prüfung läuft ohne Eingriffe des Spielers ab. Der Spieler kann das Review nur abbrechen.

### 1.3 Aufgabenstellung

Im QS-Modell erfolgt die Simulation auf Tagesbasis. Diese Zeitgranularität von einem Tag reicht für die im QS-Modell realisierten Effekte aus. Die grobe Zeitgranularität schränkt aber zusätzliche, simulierte Effekte ein. Sie erlaubt nicht, dass der Spieler am Simulator erfährt, wie sich die Zeit innerhalb eines Tages ändert. Er kann nur an einem Zeitpunkt des Tages eingreifen. Vorgänge, die in wenigen Stunden an einem Tag ablaufen, können so nur schwer beobachtet und nicht gesteuert werden. Sollen in der Ausbildung feingranulare Effekte gezeigt werden und soll der Spieler einzelne, detaillierte Projekttätigkeiten durchführen, ist deshalb eine feinere Zeitgranularität notwendig.

Eine feinere Zeitgranularität wurde in Modellen zur Ausbildung bisher nicht betrachtet. Daher sollen in der Diplomarbeit zuerst die möglichen Auswirkungen einer feineren Zeitgranularität untersucht werden. Im ersten Schritt werden deshalb Hypothesen über die Auswirkungen einer feineren Zeitgranularität aufgestellt: Welcher Nutzen kann mit einer feineren Zeitgranularität erreicht werden? Welche Probleme werden dann erwartet? Dabei sollen unterschiedliche Aspekte des SESAM-Projekts betrachtet werden. Neben den Auswirkungen auf die Modellierung und die Ausbildung werden dabei auch technische Aspekte der Simulation berücksichtigt.

Diese Hypothesen werden im Verlauf der Arbeit untersucht. Um technische Aspekte und die Auswirkungen auf das QS-Modell zu betrachten, sollen die Phasen Spezifikation und Grobentwurf jeweils mit Review und Korrektur auf eine Zeitgranularität von zwei Stunden umgestellt werden. Die Auswirkungen werden durch Tests der Variante auf Basis von zwei Stunden im Vergleich zum QS-Modell geprüft.

Anschließend soll diese Variante verfeinert und erweitert werden. Der Schwerpunkt liegt auf einer detaillierteren Modellierung des Reviewprozesses. Für dieses feingranulare Modell soll neben dem Modellierungsansatz des QS-Modells auch der Ansatz des strikt atomaren Modells SAM (Eisenbarth, Rohrbach, 1998) betrachtet werden. Das feingranulare Modell soll quantifiziert werden, wenn möglich durch geeignete empirische Daten.

Dieses feingranulare Modell soll in der Hochsprache von SESAM-2 realisiert werden. In diesem Arbeitsabschnitt soll untersucht werden, ob sich die Hochsprache für eine feingranulare Variante eignet und wie ein Modellausschnitt in ein bestehendes Modell eingebettet werden kann.

Letzter Abschnitt der Arbeit ist der Test dieser Erweiterung. Die Quantifizierung des detaillierteren Modells wird gegen das QS-Modell geprüft. Die Hypothesen werden an der feingranularen Variante untersucht.

## **1.4 Überblick über die Arbeit**

Der Aufbau dieses Berichts orientiert sich an der Aufgabenstellung und damit am Vorgehen während der Arbeit. Im Kapitel 2 werden die Grundlagen der Arbeit vorgestellt: Das SESAM-System und die Grundlagen für die Modellierung. Kapitel 3 zeigt die Hypothesen, die im Verlauf der Arbeit untersucht werden. Die Umstellung einzelner Phasen des QS-Modells auf eine Zeitgranularität von zwei Stunden und der anschließende Test werden im Kapitel 4 beschrieben.

Die Konzeption der feingranularen Variante und die Quantifizierung zeigt das Kapitel 5. Die Realisierung folgt in Kapitel 6, die Testergebnisse der feingranularen Variante werden in Kapitel 7 gezeigt.

Die Ergebnisse der Arbeit werden in Kapitel 8 zusammengefasst, dabei wird in einem Rückblick der Projektverlauf kurz beschrieben. In diesem Kapitel befindet sich ein Ausblick, der weitere Möglichkeiten für eine feingranulare Variante zeigt und offene Arbeitspakete beschreibt.

Am Ende des Berichts findet sich das Begriffslexikon und das Literaturverzeichnis.

# Kapitel 2

## Das SESAM-System

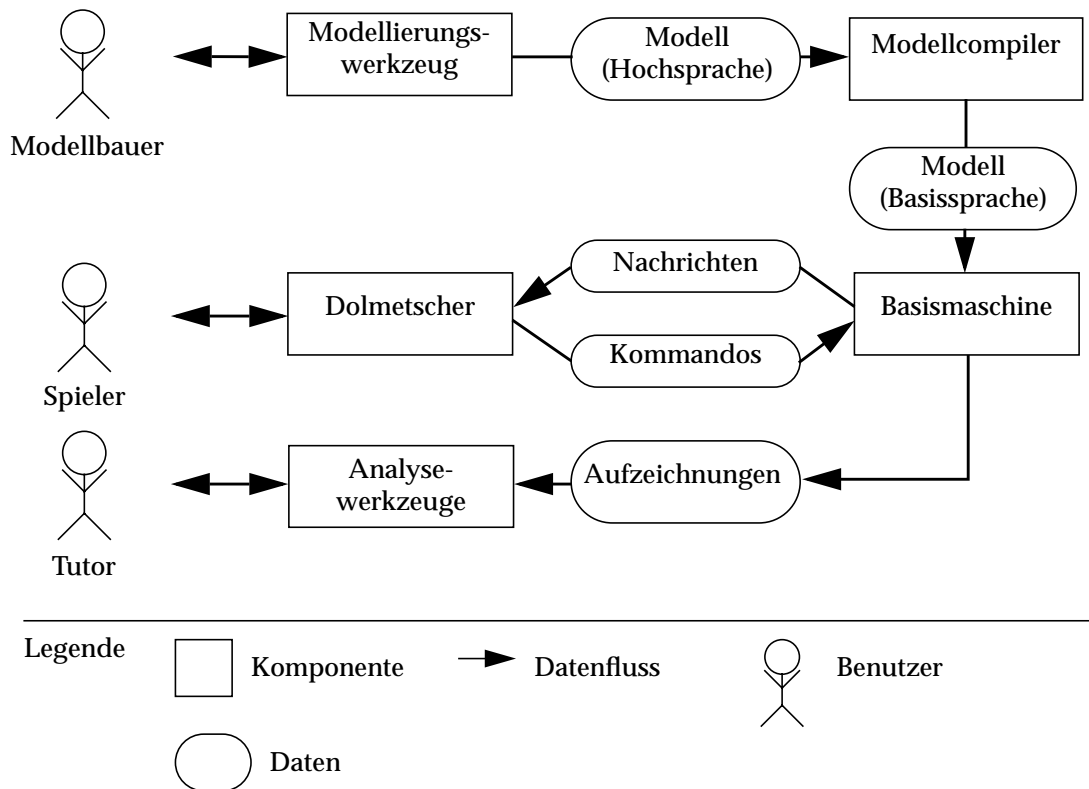
In Kapitel 1 wurde bereits ein kurzer Überblick über das SESAM-System gezeigt. In diesem Kapitel wird das System genauer betrachtet. Der erste Abschnitt zeigt, wie das System aufgebaut ist und wer das System benutzt. Der zweite Abschnitt zeigt, wie die Modelle aufgebaut sind und wie die Modelle vom Simulator ausgeführt werden.

### 2.1 Aufbau und Rollen des Systems

Mit dem SESAM-System werden alle Aspekte eines Software-Projekts mit Ausnahme des Projektleiters simuliert. Damit unterschiedliche Projekte simuliert werden können, führt der Simulator Modelle von Software-Projekten aus. Der Spieler kann so Erfahrung mit unterschiedlichen Projekten sammeln. Mit dem SESAM-Projekt werden zwei Ziele verfolgt: Durch den Einsatz des Simulators wird die Ausbildung zum Projektleiter ergänzt. Durch die Modellierung werden die Zusammenhänge in Software-Projekten erforscht.

Das SESAM-System ist in verschiedene Komponenten aufgeteilt. Diese Komponenten werden von unterschiedlichen Rollen benutzt (Abbildung 2, Seite 7, aus Reißing, 1996a).

- Der Modellbauer erstellt die Modelle. Er entscheidet, welche Objekte und Beziehungen eines Software-Projekts in das Modell abgebildet werden und welche Effekte im Modell auftreten sollen. Entwickler und Dokumente eines Projekts werden beispielsweise durch Entitäten mit den relevanten Eigenschaften dargestellt. Spezifiziert ein Entwickler, kann das durch eine Relation zwischen der Spezifikation und dem Entwickler in das Modell abgebildet werden. Das Modell wird in der Hochsprache des SESAM-Systems erstellt und dann in die ausführbare Basissprache übersetzt.
- Damit der Spieler die Projekte am Simulator leiten kann, führt der Simulator das Modell in der Basissprache aus. Der Spieler greift durch Kommandos in das Projekt ein und erhält Meldungen aus dem System. Die Kommandos können vom Spieler beschränkt natürlichsprachlich eingegeben werden. Der Dolmetscher übersetzt die Anweisungen in Kommandos des Modells. Beispielsweise wird im Modell mit dem Kommando `Lasse_Spezifizieren(Richard)` ein Entwickler zur Spezifikation eingesetzt. Der Simulator gibt daraufhin die Nachricht `Begonnen_zu_spezifizieren` aus, die der Dolmetscher in einen kurzen Text für den Spieler übersetzt.
- Der Tutor führt Schulungen durch, an denen die Spieler teilnehmen. Zu den Schulungen gehört neben anderen Komponenten, dass der Tutor die Spiele der Teilnehmer auswertet und mit den Spielern ihr Vorgehen und die erzielten Resultate diskutiert.



**Abb. 2:** Aufbau und Rollen des SESAM-Systems

## 2.2 SESAM-Modelle

Modelle sind ein wichtiger Teil des Systems. Sie bestimmen, welche Projekte simuliert werden und was geschult werden kann. SESAM-Modelle bauen auf der allgemeinen Modelltheorie auf. Im nächsten Abschnitt werden daher zuerst die Definitionen wichtiger Begriffe gezeigt.

### 2.2.1 Grundlagen der Modellbildung

Deininger (1995) beschreibt die Kernannahme der Modelltheorie und definiert die Begriffe dazu:

Die Welt ist aus Entitäten und Beziehungen zwischen Entitäten aufgebaut. Sowohl Entitäten als auch Beziehungen besitzen Attribute. Attribute beschreiben Eigenschaften von Entitäten und Beziehungen.

Die Begriffe Entität, Beziehung und Attribut sind wie folgt definiert:

**Entität.** Eine Entität (entity) ist ein Element der realen oder der Vorstellungswelt.

**Beziehung.** Eine Beziehung (relationship) verknüpft zwei oder mehr Entitäten. Die von der Beziehung verknüpften Entitäten werden Rollen, die Anzahl der Rollen Stelligkeit der Beziehung genannt.

**Attribut.** Ein Attribut (attribute) ist ein Wert, der eine Eigenschaft einer Entität oder einer Beziehung darstellt.

Entitäten und Relationen können durch gemeinsame Eigenschaften klassifiziert werden. Eine Klasse kann durch gemeinsame Eigenschaften der Mitglieder intensional oder durch die Aufzählung der Mitglieder extensional beschrieben werden.

**Entitätsklasse.** Eine Entitätsklasse (entity-class) ist eine Menge von Entitäten mit gleichen oder ähnlichen Merkmalen. Eine Entität aus einer Klasse wird Exemplar dieser Klasse genannt.

**Beziehungsklasse.** Eine Beziehungsklasse (relationship-class) ist eine Menge von Beziehungen mit gleichen oder ähnlichen Merkmalen. Insbesondere müssen die Beziehungen dieselbe Stelligkeit und die Rollen dieselben Entitätsklassen besitzen.

**Attributklasse.** Eine Attributklasse (attribute-class) ist eine Menge von Attributen mit gleichen oder ähnlichen Merkmalen.

Betrachtet man Entitäten, Relationen und Attribute und ihre Klassen unter einer bestimmten Fragestellung, spricht man von einem System:

**System.** Ein System (system) ist eine Menge von Entitäten und Entitätsklassen, Beziehungen und Beziehungsklassen sowie Attributen und Attributklassen. Die Menge wird unter pragmatischen Gesichtspunkten festgelegt, die Elemente stehen zumindest in einem logischen Zusammenhang.

**Schema.** Ein Schema (scheme) ist ein System aus Entitätsklassen, Beziehungsklassen und Attributklassen.

**Ausprägung.** Eine Ausprägung eines Schemas (instanciation) ist ein System aus Entitäten, Beziehungen und Attributen, wobei ein Schema existieren muss, das die zugehörigen Klassen definiert.

Mit diesen Definitionen kann der Begriff des Modells und des Originals definiert werden:

**Modell und Original.** Ein Modell ist ein System, das ein anderes System – das Original dieses Modells – repräsentiert. Insbesondere ist ein Schema ein Modell seiner Ausprägung.

Ein Modell hat nach Stachowiak (1973) drei Merkmale:

**Abbildungsmerkmal.** Zu einem Modell existiert ein Original. Die Entitäten und Beziehungen des Originals werden auf die Entitäten und Beziehungen des Modells abgebildet.

**Verkürzungsmerkmal.** In das Modell werden nicht alle Entitäten, Beziehungen und Attribute abgebildet, sondern nur die für das Modell relevanten. Das Modell ist verkürzt. Nicht in das Modell abgebildete Entitäten, Beziehungen und Relationen werden präteriert genannt. Entitäten, Beziehungen und Attribute, die nur im Modell vorkommen, nennt man abundant.

**Pragmatisches Merkmal.** Das Modell soll unter bestimmten Bedingungen und Fragestellungen das Original ersetzen. Welche Entitäten, Beziehungen und Attribute in das Modell abgebildet werden und welche verkürzt werden, hängt davon ab, wer das Modell verwendet, wann das Modell verwendet wird und welchen Zweck das Modell erfüllen soll.

### 2.2.2 Struktur der SESAM-Modelle

SESAM-Modelle orientieren sich an der allgemeinen Modelltheorie und bestehen aus drei Komponenten (Schneider, 1994):

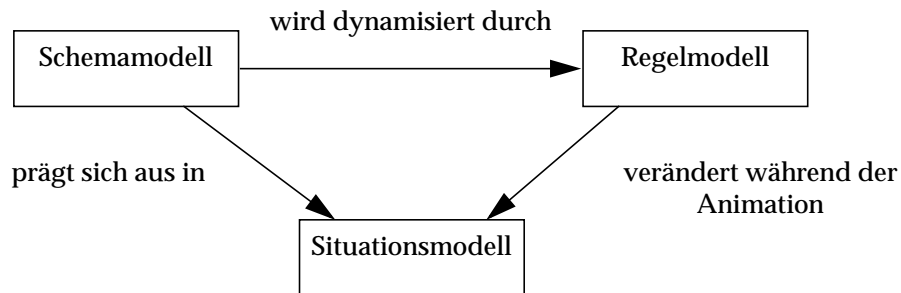
- Das Schemamodell legt fest, welche Objekte und Beziehungen und welche Eigenschaften der Objekte und Beziehungen in das Modell abgebildet werden. Objekte werden durch Entitätsklassen beschrieben. Eigenschaften der Objekte und Beziehungen werden durch Attributklassen beschrieben. Das Schemamodell definiert das Universum, in dem das Projekt abläuft.
- Das Situationsmodell beschreibt den Zustand des Projekts als Ausprägung des Schemamodells. Es besteht aus Entitäten, die durch Relationen verbunden sind und kann als Zustandsgraph dargestellt werden. Die Knoten des Zustandsgraphen sind die Entitäten, die Kanten sind die Relationen zwischen den Entitäten, jeweils mit den konkreten Attributwerten.

Die Startsituation ist das Situationsmodell, das für den Beginn der Simulation festgelegt wird.

- Die Dynamik des Projekts wird durch Effektmodelle in das Regelmodell abgebildet. Schneider (1994) definiert einen Effekt als Klasse von Veränderungen am Projektzustand. Ein Effekt tritt bei einer bestimmten Situation auf und verändert den Projektzustand. Effekte werden in SESAM-Modellen durch Regeln beschrieben. Eine Regel besteht aus zwei Teilen: Im Bedingungsteil wird durch ein Situationsmuster festgelegt, wann die Regel den Zustand verändert. Dazu wird die Regel mit dem Situationsmuster an einen Ausschnitt aus dem Zustandsgraphen gebunden. Im Aktionsteil wird festgelegt, wie die Regel den Zustand verändert. Es können Eigenschaften der Entitäten und Relationen im gebundenen Teil des Zustandsgraphen verändert werden oder neue Entitäten und Relationen erzeugt werden. Kommandos, mit denen

der Projektleiter in das Projekt eingreifen kann und Nachrichten, die der Projektleiter aus dem Projekt erhält, werden im Regelmodell definiert. Kommandos werden wie andere Regeln an einen Teil des Zustandsgraphen gebunden.

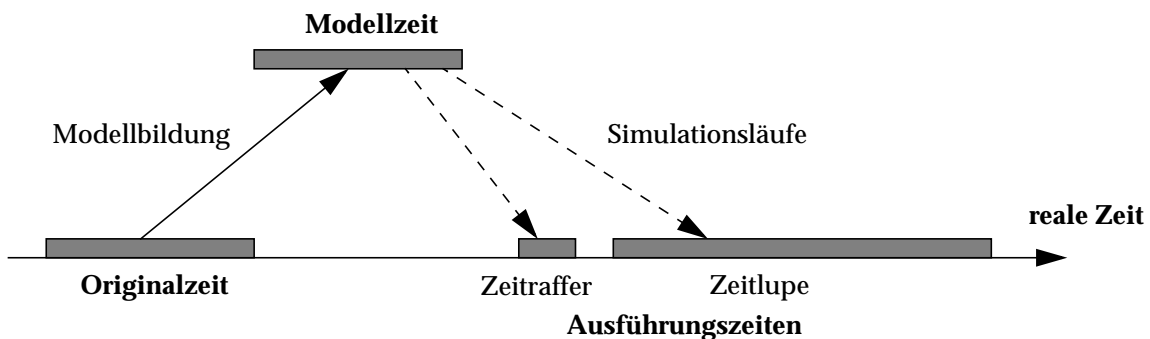
Abbildung 3 zeigt den Zusammenhang zwischen Schemamodell, Situationsmodell und Regelmodell (Schneider, 1994).



**Abb. 3:** Komponenten der SESAM-Modelle

### 2.2.3 Zeitbegriffe in SESAM-Modellen

Ein wichtiges Ziel in Software-Projekten ist, gesetzte Termine einzuhalten. In SESAM-Modellen wird die Zeit eines Projekts, die Originalzeit, auf die Modellzeit abgebildet (Schneider, 1994). Die Modellzeit beginnt bei jedem Simulationslauf von Neuem. Zeitintervalle im Modell sollen möglichst nahe am Original sein, damit der Spieler zeitliche Abläufe in der Ausbildung erfahren kann. Die Ausführungszeit dagegen ist die reale Zeit bei der Ausführung des Modells. Die Modellzeit wird durch die Ausführungszeit gestaucht (Zeitraffer) oder gedehnt (Zeitlupe). Abbildung 4 zeigt die Zusammenhänge.



**Abb. 4:** Zeitbegriffe in SESAM-Modellen

Der Zeitverbrauch des Projektleiters wird durch den Zeitverbrauch einzelner Tätigkeiten in das Modell abgebildet. Soll ein Entwickler spezifizieren, braucht der Projektleiter beispielsweise 30 Minuten, um mit dem Entwickler zu sprechen. Der Zeitverbrauch kann in Regeln oder Kommandos angegeben werden.



### **2.2.4 Ausführung von SESAM-Modelle**

Die Ausführung des Modells durch den Simulator ist zeitabhängig. Das Modell wird in einzelnen Zeitschritten ausgeführt, der Abstand der Zeitschritte ist konstant und wird als Simulationsschrittweite bezeichnet. Es handelt sich also um eine diskrete, zeitgesteuerte Simulation (Drappa, 2000, Seite 24). In jedem Zeitschritt prüft die Basismaschine, welche Regeln im Situationsmodell gebunden werden können. Wird eine Regel gebunden, wird das Situationsmodell durch die Regel verändert. Der Simulationsschritt ist beendet, wenn keine Regel mehr aus dem Regelmodell gebunden werden kann. Zum Schluss wird die Modellzeit um die Dauer eines Zeitschritts weitergesetzt.

Der nächste Simulationsschritt beginnt, wenn der Spieler die Modellzeit weiterschaltet oder durch Kommandos oder Regeln einen oder mehrere Zeitschritte verbraucht hat. Der Simulator führt so lange Simulationsschritte durch, bis die Modellzeit mindestens um die verbrauchte Zeit weitergeschaltet wurde.

### **2.2.5 Kontinuierliche und diskrete Veränderungen des Situationsmodells**

Die Basissprache wird zur Ausführung des Modells verwendet. Die Hochsprache wird vom Modellbauer benutzt und bietet daher mehr Konstrukte, um die Erstellung des Modells zu erleichtern. Sie bietet unter anderem zwei unterschiedliche Typen von Regeln, auf die in diesem Abschnitt näher eingegangen werden soll.

Für diskrete Veränderungen des Situationsmodells verwendet der Modellbauer Feuerungsregeln. Die Regeln bestehen aus einem Bedingungsteil und einem Aktionsteil. Der Bedingungsteil legt fest, an welches Situationsmuster die Regel gebunden wird. Kann eine Feuerungsregel an ein Situationsmuster gebunden werden, dann feuert die Regel. Sie verändert das Situationsmodell diskret durch den Aktionsteil.

Aktivitätsregeln ermöglichen durch ein Aktivierungs- und Deaktivierungskonzept, kontinuierliche Veränderungen über einen längeren Zeitraum zu modellieren. Sie bestehen aus einem Bedingungsteil, einem Aktivierungsteil, einem Aktivteil und einem Deaktivierungsteil. Wie bei Feuerungsregeln legt der Bedingungsteil fest, wann die Aktivität an ein Situationsmuster gebunden werden kann. Wird die Aktivität zum ersten Mal gebunden, wird der Aktivierungsteil und danach der Aktivteil durchgeführt. Solange der Bedingungsteil in den folgenden Zeitschritten erfüllt ist, wird der Aktivteil ausgeführt. Sobald der Bedingungsteil nicht mehr erfüllt ist, wird der Deaktivierungsteil ausgeführt und damit die Aktivität deaktiviert.

Mit Aktivitäten kann das Regelmodell strukturiert werden. Aktivitäten können weitere Aktivitäten oder Feuerungsregeln enthalten. Die eingeschachtelten Regeln werden nur gebunden, wenn die Aktivität aktiv ist. Der Bedingungsteil eingeschachtelter Regeln wird um den Bedingungsteil der Aktivität erweitert.

## Kapitel 3

# Hypothesen über den Nutzen und die Probleme einer feineren Zeitgranularität

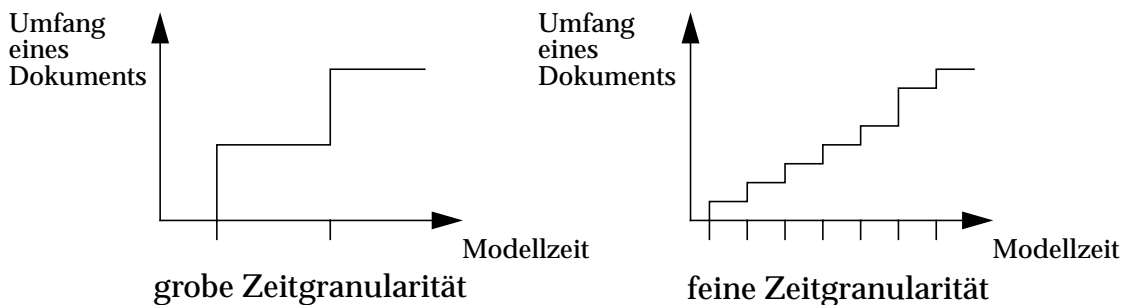
In diesem Kapitel werden die vermuteten Auswirkungen einer feineren Zeitgranularität gezeigt. Im ersten Abschnitt wird beschrieben, welche Zusammenhänge zwischen der Modellgranularität, der Zeitgranularität und den möglichen Eingriffen des Spielers bestehen. In den darauf folgenden Abschnitten werden die aufgestellten Hypothesen gezeigt. Sie beschreiben mögliche Auswirkungen einer feineren Zeitgranularität auf die Modellierung, auf die Ausbildung mit SESAM-Modellen und auf technische Aspekte des SESAM-Systems.

### 3.1 Die verschiedenen Arten der Granularität

#### 3.1.1 Zeit- und Modellgranularität

Führt der Simulator ein Modell aus, wird der Zustand in jedem Zeitschritt neu berechnet und dadurch definiert. Der Abstand in der Modellzeit zwischen den definierten Zuständen ist die Simulationsschrittweite. Sie wird vom Modellbauer in der Startsituation festgelegt. Der Modellbauer legt die Simulationsschrittweite durch die **Zeitgranularität** fest: Sie bestimmt, wann und in welchem Abstand der Projektzustand im Modell aktualisiert wird und legt fest, wann Veränderungen im Projekt in Abhängigkeit vom Ablauf in den Zustand abgebildet werden. Damit bestimmt die Zeitgranularität, wie fein die Dynamik des Originals (ein Software-Projekt) in das Modell abgebildet werden soll.

Der Modellbauer kann Veränderungen am Zustand eines Projekts etwa im Abstand von einer Woche, einem Tag oder zwei Stunden modellieren. Arbeitet zum Beispiel ein Entwickler an einem Dokument, verändert sich der Umfang des Dokuments in jedem Zeitschritt abhängig von der Schrittweite.



**Abb. 5:** Veränderungen abhängig von der Zeitgranularität

Die **Modellgranularität** dagegen legt die Abstraktionsebene fest, mit der das Original in das Modell abgebildet wird. Je nach Modellgranularität kann der Modellbauer zum Beispiel Reviews eines Dokuments insgesamt modellieren oder einzelne Sitzungen, die Vorbereitung auf die Sitzung durch Gutachter und die Nachbereitung getrennt betrachten.

Modellgranularität und Zeitgranularität sind unabhängig voneinander.

- Der Modellbauer kann die Modellierung sehr feingranular durchführen und entsprechend implementieren, obwohl eine grobe Zeitgranularität gewählt wurde. Er kann auch bei einer großen Schrittweite feine Veränderungen, etwa Effekte einer kurzen Tätigkeit, modellieren. Sitzungen, die nur eine Stunde dauern, können beispielsweise auch bei einer Schrittweite von einem Tag modelliert werden.

Eine feine Modellgranularität ohne eine angepasste Zeitgranularität hat aber nur wenig Nutzen. Der Spieler nimmt die feinen Veränderungen wie die Dauer und die Ergebnisse einer kurzen Tätigkeit nur schwer wahr. Die Auswirkungen auf den Zustand werden erst spät sichtbar. Es finden viele Effekte in einem Zeitschritt statt, der Tutor kann nur die Veränderungen insgesamt betrachten. Ein weiterer Nachteil ist, dass die zeitliche Reihenfolge von feingranularen Effekten nicht mehr von der Modellzeit abhängt, sondern durch die Modellierung erzwungen werden muss.

- Umgekehrt kann ein Modell auch mit grober Modellgranularität und feiner Zeitgranularität erstellt werden. Werden aber nur ganze Phasen modelliert, die mehrere Wochen dauern, sind Zustandsänderungen im Abstand von wenigen Stunden weder für den Spieler noch für den Tutor interessant.

Zeit- und Modellgranularität sollten also aufeinander abgestimmt sein. Die Granularitäten sollten ähnlich fein sein.

### 3.1.2 Eingriffe des Spielers

Die Eingriffsmöglichkeiten des Spielers sind abhängig von der Zeitgranularität und von der Modellgranularität. Der Spieler kann nur so fein in das Projekt eingreifen, wie es Zeit- und Modellgranularität zulassen. Dabei bestimmt die Zeitgranularität, wann und in welchem Abstand der Spieler eingreifen kann. Je kürzer der Zeitschritt ist, desto häufiger kann der Spieler in das Projekt eingreifen und desto schneller kann er auf Veränderungen reagieren.

Die Modellgranularität bestimmt, welche Aktionen der Spieler durchführen kann. Werden beispielsweise ganze Phasen modelliert, kann der Spieler nicht in einzelne Aktivitäten eingreifen, die während der Phase stattfinden: Die einzelnen Aktivitäten werden bei einer groben Modellgranularität nicht in das Modell abgebildet. Eine feine Zeit- und Modellgranularität ist also eine notwendige Voraussetzung, wenn der Spieler das Projekt detaillierter steuern soll. Zusätzlich müssen entsprechende detaillierte Kommandos und Nachrichten vom Modell bereitgestellt werden.

Es ist aber möglich, dass der Spieler auch bei einer feinen Modell- und Zeitgranularität nur grob in das Projekt eingreifen kann und nur wenige Nachrichten erhält. Eine feinere Granularität erzwingt daher nicht, dass der Spieler detaillierter in das Projekt eingreifen muss.

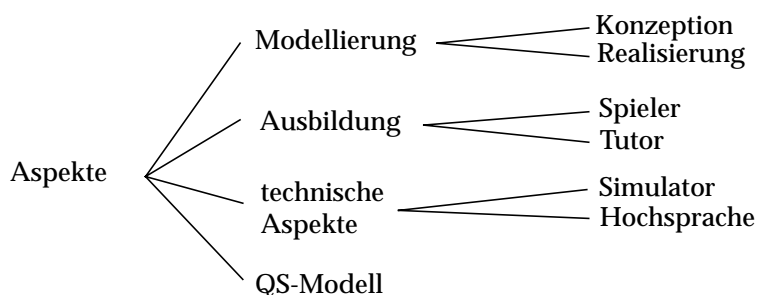
### 3.1.3 Granularität und Ausbildungsziele

Zeitgranularität, Modellgranularität und die Eingriffsmöglichkeiten des Spielers sind vom Zweck des Modells abhängig: Das Ausbildungsziel bestimmt, wie feingranular die Modellierung durchgeführt werden sollte, welche Veränderungen des Zustandes beobachtet werden müssen und welche Aktionen der Spieler durchführen soll. Der Spieler kann nur die Effekte des Modells erfahren, die er beobachten kann. Er muss ausreichend fein in das Projekt eingreifen können. Für die Schulung muss der Tutor die Effekte analysieren können und die Auswirkungen aus dem Vorgehen des Spielers beobachten können. Sollen daher in einer Schulung Auswirkungen von feingranularen Effekten gezeigt werden, ist eine feine Zeit- und Modellgranularität notwendig.

Ist zum Beispiel ein Ziel der Ausbildung, dass die Durchführung von Reviews den Aufwand und die Kosten des Projektes senkt, muss nur festgestellt werden, ob und wann ein Review durchgeführt wurde. Gehört aber zum Ausbildungsziel, dass der Spieler lernt, wie Reviews durchgeführt werden sollen, muss er entsprechend in das Projekt eingreifen und die Folgen seines Vorgehens direkt beobachten können. Er muss dann kontrollieren können, wann und wie lange sich ein Entwickler vorbereitet.

## 3.2 Überblick über die Hypothesen

Ausgehend von den Zusammenhängen zwischen Modell- und Zeitgranularität können Auswirkungen über den Nutzen und die Probleme einer feineren Zeitgranularität vermutet werden. Diese Auswirkungen werden als Hypothesen formuliert und werden durch die verschiedenen Aspekte des SESAM-Systems gegliedert. Hypothesen, die nur für das QS-Modell gelten, sind getrennt aufgeführt. Damit ergibt sich die in Abbildung 6 dargestellte Gliederung. Auswirkungen, die Forderungen als Folge einer feineren Zeitgranularität beschreiben, sind mit den Hypothesen in diese Gliederung eingeordnet.



**Abb. 6:** Gliederung der Hypothesen

Bei Auswirkungen, die nur im Zusammenhang mit einer feinen Zeitgranularität vermutet werden, wird auf eine feinere Zeitgranularität verwiesen. Der größte Teil der Hypothesen gilt aber für eine feine Zeitgranularität in Verbindung mit einer feineren Modellgranularität – die Hypothesen beziehen sich auf eine feinere Granularität.

Die verschiedenen Hypothesen sind nach vermutetem Nutzen und Problemen getrennt aufgeführt und durch den Buchstaben H gekennzeichnet. Wird eine Hypothese nur für das QS-Modell aufgestellt, wird sie zur Unterscheidung mit einem Q dargestellt. Forderungen oder Fakten werden mit einem F gekennzeichnet.

### **3.3 Auswirkungen auf die Modellierung**

Die Modellierung wird vom Modellbauer durchgeführt. Für die Konzeption werden die Ausbildungsziele festgelegt. Damit kann der Modellbauer die Effekte identifizieren, die im Modell umgesetzt werden sollen. Zur Konzeption gehört die Auswahl geeigneter Metriken und die Quantifizierung des Modells. Für die Realisierung wird der Entwurf erstellt und das Modell implementiert.

#### **3.3.1 Nutzen bei der Konzeption eines Modells**

##### **Diskrete Simulation**

Beim SESAM-System handelt es sich um ein zeitdiskretes Simulationsmodell (Drappa, 2000, Seite 35). Zustände sind nur zu bestimmten Zeitpunkten definiert und messbar. Werden kontinuierliche Vorgänge aus der Realität in das Modell abgebildet, nähert sich das diskrete Modell bei kürzerer Schrittweite stärker an den realen Verlauf des Vorgangs an. Daraus folgt die erste Hypothese.

**H 1:** Mit einer feineren Zeitgranularität nähert sich das System an ein zeitkontinuierliches System an (Quasi-kontinuierliche Simulation). Die Simulation wird realitätsnäher.

Kontinuierliche Vorgänge in der Realität können damit auf quasikontinuierliche Vorgänge in der Simulation abgebildet werden. Die Simulation bleibt aber diskret.

##### **Verfeinerung eines Modells**

Im Abschnitt 3.1 ist diskutiert, unter welchen Voraussetzungen eine feinere Zeitgranularität einen Nutzen hat. Dabei wird festgestellt, dass eine feinere Modellgranularität für den Tutor oder den Spieler erst erfahrbar wird, wenn die Simulationsschrittweite auf das feingranulare Modell abgestimmt ist.

Wird ein bestehendes Modell auf eine feinere Zeitgranularität umgestellt, können daher die folgenden Auswirkungen vermutet werden:

**H 2:** Bereits feingranular modellierte Effekte oder Ereignisse werden für den Spieler oder den Tutor einer Schulung beobachtbar.

**H 3:** Der Modellbauer kann zusätzliche feingranulare Effekte modellieren. Diese Effekte können dann vom Tutor oder vom Spieler direkt beobachtet werden.

Damit kann der Modellbauer Varianten eines Modells mit unterschiedlicher Granularität erstellen. Diese können dann je nach Ausbildungsinhalt einer Schulung vom Tutor eingesetzt werden. Der Modellbauer kann bei der Bildung der Varianten unterschiedlich vorgehen:

- Um den Schwerpunkt des Modells auf bestimmte Phasen eines Projekts zu legen, können einzelne Phasen detailliert modelliert werden. Beispielsweise kann die Durchführung von Reviews gezielt geschult werden, etwa mit einzeln anzusetzenden Sitzungen, für die sich die Gutachter einzeln vorbereiten.
- Er kann das gesamte Modell um feingranulare Effekte erweitern, etwa um Störungen des Projektleiters oder Überstunden der Mitarbeiter.

Die beiden Vorgehensweisen können kombiniert werden. Die Folgen für die Ausbildung mit einer feingranularen Modellvariante werden in Abschnitt 3.4 als Hypothesen beschrieben.

### **Modellierung der Entwickler**

Im QS-Modell können Entwickler nur begrenzt mehrere Aufgaben wahrnehmen. Nur die Teilnahme an Reviews ist parallel zu einer anderen Tätigkeit möglich. Ein Entwickler kann nicht gleichzeitig an mehreren Dokumenten arbeiten. Wie im Abschnitt 1.2.1 beschrieben, reicht diese Modellierung der Entwickler aus – der Ausbildungszweck legt den Schwerpunkt auf den Einsatz von Qualitätssicherungsmaßnahmen.

In der Realität können Entwickler mehrere Aufgaben gleichzeitig wahrnehmen. Sie entscheiden mehrmals am Tag, welche Aufgabe wichtig ist und welche Aufgabe liegenbleibt. Muss etwa ein Dokument dringend fertiggestellt werden, vernachlässigen Entwickler andere Arbeiten. Bei einer Schrittweite von einem Tag kann der Zeitschritt fest für die Tätigkeiten aufgeteilt werden. Eine Änderung am Tagesablauf ist damit aber nicht mehr möglich, es kann nicht beobachtet werden, wann der Entwickler welche Tätigkeit durchführt. Bei einer Schrittweite von zwei Stunden dagegen kann sich der Entwickler realistischer verhalten. Das Verhalten kann beobachtet werden.

**H 4:** Das Verhalten der Entwickler lässt sich realistischer modellieren.

Im SAM (Eisenbarth, Rohrbach, 1998) wurden Entwickler feingranularer modelliert, sie entscheiden sich jede Stunde, welche Aufgabe durchgeführt werden soll.

Wird das Verhalten der Entwickler mit einer feineren Zeitgranularität modelliert, können die Entwickler innerhalb eines Tages flexibel reagieren. Die Entwickler können dann

vom Spieler feiner gesteuert werden. Der Spieler kann mehrmals am Tag eingreifen, etwa um kurzfristig auf die Erledigung einer Aufgabe zu drängen.

Wichtig für eine realistische Modellierung der Entwickler ist, wann ein Entwickler arbeitet. Mit einer feinen Zeitgranularität kann beobachtet werden, wann ein Entwickler mit der Arbeit anfängt und aufhört.

**H 4.1:** Die Arbeitszeit der Entwickler kann realistischer modelliert werden.

Damit lassen sich auch Überstunden modellieren, die der Projektleiter anordnen kann.

### **3.3.2 Probleme bei der Konzeption eines feingranularen Modells**

#### **Komplexität des Modells**

Das QS-Modell ist bereits komplex. Wird feingranularer modelliert, dann werden mehr und feinere Effekte in das Modell abgebildet. Sie überlagern sich gegenseitig, so dass der Modellbauer die Ursachen für Veränderungen des Zustandes nur schwer erkennen kann. Der Modellbauer muss diese Effekte trotzdem überschauen können.

**H 5:** Wenn feingranular modelliert wird, steigt die Komplexität des Modells. Das Modell kann für den Modellbauer schlecht überschaubar werden.

Die Komplexität des Modells hat auch Folgen für die Realisierung. Das Schema- und Regelmodell wird größer, das Situationsmodell wird komplexer.

**H 6:** Dynamik und Zustand des Modells werden komplexer und schwerer zu durchschauen. Die Wartung, vor allem Fehlersuche und -behebung, wird erschwert.

#### **Abstaktionsstufen der Modellierung**

Der Modellbauer legt fest, wie feingranular Software-Projekte in das Modell abgebildet werden. Erstellt ein Entwickler im QS-Modell ein Dokument, wird dieser Vorgang als eine einzige Tätigkeit modelliert. Im SAM dagegen zerfällt die Tätigkeit in einzelne Schritte, der Autor liest die Vorgabe, spricht mit einem Entwickler oder schreibt das Dokument. Die Modelle liegen damit auf unterschiedlichen Abstraktionsstufen.

Werden einzelne Phasen detaillierter modelliert (Hypothese H 3 auf Seite 16), um einzelne Phasen zu schulen, ändert sich die Modellgranularität und damit die Abstraktionsstufe innerhalb des Modells. Schneider (1994, Seite 175) betrachtet diesen Ansatz kritisch. Einzelne Effekte können mehrfach modelliert werden oder sich gegenseitig überlappen und beeinflussen. Für den Spieler ergeben sich ebenfalls Probleme (Hypothese H 17, Seite 23).

**F 1:** Der Modellbauer muss beachten, dass er verschiedene Phasen nicht auf unterschiedlichen Abstraktionsstufen modelliert.

Das Modell sollte daher vollständig auf eine feinere Modellgranularität umgestellt werden. Oder es werden einzelne Phasen aus dem Modell herausgelöst und verfeinert. In

dieser Diplomarbeit werden einzelne Phasen verfeinert - da der Zeitrahmen begrenzt ist, kann nicht das ganze Modell verfeinert werden.

### **Quantifizierung und Validierung**

Quellen mit repräsentativen Daten über Software-Projekte sind rar. Beim QS-Modell gab es bereits Schwierigkeiten, geeignete Daten zu finden (Joos, 1997, Seite 32). Werden einzelne Phasen feingranularer modelliert, können vermutlich noch weniger Daten gefunden werden. Beim SAM spricht Eisenbarth und Rohrbach (1998) eher von einer Qualifizierung als von einer Quantifizierung.

In der Literatur werden meist Durchschnittswerte auf einer abstrakten Ebene angegeben, etwa der Aufwand für Spezifikationsreviews in einer bestimmten Projektklasse. Für ein detailliertes Reviewmodell muss der Aufwand auf die Vorbereitung einzelner Gutachter und die Sitzung verteilt werden. Insgesamt soll aber der Durchschnittswert erreicht werden. Reviews sind gut untersucht, soll aber der Test in Vorbereitung, Fehlerentdeckung, Fehlersuche und Fehlerbehebung aufgeteilt werden, sind vermutlich kaum noch Daten in der Literatur vorhanden.

**H 7:** Bei einer feineren Modellierung von Effekten wird die Quantifizierung schwieriger.

Eine mögliche Validierung ist die Prüfung gegen ein vorhandenes, bereits validiertes Modell oder gegen Daten aus der Literatur. Eine genaue Validierung der einzelnen detaillierten Vorgänge im Modell wird kaum möglich sein.

### **3.3.3 Weitere Auswirkungen auf die Konzeption eines feingranularen Modells**

#### **Variabilität**

Die Variabilität des Modells bestimmt, welchen Handlungsspielraum ein Spieler hat (Schneider, 1994, Seite 75). So kann der Spieler im QS-Modell nicht nur nach dem Wasserfallmodell vorgehen, er kann das Vorgehen so weit verkürzen, dass ein Code-and-Fix-Ansatz simuliert wird oder dass sich die einzelnen Tätigkeiten überlappen. Den Abstand einzelner Reviewsitzungen kann er aber nicht bestimmen. Bei einer feineren Modellgranularität sind mehr und unterschiedlichere Vorgehensweisen möglich, die der Modellbauer berücksichtigen muss, beispielsweise dass die Zeit zwischen den Sitzungen für die Vorbereitung der Gutachter ausreicht.

**H 8:** Der Modellbauer muss Effekte, die aus einem unterschiedlichen Vorgehen resultieren können, berücksichtigen. Die Variabilität des Modells wird größer.

Der Vorteil ist, dass der Spieler mehr Freiheit bei der Leitung des Projekts hat. Der Nachteil für den Modellbauer ist aber, dass er berücksichtigen muss, welche Folgen das Vorgehen haben kann. Damit müssen mehr Effekte modelliert werden, das Modell wird komplexer.



### 3.3.4 Probleme bei der Implementierung eines feingranularen Modells

#### Arbeitszeit der Entwickler und des Projektleiters

In der Realität wird typisch weder am Wochenende noch nachts gearbeitet. Es wäre unrealistisch, wenn die Entwickler außerhalb der üblichen Arbeitszeit weiter arbeiten würden. Im QS-Modell beispielsweise haben die Entwickler pro Tag acht Stunden Zeit, um eine Tätigkeit durchzuführen. Am Wochenende wird diese verfügbare Zeit auf Null gesetzt, so dass die Entwickler nicht arbeiten können.

Ähnliches gilt für den Projektleiter. Er darf weder nachts noch am Wochenende in das Projekt eingreifen können. Damit der Spieler keine Kommandos absetzen kann, muss die Projektleiterzeit verbraucht werden. Im QS-Modell beispielsweise wird am Wochenende die Projektleiterzeit von zwei Tagen verbraucht.

**F 2:** Der Modellbauer muss verhindern, dass die Entwickler oder der Projektleiter nachts oder am Wochenende arbeiten.

### 3.3.5 Weitere Auswirkungen auf die Implementierung eines Modells

#### Aufbau des Modells

Implementiert der Modellbauer die im konzeptionellen Teil der Modellierung festgelegten Effekte, kann er auf zwei unterschiedliche Konstrukte der Hochsprache zurückgreifen (siehe auch Abschnitt 2.2.5 auf Seite 11): Diskrete Änderungen des Zustands werden mit Feuerungsregeln implementiert, für kontinuierliche Änderungen des Zustandes über einen längeren Zeitraum werden Aktivitätsregeln verwendet.

In einer feingranularen Variante werden die Tätigkeiten eines Entwicklers detaillierter betrachtet. Führt ein Entwickler eine Tätigkeit durch, setzt sich diese Tätigkeit aus verschiedenen Teilaufgaben zusammen. Im QS-Modell werden Reviews eines Dokuments durch eine Aktivität modelliert. Feingranular setzen sich Reviews aus der Vorbereitung der Gutachter und der Sitzung zusammen. Die Vorbereitung dauert nur wenige Stunden und kann damit nicht mehr als Aktivität implementiert werden.

**H 9:** Wird das Modell feingranularer, werden weniger kontinuierliche Veränderungen modelliert. Es werden mehr Regeln mit Feuerungssemantik implementiert.

## 3.4 Auswirkungen auf die Ausbildung

Wie in Kapitel 1 kurz beschrieben, führen die Teilnehmer einer Schulung ein am Rechner simuliertes Projekt durch. Als Spieler übernehmen die Teilnehmer dabei die Rolle des Projektleiters. Alle übrigen Aspekte eines Software-Entwicklungsprozesses werden durch das SESAM-System simuliert.

Die Simulation ist in eine Schulung eingebunden, damit ein Lerneffekt der Teilnehmer erzielt wird. Wichtiger Bestandteil der Schulung ist eine Analyserunde. Der Tutor, der

die Schulung leitet, stellt in der Analyserunde das Vorgehen und die erzielten Ergebnisse der Projektsimulationen vor. Er diskutiert die Ergebnisse und das Vorgehen mit den Spielern. Für diese Analyserunde muss der Tutor die Spiele der Teilnehmer auswerten.

### **3.4.1 Nutzen für den Spieler**

#### **Planung**

Der Spieler soll wie ein realer Projektleiter das simulierte Projekt zuerst planen. Es werden beispielsweise Dauer und Aufwand einzelner Maßnahmen geschätzt, Mitarbeiter einzelnen Tätigkeiten zugeordnet und Termine für Beginn und Ende einzelner Maßnahmen festgelegt.

Bei einer feingranularen Variante muss der Projektleiter nicht nur ganze Maßnahmen, sondern zusätzliche Teilaufgaben innerhalb dieser Maßnahmen im Plan berücksichtigen. Es reicht beispielsweise nicht, nur die Durchführung eines Spezifikationsreviews und die nötigen Gutachter einzuplanen. Es muss die Vorbereitung für die Mitarbeiter, die Termine einzelner Sitzungen und die Zeit für die Nachbereitung eingeplant werden.

**H 10:** Der Spieler muss bei einer feingranularen Variante detaillierter planen. Das Spiel wird dadurch realitätsnäher.

#### **Aktionen des Spielers**

Im Spiel weist der Spieler den Entwicklern einzelne Tätigkeiten zu. Soll ein Entwickler spezifizieren, so gibt der Spieler einen entsprechenden Text ein. Die Entwickler melden, wenn Sie mit einer Tätigkeit beginnen und wenn sie die Tätigkeit beenden.

Der Simulator muss erst einen Simulationsschritt durchführen, damit der Zustand an die Aufforderungen des Spielers angepasst wird. Beendet ein Entwickler eine Tätigkeit, wird die entsprechende Meldung nicht im selben Zeitschritt ausgegeben, in dem der Entwickler fertig wird, sondern erst im nächsten Zeitschritt. Der Spieler kann erst wieder im darauf folgenden Zeitschritt eingreifen. Bei einer Schrittweite von einem Tag geht daher mindestens ein Tag verloren.

Um die Folgen einer Anweisung am Simulator zu sehen, muss der Spieler den nächsten Zeitschritt abwarten oder sicherheitshalber die Modellzeit fortschalten. Dabei verliert er erneut einen Tag. Je kürzer der Zeitschritt, desto früher bemerkt daher der Spieler die Folgen einer Aktion.

**H 11:** Bei feinerer Zeitgranularität erhält der Spieler Rückmeldungen des Systems zu einem früheren Zeitpunkt.

**H 12:** Der Spieler kann bei einer feineren Zeitgranularität früher eingreifen, da der Zustand des Simulators häufiger angepasst wird.

Für eine feingranulare Modellierung wurden die Hypothesen aufgestellt, dass das Verhalten der Entwickler realistischer modelliert werden kann (Hypothese H 4, Seite 16)

und dass einzelne Phasen detaillierter modelliert werden können (Hypothese H 3, Seite 16). Damit kann der Spieler feiner in das Projekt eingreifen:

**H 13:** Der Spieler kann das Verhalten der Entwickler in einer feingranularen Variante gezielter steuern und beobachten. Er kann dadurch feingranularere Entscheidungen über den Projektverlauf fällen.

Wenn der Spieler das Verhalten der Entwickler gezielt für einzelne Tätigkeiten steuert, kann er unterschiedlich vorgehen: Zum Beispiel können Reviewsitzungen in einem größeren Abstand stattfinden, die Gutachter können sich den ganzen Tag auf eine Review vorbereiten oder nur wenig Zeit pro Tag aufwenden. Je nach Ausbildungszweck des Modells kann eine so feingranulare Steuerung der Entwickler sinnvoll sein. Beispielsweise soll der Spieler lernen, dass genügend Zeit zur Vorbereitung reserviert werden muss, damit das Review erfolgreich durchgeführt werden kann.

Die Einflussmöglichkeiten des Projektleiters werden durch den Modellbauer vorgegeben. Er entscheidet, welche Kommandos möglich sind und wie die Entwickler die Aufgaben durchführen.

### **Tagesablauf des Projektleiters**

Ein Projektleiter erledigt unterschiedliche Aufgaben: Er weist Mitarbeiter an, kontrolliert den Fortschritt, beantwortet Fragen von Mitarbeitern oder plant nach Verzögerungen neu. Die verschiedenen Tätigkeiten des Projektleiters verbrauchen Zeit. Dieser Effekt soll auch in den simulierten Projekten auftreten (Drappa, 2000, Seite 31). Der Zeitverbrauch der Kommandos liegt zwischen wenigen Minuten und einigen Stunden. Wird die Dauer eines Zeitschritts verbraucht, schaltet der Simulator weiter. Bei einer grobgranularen Variante wie dem QS-Modell kann der Spieler nur schlecht erkennen, welche Tätigkeit wieviel Zeit in Anspruch nimmt, da der Simulator erst weiterschaltet, wenn acht Stunden verbraucht werden.

Bei feiner Zeitgranularität auf Basis von zwei Stunden kann der Spieler beobachten, wie die Zeit innerhalb eines Tages vergeht, da der Simulator bereits nach einem Zeitverbrauch von zwei Stunden weiterschaltet und die Modellzeit verändert. Der Spieler bemerkt also bei feiner Zeitgranularität den Zeitverbrauch seiner eigenen Tätigkeiten zur Führung der Mitarbeiter und zur Kontrolle des Projekts.

Nicht nur die Tätigkeiten verbrauchen Zeit. Beispielsweise melden Mitarbeiter das Ende einer Tätigkeit und verbrauchen damit die Zeit des Projektleiters. Wie beim Zeitverbrauch der Kommandos kann auch hier der Spieler bei einer Schrittweite von zwei Stunden den Einfluss auf seinen Tagesablauf bemerken, da der Simulator bereits nach zwei Stunden weiterschaltet und die Modellzeit verändert. Es ist daher in einer feingranularen Variante sinnvoller, diese Störungen zu modellieren. Treten viele Störungen auf, wird der Spieler unterbrochen. Während den Störungen kann er nicht mehr in das Projekt eingreifen. Damit bemerkt der Spieler den Einfluss von Störungen auf den Verlauf des Tages und kann diesen Einfluss als Lerneffekt auf die Realität übertragen.

Zusammenfassend kann diese Hypothese aufgestellt werden:

**H 14:** Der Spieler nimmt bei feinerer Zeitgranularität den Tagesablauf eines Projektleiters stärker wahr.

### 3.4.2 Probleme für den Spieler

#### Fortschalten der Modellzeit

Im Spielverlauf kann es Modellzeiträume geben, die für den Spieler nur wenig interessant sind. Der Spieler wartet etwa, bis ein Entwickler die Spezifikation zu Ende geschrieben hat, um die Spezifikation durch Reviews zu prüfen. Diese Zeiträume kürzt der Spieler ab, in dem er die Modellzeit am Simulator um einen oder mehrere Zeitschritte weiterschaltet. Wird ein Modell auf eine kürzere Schrittweite umgestellt, ohne dass zusätzliche Effekte und Eingriffe des Spielers modelliert werden, muss der Spieler mehr Zeitschritte weiterschalten, um den gleichen Zeitraum zu überbrücken.

**H 15.1:** Der Spieler muss bei feiner Zeitgranularität die Modellzeit häufiger weiterschalten, um uninteressante Zeiträume zu überbrücken. Möglicherweise ist es für den Spieler mühsam, wenn er nur kurze Zeitschritte weiterschalten kann.

Muss der Spieler aber in einer feingranularen Variante häufiger in das Projekt eingreifen und die Entwickler gezielter steuern und kontrollieren, werden diese Zeiträume interessanter. Zusätzlich wird die Modellzeit durch die einzelnen Tätigkeiten und Störungen verbraucht.

**H 15.2:** Für den Spieler treten in einer feingranularen Variante weniger und kürzere Zeiträume auf, die durch Weiterschalten der Modellzeit überbrückt werden müssen.

#### Ausführungszeit

Die Ausführungszeit bezeichnet die Zeit, in der ein Computer mit der Simulation beschäftigt ist, sie enthält auch Wartezeiten. Wird die Simulation in Schulungen eingesetzt, wird auch von der Spielzeit gesprochen (Schneider, 1994, Seite 51). Für Schulungen mit dem QS-Modell hat Drappa (2000, Seite 53) die Ausführungszeit auf einen halben Tag begrenzt.

Bei einer feingranularen Variante benötigt der Spieler aber mehr Zeit, um zu planen. Außerdem muss er häufiger in das Projekt eingreifen. Vermutet wird in der Hypothese H 25 (Seite 26) zusätzlich eine höhere Rechenzeit zur Ausführung des Modells durch den Simulator.

**H 16:** Eine feingranulare Variante kann vom Spieler nicht mehr an einem halben Tag durchgeführt werden (Drappa, 2000, Seite 53).

#### Abstraktionsebenen des Prozesses

Soll sich der Spieler bei einer feingranularen Variante beispielsweise sehr detailliert um Vorbereitung, Durchführung und Nachbereitung eines Reviews kümmern, hat aber bei der Erstellung der Spezifikation nur wenig Einflussmöglichkeiten, verwirrt das den

Spieler. Teile des Prozesses sind dann auf unterschiedlichen Abstraktionsstufen modelliert.

**H 17:** Unterschiedlich feingranulare Phasen eines Modells sind für den Spieler schwer zu durchschauen.

Die Modellierung auf unterschiedlichen Abstraktionsstufen ist nicht nur ein Problem für den Spieler, sondern auch für den Modellbauer und wird als Forderung F 1 auf Seite 17 beschrieben.

### **3.4.3 Nutzen für den Tutor**

#### **Zweck der Schulung**

Eingebunden in ein Schulungskonzept wurde das SESAM-System bereits zur Ausbildung eingesetzt. In Experimenten von Mandl-Striegnitz (2000) hat sich in der Abschlussbefragung gezeigt, dass unerfahrene Teilnehmer das QS-Modell als komplex und unüberschaubar bezeichnen. Erfahrene Teilnehmer dagegen forderten Erweiterungen des Modells um weitere Schwierigkeiten, die in realen Projekten auftreten. Sie verlangen größere Herausforderungen. Dazu zählen auch Störungen, die mit einer feinen Zeitgranularität sinnvoller modelliert werden können (Hypothese H 14, Seite 22).

Wird ein feingranulares Modell eingesetzt, müssen die Spieler detaillierter planen und häufiger in das Projekt eingreifen. Sie werden damit stärker gefordert.

**H 18:** Ein feingranulares Modell ist für die Schulung von fortgeschrittenen Spielern interessanter.

#### **Analyse des Vorgehens**

Inspiziert der Spieler nach der Fertigstellung des Entwurfs das Dokument, stellt er vielleicht fest, dass zu wenig Anforderungen enthalten sind. Er ordnet daher im nächsten Zeitschritt ein Review an. Da bei einer feineren Zeitgranularität weniger Interaktionen pro Zeitschritt mit dem Simulator möglich sind, wird der Zusammenhang zwischen Nachrichten und Kommandos des Spielers zwischen zwei Zuständen klarer.

Die Reihenfolge einzelner Kommandos im Ablauf des Tages kann vom Tutor betrachtet werden. Hat der Spieler kontrolliert, ob noch an einem Dokument gearbeitet wird, bevor es als Vorgabedokument für die nächste Phase verwendet wird? Bei einer feingranularen Variante mit vielen verschiedenen Tätigkeiten kann die Reihenfolge der Kommandos einen Einfluss auf die Ergebnisse haben.

**H 19:** Das Vorgehen der Spieler kann feiner betrachtet werden. Der Zusammenhang zwischen den Meldungen des Systems und Aktionen des Projektleiters zwischen zwei Zuständen wird bei der Analyse klarer.

#### **Analyse des Zustands**

Der Spieler kann an einem Tag sehr viel erledigen. Er kann Mitarbeiter einstellen, anderen Mitarbeitern Tätigkeiten zuweisen oder Mitarbeiter kontrollieren. Wird pro Tag nur

ein Simulationsschritt durchgeführt, wird der Zustand nur einmal angepasst. Ein Vergleich zum Zustand des vorherigen Tages enthält alle Veränderungen aus den Anweisungen des Projektleiters. Wird der Zustand alle zwei Stunden angepasst, verändert sich der Zustand nur als Folge von den wenigen Anweisungen, die in diesen zwei Stunden eingegeben wurden.

**H 20:** Der Tutor kann bei feiner Zeitgranularität den Einfluss eines Ereignisses durch einen Vergleich der Zustände vor und nach dem Ereignis besser feststellen, da pro Zeitschritt weniger Ereignisse stattfinden.

Wird aber eine feingranulare Modellierung durchgeführt, treten im gleichen Zeitraum mehr Effekte als bei einer groberen Modellierung auf. Durch eine kurze Simulationsschrittweite dauert ein Zeitschritt weniger lang, der Tutor kann die Veränderung des Zustandes durch ein Ereignis leichter auf das Ereignis zurückgeführt werden.

### 3.4.4 Nachteile für den Tutor

#### **Komplexe Effekte aus komplexen Situationen**

Im QS-Modell wird der Einfluss der Qualitätssicherung auf die Resultate des Projekts modelliert: Die Qualität der Software steigt, wenn in den frühen Phasen Reviews durchgeführt werden. Wird in einer feingranularen Variante das Review feiner modelliert, können Fehler in der Durchführung des Reviews dazu führen, dass die Qualität kaum verbessert wird.

Erfolg oder Misserfolg des Projekts folgen bereits im QS-Modell aus einer Vielzahl sich überlappender Effekte. Bei einer feingranularen Variante kann der Spieler mehr Fehler bei der Führung des Projekts machen. Das Zusammenspiel zwischen den Aktionen des Spielers, Eigenschaften der Entwickler und Dokumenten und den Folgen während der Simulation bemerkt der Spieler noch schlechter.

**H 21:** Der Spieler benötigt ein detaillierteres Feedback bei einer feingranularen Variante. Der Erklärungsbedarf für den Tutor wird höher.

Der Tutor muss die komplexe Situation und die komplexen Effekte durchschauen.

**H 22:** Der Aufwand für die Auswertung steigt, da die Situation und die Effekte in einer feingranularen Simulation komplexer werden.

## 3.5 Technische Aspekte

In diesem Abschnitt werden die Hypothesen gezeigt, die vermutete Auswirkungen einer feineren Zeitgranularität auf den Simulator bei der Ausführung eines Modells beschreiben. Im zweiten Teil wird eine mögliche Anforderung an die Hochsprache gezeigt.

### 3.5.1 Probleme bei der Simulation

In diesem Abschnitt werden die Hypothesen getrennt für eine feine Zeitgranularität ohne zusätzliche Effekte und für eine feingranulare Variante mit zusätzlichen Effekten betrachtet.

#### 3.5.1.1 Simulation bei feiner Zeitgranularität

##### Performanz

Bei einer feineren Zeitgranularität wird der Ausführungsmechanismus der Basismaschine öfter angestoßen. Bei einer Schrittweite von zwei Stunden wird das Situationsmodell 12 mal mehr angepasst.

**H 23:** Der Simulator braucht bei feiner Zeitgranularität mehr Zeit, um das Situationsmodell anzupassen. Die Ausführungsdauer steigt vermutlich um den Faktor 10.

Wird nur die Zeitgranularität umgestellt, ändern sich das Situationsmodell und die Regelmenge nur geringfügig. Es müssen also die gleichen Regeln auf das gleiche Situationsmodell angewendet werden. Der größte und damit bestimmende Einfluss auf die Dauer eines Simulationsschritts ist die Größe des Situations- und des Regelmodells.

**H 23.1:** Die Dauer eines Simulationsschritts wird sich bei einer feineren Zeitgranularität ohne zusätzliche Effekte nur wenig ändern.

**H 23.2:** Das Antwortverhalten des Simulators auf Kommandos des Spielers wird sich nicht ändern, wenn die Zeitgranularität feiner wird.

Die Zeit, um das Modell in den Simulator zu laden, ist abhängig von der Größe des Modells. Wird ein Modell auf eine feinere Zeitgranularität umgestellt, ändert sich die Größe des Modells nur wenig.

**H 23.3:** Die Zeit, um ein Modell mit feiner Zeitgranularität in den Simulator zu Laden, wird sich nur geringfügig ändern.

##### Speicherbedarf

Auch der Speicherbedarf während der Simulation ist vermutlich hauptsächlich von der Größe des Modells abhängig und wird sich daher kaum verändern.

**H 24:** Der Speicherbedarf ändert sich bei einer Änderung der Zeitgranularität nur gering.

#### 3.5.1.2 Simulation einer feingranularen Variante

##### Performanz

Werden zusätzliche Effekte für eine feingranulare Variante eingefügt, verändert sich die Größe des Modells: Es enthält mehr Komponenten, damit müssen während der Simulation mehr Regeln geprüft und ausgeführt werden. Es werden mehr Entitäten und Relati-

onen erzeugt, das Situationsmodell wird größer. Damit ändert sich die Dauer eines Simulationsschritts.

**H 25.1:** Mit einer feingranularen Variante dauert ein Simulationsschritt länger.

**H 25.2:** Das Laden eines um feingranulare Effekte erweiterten Modells in den Simulator dauert länger, vermutlich linear zur Größe des Modells.

Diese beiden Auswirkungen führen damit zu einer insgesamt längeren Ausführungsdauer:

**H 25:** Die Ausführungsdauer einer feingranularen Variante steigt noch stärker.

Das Antwortverhalten auf Kommandos des Spielers wird sich aber nur wenig ändern, da nur das Kommando gebunden werden muss.

**H 25.3:** Das Antwortverhalten auf Kommandos des Spielers wird sich bei einer feingranularen Variante nur wenig ändern.

### **Speicherbedarf**

Der Speicherbedarf des Simulators ist hauptsächlich von der Größe des Situations- und Regelmodells abhängig. Mit einer feingranularen Variante werden mehr Regeln geprüft und mehr Entitäten und Relationen implementiert und erzeugt.

**H 26:** Der Speicherbedarf des Simulators wird steigen.

Die Veränderungen der Ausführungsdauer und des Speicherbedarfs bei der Simulation eines feingranularen Modells sind von den erzeugten Entitäten und Relationen und von der Anzahl Regeln des Modells abhängig. Sie können deshalb noch nicht quantifiziert werden. Vermutlich steigen sie aber linear mit der Größe des Modells.

## **3.5.2 Anforderungen an die Hochsprache**

### **Arbeitszeit der Entwickler und des Projektleiters**

Die Hochsprache bietet kein Zeitmodell. Damit ist der Modellbauer verantwortlich für die Arbeitszeiten der Entwickler und des Projektleiters. Eine Vereinfachung ist für den Modellbauer, wenn die Hochsprache ein Zeitmodell anbietet. In der Sprachbeschreibung von Schneider et al. (1999) wird bereits ein einfaches, unflexibles Zeitmodell beschrieben.

**F 3:** Die Hochsprache sollte ein flexibles Zeitmodell bieten.

Der Modellbauer muss dann nur noch die Arbeitszeit angeben.



## 3.6 Auswirkungen auf das QS-Modell

In diesem Abschnitt werden die vermuteten Auswirkungen einer feineren Zeitgranularität auf das QS-Modell beschrieben. Da das QS-Modell auf eine Schrittweite von zwei Stunden umgestellt wird, werden zusätzlich die Anforderungen an die Umstellung gezeigt.

### 3.6.1 Nutzen für das QS-Modell

#### Arbeitspakete der Entwickler

Bearbeitet ein Entwickler im QS-Modell ein Dokument, wird für jeden Entwickler ein Arbeitspaket geschnürt. Dieses Paket enthält die Anforderungen, die der Entwickler in das Dokument überträgt. Dadurch wird möglich, dass mehrere Entwickler parallel an einem Dokument arbeiten, ohne sich gegenseitig zu stören. Die Effekte einer Tätigkeit werden bei einer Schrittweite von einem Tag nur einmal pro Tag berechnet. Die Arbeitspakete müssen deshalb so groß sein, dass der Entwickler länger als einen Tag zur Bearbeitung braucht, da der Entwickler bei einem zu kleinen Arbeitspaket an diesem Tag nichts anderes mehr tun kann. Die verlorene Zeit wird als Zeitverschnitt bezeichnet. Bei einer Schrittweite von zwei Stunden verliert der Entwickler nicht mehr so viel Zeit, da er nur die restliche Zeit des Zeitschritts nicht nutzen kann.

**Q 1:** Für die Entwickler wird weniger Zeitverschnitt pro Arbeitspaket anfallen. Die Mindestgröße der Arbeitspakete kann herabgesetzt werden.

Werden mehrere Arbeitspakete nacheinander für einen Entwickler geschnürt, verwendet der Entwickler die übriggebliebene Zeit des Tages, an dem ein Paket fertig bearbeitet wurde, für das nächste Arbeitspaket. Wird er mit dem letzten Arbeitspaket fertig, kann er die restliche Zeit des Tages nicht nutzen. Diese Zeit geht verloren.

**Q 2:** Aus einem kürzeren Zeitschritt folgt weniger Zeitverschnitt, die Entwickler verlieren weniger Zeit.

### 3.6.2 Probleme für das QS-Modell

#### Modellierung der Entwickler

Im QS-Modell ist nur die Teilnahme an Reviews neben einer anderen Tätigkeit möglich. Ein Entwickler kann nicht ein Dokument erstellen und gleichzeitig ein anderes Dokument korrigieren. Entwickler, die als Gutachter an einem Review teilnehmen, verwenden einen bestimmten Zeitanteil der Arbeitszeit, um sich für das Review vorzubereiten und um an der Sitzung teilzunehmen.

Bei einer feineren Zeitgranularität ist die Aufteilung des Zeitschritts aber unrealistisch: Der kurze Zeitschritt von zwei Stunden würde weiter unterteilt werden. Soll das Verhalten der Entwickler realistischer modelliert werden, so dass ein Entwickler mehrere Tätigkeiten parallel durchführt (Hypothese H 4, Seite 16), dann wird der Zeitschritt noch

weiter aufgeteilt. Realistischer ist, wenn die Entwickler etwa alle zwei Stunden die Tätigkeit wechseln und damit die Tätigkeiten quasi-parallel durchführen. Die Effekte dieses Verhaltens, etwa auf den Umfang der Dokumente, können bei einer kurzen Schrittweite beobachtet werden. Diese Modellierung ist nicht im QS-Modell vorgesehen.

**Q 3:** Sollen die einzelnen Tätigkeiten der Entwickler sichtbar werden, muss das gesamte Modell umgebaut werden.

### **Zeitverbrauch der Regeln**

Jede Tätigkeit des Projektleiters verbraucht Zeit. Darum ist im QS-Modell für jedes Kommando des Spielers ein Zeitverbrauch eingetragen. Bei einer feineren Zeitgranularität kann sich herausstellen, dass der Zeitverbrauch nicht realistisch genug ist.

**Q 4:** Möglicherweise muss in den Kommandos und Regeln der Zeitverbrauch des Projektleiters überarbeitet werden, wenn dieser nicht realistisch genug erscheint.

### **Dauer und Gesamtkosten**

Da der Spieler bei feinerer Zeitgranularität früher eingreifen kann (Hypothese H 11, Seite 20), verändert sich die gesamte Dauer und damit auch die Kosten des Projekts. Einzelne Phasen dauern dadurch nicht länger. Der Spieler kann aber den Abstand zwischen einzelnen Tätigkeiten verkürzen.

**Q 5:** Wenn der Spieler die feinere Zeitgranularität nutzt, kann das Projekt schneller und mit weniger Kosten durchgeführt werden.

## **3.6.3 Anforderungen an die Umstellung**

### **Projektresultate**

Die Quantifizierung des QS-Modells gilt als realistisch. Wird die Zeitgranularität verändert, sollten trotz veränderter Schrittweite die entstehenden Dokumente den gleichen Umfang in AFP und die gleiche Anzahl Fehler enthalten.

**F 4:** Die Projektresultate sollen sich bei einer feineren Zeitgranularität nicht ändern.

### **Aufwand, Dauer und Kosten**

Neben den Resultaten des Projekts werden im QS-Modell der Aufwand und die Dauer einzelner Phasen und die durch die eingestellten Entwickler aufgelaufenen Projektkosten berechnet. Das Modell wurde gegen empirische Daten validiert (Drappa, 2000, Seite 135).

**F 5:** Der Aufwand, die Dauer und die Kosten einzelner Phasen des Projekts sollen sich bei einer feineren Zeitgranularität nicht ändern.

Eine Änderung insgesamt ist aber möglich, wenn der Spieler die feinere Zeitgranularität nutzt (Hypothese Q 5).

## **Kapitel 4**

# **Umstellung des QS-Modells auf eine feinere Zeitgranularität**

In diesem Kapitel werden im ersten Abschnitt die notwendigen Änderungen am QS-Modell beschrieben, um eine Modellvariante mit feinerer Zeitgranularität zu erstellen. Für die feinere Zeitgranularität wird die Schrittweite auf zwei Stunden gesetzt. Der zweite Abschnitt beschreibt das Vorgehen für den Test der Modellvariante. Die Testergebnisse werden im dritten Abschnitt dargestellt.

### **4.1 Umstellung des QS-Modells**

Die Phasen Spezifikation und Grobentwurf, jeweils mit Review und Korrektur, werden auf eine feinere Zeitgranularität umgestellt. Die Schrittweite beträgt zwei Stunden. Die Implementierung des QS-Modells ist teilweise abhängig von der Schrittweite. In einigen Regeln wird von einer Schrittweite von einem Tag ausgegangen. Das Modell wird bei der Umstellung entsprechend angepasst. Das durch die Umstellung erstellte Modell ohne Erweiterung um feingranulare Effekte wird in diesem Bericht als Variante 1 bezeichnet. Die Variante mit Erweiterungen wird Variante 2 genannt.

#### **4.1.1 Arbeitszeiten des Projektleiters und der Mitarbeiter**

Im QS-Modell ist ein einfaches Zeitmodell implementiert. Die verfügbare Zeit des Entwicklers wird an Werktagen auf acht Stunden gesetzt, am Wochenende haben die Entwickler keine Zeit zur Verfügung. Damit der Projektleiter am Wochenende nicht in das Projekt eingreifen kann, wird am Wochenende die Zeit des Projektleiters durch eine Regel verbraucht.

Eine Anforderung an die Umstellung ist, dass bei einer Schrittweite von zwei Stunden weder nachts noch am Wochenende gearbeitet werden darf, es soll sich nichts im Vergleich zum QS-Modell ändern (Anforderung F 2, Seite 19). In jedem Simulationsschritt wird festgestellt, ob die Entwickler gerade arbeiten dürfen. Dann bekommen sie die Dauer des Zeitschritts zur Verfügung. Für den Projektleiter sind zusätzliche Regeln implementiert, so dass die Zeit des Projektleiters nicht nur am Wochenende, sondern auch in der Nacht zwischen den Arbeitstagen verbraucht wird. Er kann dann nur zur Arbeitszeit in das Projekt eingreifen.

#### **4.1.2 Ausgabe des aktuellen Datums**

Das Datum wird für den Spieler vom Simulator angezeigt. Die Ausgabe erfolgt nur, wenn der Simulator ruht. Zusätzlich wird das Datum vom Modell an den Spieler ausgegeben. Für die Analyse wird dem Tutor der Wochentag gemeldet. Diese Regeln sind an

die feine Zeitgranularität angepasst: Das Datum wird nur während der Arbeitszeit, der Wochentag einmal am Tag bei Arbeitsbeginn ausgegeben.

### 4.1.3 Tätigkeiten der Entwickler

Durch die feinere Modellierung der Arbeitszeit ist in der Modellvariante bereits sichergestellt, dass die Entwickler während der Arbeitszeit zwei Stunden pro Zeitschritt arbeiten. Die Quantifizierung des QS-Modells arbeitet mit normierten Werten, die Produktivität und die Prüfleistung werden in AFP pro Stunde angegeben. Die Quantifizierung kann deshalb unverändert in die Variante 1 übernommen werden.

Zusätzlich habe ich in der Modellvariante verhindert, dass Entwickler ein Dokument außerhalb der Arbeitszeit bearbeiten können. Wird im QS-Modell in einem Zeitschritt die verfügbare Zeit nicht völlig verbraucht, kann diese Zeit (der Zeitverschnitt) im nächsten Schritt verwendet werden. Dieser Zeitverschnitt kann auch am Wochenende benutzt werden und führt dann zu unrealistischen Effekten. Dieser Effekt wurde bei der Umstellung verhindert.

### 4.1.4 Berechnung der Kosten

Die Projektkosten berechnen sich im QS-Modell aus fixen Kosten und den Gehältern für den Projektleiter und die eingestellten Mitarbeiter. Gehälter werden im QS-Modell einmal im Zeitschritt ausgerechnet. In dieser Modellvariante werden die Gehälter nicht alle zwei Stunden, sondern einmal am Tag berechnet. Kosten entstehen damit konsistent zum QS-Modell.

### 4.1.5 Berechnung des Aufwands

Im QS-Modell werden zwei unterschiedliche Aufwände berechnet:

- Die Aufwände für die Projektphasen Spezifikation, Entwurf, Implementierung und Handbuch werden im Projektzustand gespeichert. Er enthält den Aufwand jeder Phase für die Erstellung, die Prüfung und die Korrektur insgesamt. Die Phase Entwurf fasst Grobentwurf und Feinentwurf zusammen. Zusätzlich wird der Aufwand für Reviews insgesamt, die Korrektur nach Reviews, Tests und Korrektur nach den Tests im Projektzustand gespeichert. Ich bezeichne diesen Aufwand als Phasenaufwand. Diese Aufwände werden in ganzen, aufgerundeten Manntagen berechnet. Ausnahme ist der Aufwand für Reviews. Da für Reviews nur ein Teil des Tages verwendet wird, wird der Aufwand aus der tatsächlich investierten Zeit der Entwickler berechnet.
- Der tatsächlich investierte Aufwand für einzelne Tätigkeiten der Entwickler wird im Projektlogbuch festgehalten. Für die verschiedenen Dokumente wird also getrennt berechnet, wieviel Zeit für die Erstellung, die Prüfung und die Korrektur aufgewendet wurde. Dieser Aufwand wird feiner berechnet, es fließt nur die tatsächlich investierte Zeit ein.

Die Aufwandsberechnung für Reviews im Projektzustand und im Projektlogbuch musste für die Variante 1 verändert werden, da der Zeitschritt im QS-Modell direkt verwendet wird.

#### 4.1.6 Zeitverbrauch der Kommandos

Während der Umstellung auf eine feine Zeitgranularität wurde der Zeitverbrauch einzelner Kommandos betrachtet. Damit wird die Hypothese Q 4 (Seite 28) untersucht. Für den umgestellten Ausschnitt des Modells sind Kommandos zum Erstellen der Spezifikation und des Entwurfs, zur Prüfung durch Reviews und zur Korrektur und Inspektion der Dokumente relevant. Möchte der Projektleiter den Zustand eines Dokuments herausfinden, dann schaut er sich das Dokument an, etwa mit dem Kommando *“Inspect Specification”*. Er erhält daraufhin eine Meldung mit dem Umfang des Dokuments in Seiten und eine grobe Abschätzung der bereits errechneten Funktionalität: *“The Specification has a volume of 88 pages and between 87 percent to 95 percent of the functionality included.”*. Das Kommando verbraucht 20 Minuten – um die Funktionalität abzuschätzen, braucht man meiner Meinung nach mehr Zeit. Der Projektleiter muss das Dokument lesen. Der Zeitverbrauch wurde deshalb auf 120 Minuten für die Spezifikation erhöht, der Systementwurf ist komplexer, der Projektleiter braucht 180 Minuten.

## 4.2 Organisation des Tests

Die Variante 1 mit feinerer Zeitgranularität aber ohne zusätzliche, feingranulare Effekte wird durch Test überprüft. Damit werden mehrere Ziele verfolgt:

- Überprüft werden soll, ob sich Veränderungen an den Ergebnissen durch die feinere Zeitgranularität ergeben. Hat die Zeitgranularität einen Einfluss auf die Qualität der Dokumente? Verändern sich Dauer, Aufwand oder Kosten des Projekts? Wirken sich Effekte des Modells, die von der Modellzeit abhängen, unterschiedlich aus?
- Mit der Modellvariante kann bereits ein Teil der Hypothesen für die Modellierung und die Ausbildung untersucht werden.
- Vor allem können Hypothesen über technische Aspekte einer feineren Zeitgranularität überprüft werden.

### 4.2.1 Vorgehen

Die Modellvariante wurde anhand einer Testvorschrift geprüft. Diese Testvorschrift gliedert sich in die verschiedenen Ziele, die mit dem Test verfolgt werden. Im ersten Teil werden Hypothesen über Modellaspekte und die erzielten Projektergebnisse überprüft. Im zweiten und dritten Teil werden die Hypothesen über die Ausbildung beziehungsweise technische Aspekte untersucht.

Für einzelne Testfälle definiert die Testvorschrift den Zweck und den Ablauf des Tests, welches Modell benutzt wird, die Eingabedatei für den Simulator und die entstehenden Protokolldateien. Die Testvorschrift orientiert sich an Frühauf et al. (2000).

Einflüsse der Zeitgranularität auf die Ergebnisse von Projektsimulationen werden durch einen Vergleich zwischen Simulationen mit dem QS-Modell und der Modellvariante mit feiner Zeitgranularität festgestellt. Technische Aspekte wie die Performanz des Simulators werden wie die Einflüsse der Zeitgranularität durch einen Vergleich zwischen den beiden Modellen untersucht.

Die Ergebnisse des Tests sind in einem Testbericht zusammengefasst.

### 4.2.2 Untersuchte Hypothesen

Im ersten Teil des Tests werden zwei Hypothesen über die Modellierung untersucht:

- In der Hypothese H 1 (siehe Seite 15) wird vermutet, dass kontinuierliche Vorgänge besser in das Modell abgebildet werden können.
- Die Hypothese H 2 beschreibt auf Seite 16, dass bereits vorhandene feingranulare Effekte beobachtet werden können.

Unter den Modellaspekten werden außerdem die Anforderungen F 4 (Seite 28) und F 5 (Seite 28), die die gleichen Projektergebnisse und Ergebnisse fordern, und die folgenden Hypothesen geprüft:

- Es fällt weniger Zeitverschnitt an (Hypothese Q 1, Seite 27) und es können kürzere Arbeitspakete definiert werden (Hypothese Q 2, Seite 27).

Ein Teil der Hypothesen über die Ausbildung mit einer feineren Zeitgranularität kann bereits mit dieser Modellvariante nachgeprüft werden:

- Der Spieler erhält Rückmeldungen des Modells früher und kann früher in das Projekt eingreifen (Hypothese H 11 und H 12, Seite 20).
- Wird die feinere Zeitgranularität genutzt, kann der Spieler das Projekt in kürzerer Zeit und mit weniger Kosten durchführen (Hypothese Q 5 auf Seite 28).
- Der Spieler kann den Tagesablauf wahrnehmen (Hypothese H 14, siehe Seite 22). Er bemerkt bei feinerer Zeitgranularität den Zeitverbrauch der Aktionen als Projektleiter.
- Es finden weniger Ereignisse in einem Zeitschritt statt (Hypothese H 20, Seite 24). Der Tutor kann deshalb besser den Einfluss der Ereignisse durch einen Vergleich der Zustände vor und nach einem Ereignis feststellen.

Hypothesen der technischen Aspekte, die die Ausführung eines feingranularen Modells betreffen, werden ebenfalls betrachtet: Auswirkungen auf die Performanz des Simulators werden durch die Hypothese H 23 und ihre Teilhypothesen ab Seite 24 beschrieben

und in diesem Test untersucht, die Hypothese H 24 befasst sich mit dem Speicherplatz, den der Simulator benötigt.

## 4.3 Erzielte Ergebnisse

In diesem Abschnitt werden die im Test erzielten Ergebnisse dargestellt.

### 4.3.1 Modellaspekte

#### 4.3.1.1 Projektresultate und -ergebnisse bei unterschiedlicher Zeitgranularität

##### Testbeschreibung

Um die Anforderungen F 4 und F 5 zu prüfen, werden drei Testfälle jeweils mit dem QS-Modell und der Modellvariante mit einer Schrittweite von zwei Stunden durchgeführt. Jeder Testfall spielt die Phasen Spezifikation und Entwurf jeweils mit Reviews und darauf folgender Korrektur durch. Im Fall 1 werden in den Reviews zwei Gutachter eingesetzt, im Fall 2 nimmt der Kunde am Spezifikationsreview teil. Drei Gutachter werden im Spezifikations- und Entwurfsreview im Testfall 3 eingesetzt. Alle Gutachter haben eine mittlere Erfahrung, um ungewollte Effekte durch individuelle Eigenschaften der Entwickler auszuschließen.

##### Ergebnisse

Unabhängig von der Zeitgranularität sind die erzielten Resultate des Projekts, die Dokumente Spezifikation und Entwurf, gleich. Die Prüfberichte der Reviews enthalten die gleiche Anzahl Fehler und Verluste. Der tatsächlich benötigte Aufwand im Projektlogbuch bleibt trotz unterschiedlicher Schrittweite gleich.

Bei unterschiedlicher Schrittweite verändert sich die Dauer der einzelnen Tätigkeiten leicht. Bei grober Zeitgranularität wird nicht berücksichtigt, wann genau ein Entwickler die Tätigkeit beendet hat. Die Uhrzeit hat bei grober Zeitgranularität keine Bedeutung. Das Ende der Tätigkeit wird immer um 08:00 Uhr im Projektlogbuch markiert, auch wenn der Entwickler einen Teil des Tages verwendet hat. Bei feiner Zeitgranularität wird auf volle zwei Stunden abgerundet. Ein Beispiel dafür ist das Ende des Entwurfs in Abbildung 7.

Tätigkeit		QS-Modell	Variante 1
Entwurf	Aufwand (MH)	126.117	126.117
	Beginn	1999/09/17/08:00	1999/09/17/08:00
	Ende	1999/10/08/08:00	1999/10/08/14:00

**Abb. 7:** Aufwand und Dauer des Entwurfs

Hier kann ein feingranularer Effekt erkannt werden: Der Entwickler benötigt am letzten Tag des Entwurfs noch einen Teil des Tages. Erst bei feiner Zeitgranularität kann dieser Effekt beobachtet werden (Hypothese H 2).

Unterschiede ergeben sich beim Phasenaufwand und bei den Kosten des Projekts:

Phasenaufwand	QS-Modell	Variante 1
Spezifikationsphase (MH)	153.465	133.465
Entwurfsphase (MH)	189.912	171.912
Reviews (MH)	54.951	54.164
Korrektur (MH)	56.0	24.0
Gesamtaufwand (MM)	2.6013	2.3135
Kosten insgesamt (DM)	154440.0	154680.0

**Abb. 8:** Phasenaufwand und Kosten im Vergleich

Der geringe Unterschied des Aufwandes für Reviews ist die Folge von Rundungsfehlern, da der Aufwand bei feiner Zeitgranularität alle zwei Stunden, bei grober Zeitgranularität einmal am Tag berechnet wird. Weitere Unterschiede resultieren aus der Berechnung des Aufwandes beim Erstellen – der Zeitverschnitt wird beim Phasenaufwand mitgezählt, im QS-Modell werden für jeden angefangenen Tag volle acht Stunden berechnet. Die große Differenz der Korrektur wird durch zu kleine Arbeitspakete bestimmt. Dieser Effekt wird daher näher betrachtet.

Korrigiert ein Entwickler ein Dokument, werden einzelne Arbeitspakete mit einem Teil des Dokuments geschnürt. Pro Zeitschritt kann höchstens ein Arbeitspaket erstellt werden. Bei der Korrektur werden im QS-Modell zu kleine Arbeitspakete geschnürt, der Entwickler ist bereits nach rund zwei Stunden fertig und muss bis zum nächsten Tag auf ein neues Arbeitspaket warten. Für den Phasenaufwand wird aber trotzdem der gesamte Tag berechnet. Der Entwickler braucht damit aber länger, der Zeitverschnitt ist sehr groß (Hypothese Q 1, siehe Seite 27) und kann nicht genutzt werden. Bei feiner Zeitgranularität kommt es nicht zu diesen unrealistischen Effekten, da Arbeitspakete mehrmals am Tag geschnürt werden. Mit feinerer Zeitgranularität sind also kleinere Arbeitspakete möglich (Hypothese Q 2).

Die Unterschiede bei der Erstellung, der Prüfung und der Korrektur summieren sich im Aufwand für die Spezifikationsphase und die Entwurfsphase.

Die leichte Differenz bei den Projektkosten entsteht in der Projektinitialisierung im ersten Zeitschritt: Während im QS-Modell der gesamte erste Tag für die Projektinitialisierung verwendet wird, kann der Projektleiter in der Variante 1 nur den ersten Zeitschritt nicht nutzen. Sein Gehalt wird daher bereits am ersten Tag zu den Kosten des Projekts addiert.



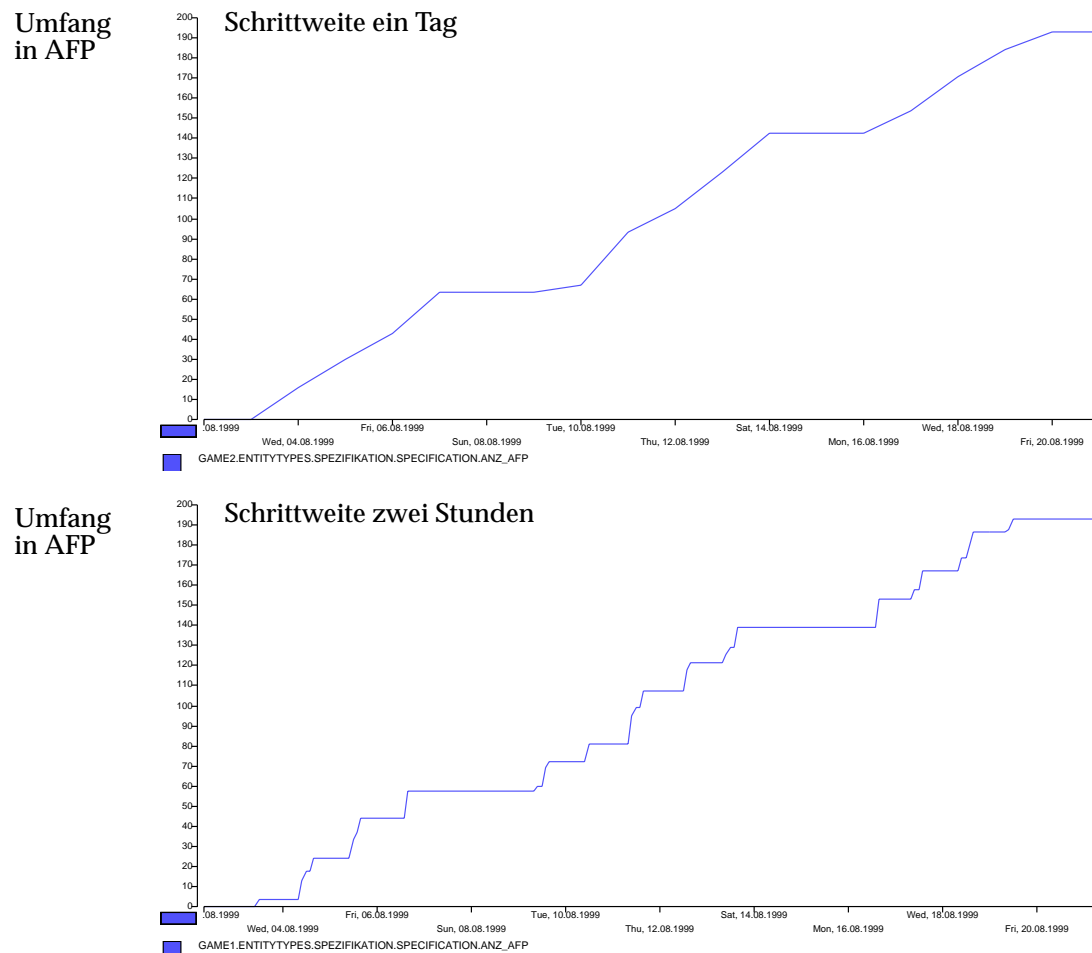
### 4.3.1.2 Kontinuierliche Vorgänge

#### Testbeschreibung

Die Auswirkung der feineren Zeitgranularität auf kontinuierliche Vorgänge (Hypothese H 1, Seite 15) wird an der Spezifikation gezeigt. Ein Entwickler erstellt die Spezifikation, der Umfang des Dokuments, der in AFP gemessen wird, nimmt mit der Zeit zu.

#### Ergebnisse

Der Vergleich zwischen den Ergebnissen bei grober und bei feiner Zeitgranularität zeigt einen deutlichen Unterschied im Verlauf (Abbildung 9). Im oberen Teil der Abbildung wird der Umfang der Spezifikation bei einer Schrittweite von einem Tag dargestellt, der untere Teil zeigt den Umfang in AFP bei einer Schrittweite von zwei Stunden. Bei feiner Zeitgranularität sind die einzelnen Stufen des Umfangs feiner. Die Arbeitszeit des Entwicklers kann erkannt werden, der Verlauf entspricht stärker der Realität.



**Abb. 9:** Umfang der Spezifikation im Vergleich

## 4.3.2 Einfluss auf die Ausbildung

### 4.3.2.1 Eingriffe des Spielers

#### Testbeschreibung

Bei feiner Zeitgranularität kann der Spieler früher eingreifen. Er erhält Rückmeldungen früher (Hypothesen H 11 und H 12, Seite 20). Im Test hat ein Entwickler die Spezifikation erstellt, der Projektleiter wartet mit der Prüfung durch Reviews, bis die Spezifikation vollständig erstellt wurde. Sofort danach wird ein Review mit zwei Gutachtern angewiesen.

#### Ergebnisse

Die Hypothesen werden durch einen Auszug aus einem Spiel am Simulator gezeigt.

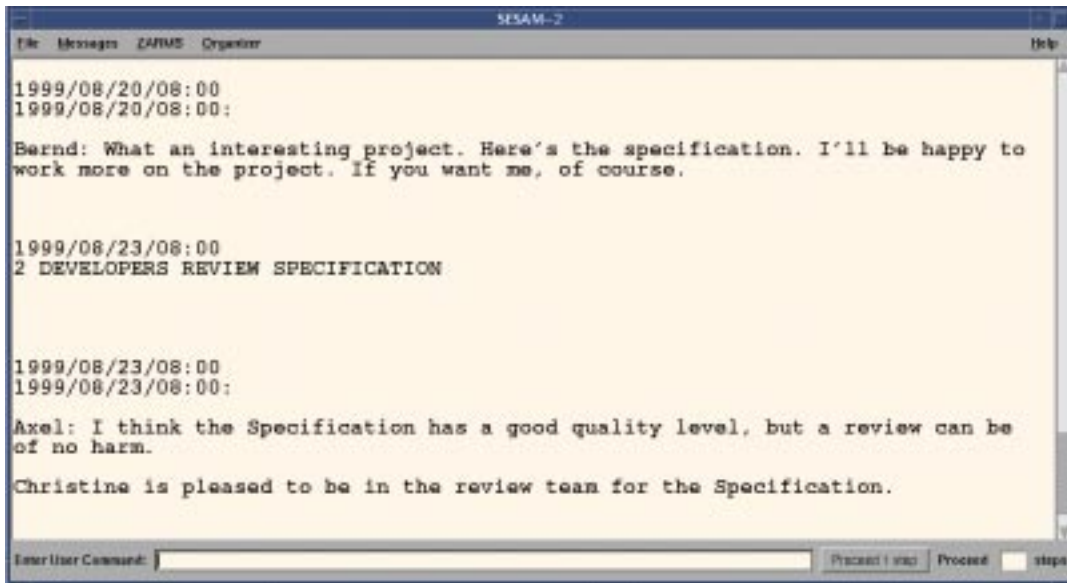
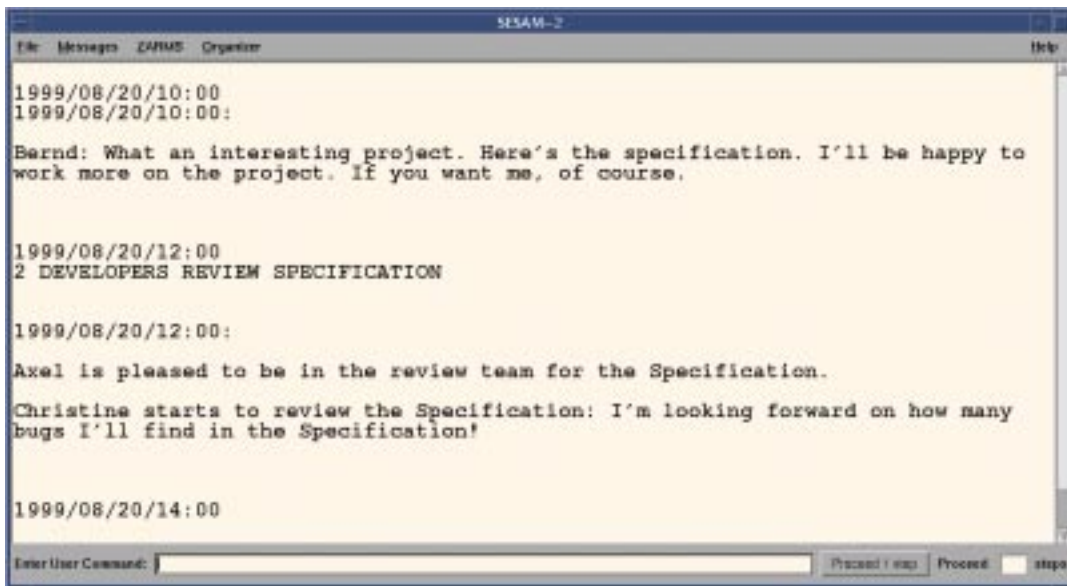


Abb. 10: Aktion bei grober Zeitgranularität

In Abbildung 10 erhält der Spieler die Meldung erst, wenn der Zeitschritt am 20.08. vorbei ist, also am Ende des Tages. Er kann erst am nächsten Tag wieder in das Projekt eingreifen. Im Beispiel meldet der Entwickler an einem Freitag, dass er die Spezifikation erstellt hat, der Spieler kann erst nach dem Wochenende eingreifen.



**Abb. 11:** Aktion bei feiner Zeitgranularität

Der Spieler erhält die Meldung bei feiner Zeitgranularität noch während des Tages und kann noch am gleichen Tag die Gutachter für die Reviews einladen.

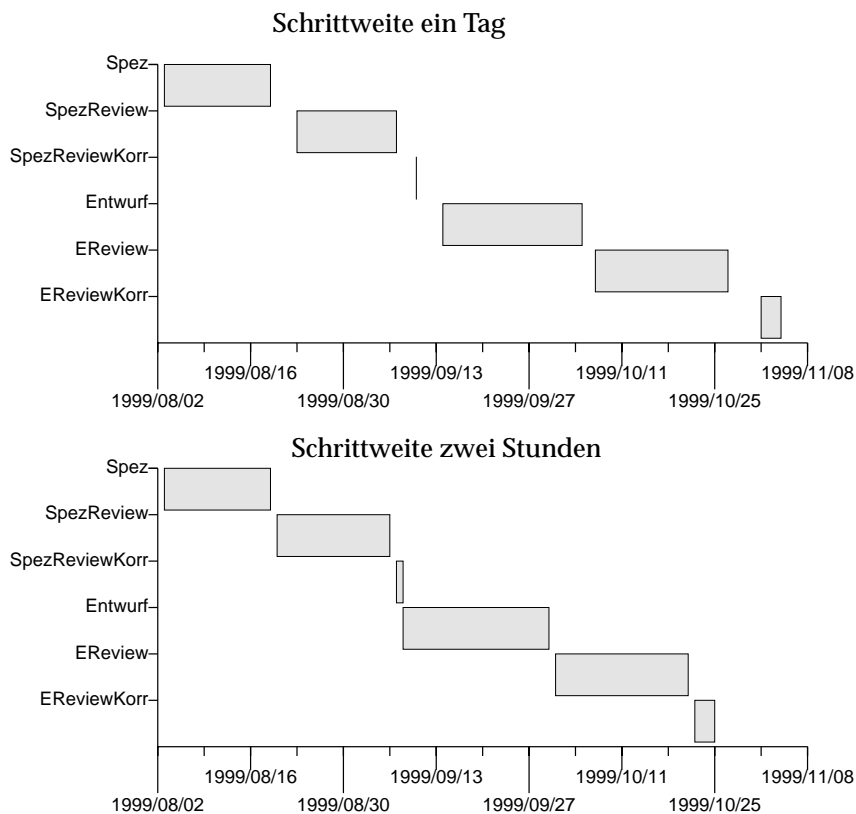
#### 4.3.2.2 Dauer und Gesamtkosten

##### Testbeschreibung

Aus den im vorigen Abschnitt gezeigten Hypothesen H 11 und H 12 folgt die Hypothese Q 5. Der Spieler kann den Projektablauf straffen, dadurch sinken die Kosten des Projekts. Der Testfall führt daher die Phasen Spezifikation und Entwurf (mit Prüfung und Korrektur) nacheinander mit möglichst wenig Abstand zwischen den einzelnen Tätigkeiten durch.

##### Ergebnisse

Die beiden Gantt diagramme zeigen, dass die einzelnen Tätigkeiten gleich lange dauern. Der Zeitraum zwischen den Tätigkeiten ist bei der Variante 1 mit feiner Zeitgranularität kürzer. Abbildung 12 zeigt auf Seite 38 im oberen Gantt diagramm den Spielverlauf bei einer Schrittweite von einem Tag, darunter bei einer Schrittweite von zwei Stunden. Durch die Schrittweite von zwei Stunden werden neun Arbeitstage gespart.



**Abb. 12:** Vergleich der Projektverläufe

### 4.3.2.3 Tagesablauf des Projektleiters

#### Testbeschreibung

In der Modellvariante mit zwei Stunden Schrittweite sind keine zusätzlichen feingranularen Effekte modelliert. Trotzdem kann bereits ein Hinweis für die Hypothese H 14 (Seite 22) gezeigt werden. Bei einer kurzen Schrittweite bemerkt der Spieler, wie lange die Eingriffe in das Projekt dauern. Im Test wird ein Spielausschnitt nachgespielt, der ein etwas chaotisches Vorgehen zeigt. Der Projektleiter hat einen Entwickler zur Spezifikation eingesetzt. Nach einigen Tagen schaut er sich die bisher entstandene Spezifikation an und stellt fest, dass die Spezifikation nicht schnell genug wächst. Damit der Spieler seinen Projektplan einhalten kann, stellt er deshalb zwei weitere Entwickler ein, die bei der Spezifikation helfen sollen.

#### Ergebnis

Da der Simulator durch den Zeitverbrauch der Spielereingriffe weiterschaltet, sieht der Spieler, dass sich die Modellzeit verändert. Vor allem die Einstellung von Entwicklern dauert lange, im QS-Modell verbraucht dieses Kommando zwei Stunden der Projektleiterzeit. Erst im letzten Zeitschritt schaltet der Spieler die Modellzeit weiter. Abbildung 13 zeigt Modellzeit, Kommandos und Nachrichten der Variante 1. Kommandos des Spielers sind in Großbuchstaben geschrieben.

```
1999/08/16/08:00
INSPECT SPECIFICATION
The Document Specification comprises 61 pages. Given your experience,
that should be between 32 percent and 36 percent of the functionality.

1999/08/16/10:00
HIRE DIANA
Diana is glad somebody recognised her talents and hired her.

1999/08/16/12:00
DIANA SPECIFY
HIRE AXEL
Axel: You hired me. Now what? How can I help you?
Diana: Specifying the system requirements is not my favorite activity,
nevertheless I'll do my best.

1999/08/16/14:00
AXEL SPECIFY
PROCEED 1 STEP

Axel: If you need somebody to specify the system requirements, I'll be
happy to help out.
```

**Abb. 13:** Tagesablauf des Projektleiters

Dieser Testfall liefert auch einen Hinweis für die Hypothese H 20 (Seite 24). Bei grober Zeitgranularität werden im Verlauf eines Zeitschritts die Relationen für die Bearbeitung der Spezifikation durch zwei weitere Entwickler angelegt, in der Entität der Entwickler wird vermerkt, dass sie eingestellt sind und die Arbeitspakete zum Spezifizieren werden geschnürt. Der Zustand verändert sich in diesem Zeitschritt stark. Bei feiner Zeitgranularität verteilen sich die Veränderungen auf die verschiedenen Zeitschritte.

### 4.3.3 Technische Aspekte

#### 4.3.3.1 Performanz und Speicherbedarf des Simulators

##### Testbeschreibung

Die Auswirkungen einer feineren Zeitgranularität auf die Ausführung eines Modells werden mit den in Abschnitt 4.3.1.1 beschriebenen Testfällen durchgeführt – Testfall 1 spielt Spezifikation und Entwurf mit zwei Gutachtern in den Reviews durch, im Testfall 2 wird der Kunde eingebunden und im Testfall 3 werden drei Gutachter eingesetzt. Alle Testfälle wurden mehrmals durchgeführt, Ausführungszeit und Speicherbedarf wurden in Protokolldateien festgehalten. Die Testfälle wurden nachts ohne interaktive Eingriffe des Spielers durchgeführt, wenn die Belastung des Rechners möglichst gering ist. Der eingesetzte Rechner ist eine SUN Enterprise 450 mit vier UltraSparc-II CPUs mit 400 MHz und 1 GByte Hauptspeicher.

## Ergebnisse

Die Hypothese H 23 (Seite 25) wird durch die gesamte Ausführungszeit der Testläufe geprüft. Abbildung 14 zeigt die Durchschnittswerte der Testfälle.

Testfall	Dauer (s)		Faktor
	QS-Modell	Variante 1	
1	58.5	457.8	7.83
2	58.4	448.9	7.69
3	60.5	471.5	7.80

**Abb. 14:** Ausführungszeit

Die Ausführungszeit steigt nicht ganz so stark an wie in der Hypothese befürchtet. Zusätzliche, feingranulare Effekte sind in der Variante 1 nicht enthalten. Zwar wird der Zustand zwölf mal öfter berechnet, die Veränderungen an sich sind aber die gleichen, unabhängig von der Zeitgranularität.

Die Ausführungszeit wurde anhand der aufgestellten Hypothesen genauer untersucht. Dazu wurde die Ausführungszeit einzelner Zeitschritte gemessen, ohne dass der Spieler eingreift. Das Modell läuft also leer, Anforderungen werden nicht von einem Dokument in ein anderes übertragen. Die Ausführungszeit für einen einzelnen Simulationsschritt wird bei einer feineren Zeitgranularität kaum größer, sie steigert sich von 168 ms auf 171 ms, da die zusätzlichen Regeln für die Arbeitszeit der Entwickler und des Projektleiters berücksichtigt werden müssen (Hypothese H 23.1). Diese Regeln führen auch zu einer kaum merkbar längeren Ladezeit des Modells. Der Simulator braucht für beide Modelle ungefähr 13 Sekunden zum Laden, für die Variante 1 mit feiner Zeitgranularität 60 ms länger (Hypothese H 23.3 auf Seite 25).

Das Antwortverhalten auf Kommandos des Spielers ändert sich, wie in der Hypothese H 23.2 vermutet, nicht. Ausnahmen treten immer dann auf, wenn der Simulator durch den Zeitverbrauch der eingegebenen Kommandos weiterschaltet, da sich an das zuletzt eingegebene Kommando sofort ein Simulationsschritt anschließt. Die Hypothese wird bestätigt.

Es ändert sich aber der Speicherbedarf: In Hypothese H 24 wird keine Veränderung des Speicherbedarfs vermutet, im Test hat sich aber gezeigt, dass bereits eine Änderung der Zeitgranularität zu einem höheren Speicherbedarf führt (Abbildung 15).

Testfall	Memory (MB)		Faktor
	QS-Modell	Variante 1	
1	20	69	3.45
2	20	69	3.45
3	20	71	3.55

**Abb. 15:** Speicherbedarf

Vermutlich wird in jedem Simulationsschritt Speicher belegt, der nicht mehr freigegeben wird.

# Kapitel 5

## Konzeption einer feingranularen Variante

Im vorigen Kapitel wird die Umstellung des QS-Modells auf eine feinere Zeitgranularität beschrieben. Dieses Kapitel beschreibt die Erweiterung des QS-Modells um feingranulare Effekte. Im ersten Abschnitt wird ein Überblick über das feingranulare Modell gegeben und das Vorgehen bei der Modellentwicklung erläutert. Abschnitt 5.2 definiert den Zweck und das Original für den Reviewprozess, Abschnitt 5.3 zeigt die Modellierung mit Effekten und Zusammenhängen. Die Quantifizierung wird in Abschnitt 5.4 dargestellt.

### 5.1 Überblick und Vorgehen

#### 5.1.1 Überblick über die feingranulare Variante

Die Variante 1 des QS-Modells mit feiner Zeitgranularität aus Kapitel 4 wird aus zwei verschiedenen Richtungen um feingranulare Effekte erweitert. Die erweiterte Variante wird als Variante 2 oder als feingranulare Variante bezeichnet. Das bisherige Modell für das Review der Spezifikation und des Grobentwurfs wird durch ein feingranulares Reviewmodell ersetzt. Die Vorbereitung durch die Gutachter und die Sitzung werden getrennt und detaillierter modelliert. Zusätzlich werden bestehende Modellkomponenten um feingranulare Effekte erweitert. Dazu gehören das Verhalten und die Arbeitszeit der Mitarbeiter, Störungen des Projektleiters und Treffen des Projektleiters mit den Mitarbeitern. Das QS-Modell ist in Abschnitt 1.2 kurz beschrieben, dort werden auch Effekte im Review gezeigt.

Mit den feingranularen Erweiterungen werden die Hypothesen aus Kapitel 3 untersucht.

#### 5.1.2 Vorgehen bei der Erweiterung

Die Variante 2 wird nach dem gleichen Prinzip wie ein neues Modell erstellt. Das Vorgehen orientiert sich daher an Bossel (1992) und dem Vorgehen von Joos (1997) und Eisenbarth und Rohrbach (1998).

Der erste Schritt ist die Definition des Modellzwecks. Daraus wird abgeleitet, was modelliert werden soll und wo die Grenzen des Modells liegen. SESAM-Modelle dienen zur Ausbildung, daher werden die Ausbildungsziele definiert (Drappa, 2000). Für die Variante 2 wird bestimmt, um welche Ausbildungsziele die Ziele des QS-Modells erweitert werden.



Nachdem die Ausbildungsziele der feingranularen Variante festgelegt wurden, können die Effekte und Zusammenhänge der Variante qualitativ beschrieben werden: Welche Objekte und Beziehungen werden modelliert? Welche dynamischen Effekte sollen im Modell auftreten? Wie kann der Spieler eingreifen?

Im nächsten Schritt wird festgelegt, wie die Eigenschaften der Objekte und Beziehungen in die Variante abgebildet werden sollen. Damit werden die Metriken des Modells definiert. Ein großer Teil der Metriken ist durch das QS-Modell vorgegeben.

Als letzter Schritt der Modellkonzeption wird die Quantifizierung der Variante 2 erstellt. Die Quantifizierung stützt sich auf Literatur und auf das QS-Modell. Das QS-Modell gilt als realistisch, Spiele mit der feingranularen Variante sollen ähnliche Resultate zeigen wie Spiele mit dem QS-Modell.

## 5.2 Ausbildungsziele und Abgrenzung des Originals

Der Zweck des QS-Modells leitet sich aus dem Ausbildungsbedarf von Projektleitern ab (Drappa, 2000, Kapitel 3). Es werden die folgenden Ausbildungsziele festgelegt:

- Der Spieler soll lernen, ein Projekt zu planen und durchzuführen.
- Er soll die Schwierigkeiten eines Projektleiters erfahren. So ist etwa der Projektstand kaum bekannt und muss durch Maßnahmen wie Reviews erst festgestellt werden.
- Der Spieler soll Zusammenhänge zwischen seiner Steuerung des Projekts und den Projektergebnissen erfassen.
- Dem Spieler sollen quantitative Daten über Projekte vermittelt werden.

Die feingranulare Variante soll die Ausbildungsziele des QS-Modells erweitern, dabei wird in dieser Arbeit vor allem der Reviewprozess betrachtet:

- Der Spieler soll die Aufgaben im Zusammenhang mit dem Projektmanagement für das technische Review (Frühauf et al., 2000) erlernen. Er muss die Durchführung der Vorbereitung, der Sitzung und der Nachbereitung steuern und kontrollieren.
- Der Spieler soll im Hinblick auf die Reviews lernen, detaillierter zu planen. Einzelne Sitzungen müssen geplant werden, die Gutachter müssen genügend Vorbereitungszeit bis zur Sitzung haben.
- Der Spieler soll für Reviews lernen, Mitarbeiter mit ähnlicher Granularität wie in der Realität zu führen und zu kontrollieren.
- Der Spieler soll dabei den Tagesablauf eines Projektleiters erfahren.

Um diese Ziele zu erreichen, wird das Review detaillierter modelliert. Dieses Reviewmodell wird im Abschnitt 5.3.1 beschrieben. Ergänzend zum Review werden weitere feingranulare Effekte modelliert: Für eine feinere Steuerung des Projekts wird das Verhalten der Entwickler und ihre Arbeitszeit detaillierter modelliert (Abschnitt 5.3.3). Der Tagesablauf wird außerdem durch Störungen und Treffen mit Mitarbeitern geprägt, diese Effekte werden in Abschnitt 5.3.2 gezeigt.

Bestätigen sich die Hypothesen, können diese Ziele erst durch eine feinen Zeitgranularität erreicht werden. In den Hypothesen in Kapitel 3 wurde vermutet, dass die feine Zeitgranularität für den Tagesablauf des Projektleiters (Hypothese H 14, Seite 22), eine feinere Steuerung des Projekts durch den Spieler (Hypothese H 13, Seite 21) und eine detailliertere Planung (Hypothese H 10, Seite 20) nützlich ist. Die feine Zeitgranularität erlaubt, das Verhalten der Entwickler und ihre Arbeitszeit detaillierter zu modellieren (Hypothese H 4 und H 4.1 auf Seite 16).

## **5.2.1 Die Modellabgrenzung und das Original des Reviewprozesses**

In diesem Abschnitt wird der Reviewprozess vorgestellt, der in die feingranulare Variante abgebildet werden soll.

### **5.2.1.1 Der Reviewprozess**

Ein Review ist die inhaltliche Prüfung eines Dokuments. Dabei soll festgestellt werden, ob der Prüfling die Qualitätskriterien erfüllt oder ob Verbesserungen nötig sind. Der Projektleiter kann diese Prüfung nicht selbst durchführen, da er die nötige Zeit oder die nötige Qualifikation nicht hat. Die Beschreibung in diesem Abschnitt ist eine Zusammenfassung aus Frühauf et al. (2000).

#### **Beteiligte Rollen**

Der Projektleiter gibt den Auftrag für das Review. Der Moderator ist für die Durchführung des Reviews verantwortlich und leitet die Sitzung. Die Gutachter bereiten sich auf die Sitzung vor, sie prüfen den Prüfling gegen ein Vorgabedokument und gegen Prüfkriterien und Richtlinien in Fragenkatalogen. In der Reviewsitzung werden die Befunde, die die Gutachter entdeckt haben, zusammengetragen und bewertet. Der Aktuar protokolliert die Befunde. Für Fragen zum Prüfling steht in der Sitzung der Autor zur Verfügung.

Moderator, Gutachter und Aktuar werden als Reviewteam bezeichnet. Dabei müssen aber nicht alle Rollen durch verschiedene Personen besetzt sein – die Rolle des Aktuars kann auf Gutachter verteilt oder vom Autor wahrgenommen werden.

#### **Hilfsmittel**

Ein wichtiges Hilfsmittel zur Prüfung sind Fragenkataloge. Sie enthalten Richtlinien und Prüfkriterien in Frageform. Fragenkataloge sind nach Aspekten gegliedert.

#### **Ablauf eines technischen Reviews**

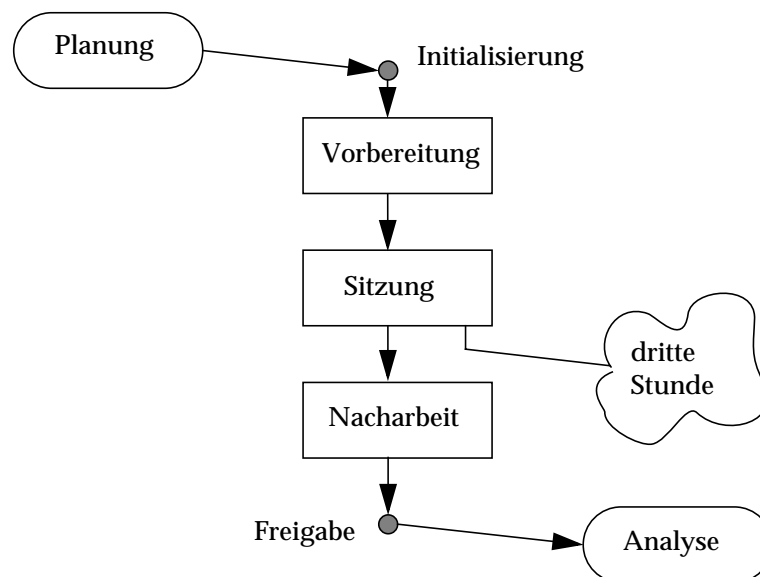
In der Planung bestimmt der Projektleiter den Moderator und wählt mit dem Moderator aus, nach welchen Aspekten der Prüfling begutachtet werden soll und welche Gutachter eingeladen werden. Sie legen fest, welcher Gutachter welche Aspekte prüft. In der Initialisierung stellt der Projektleiter fest, ob der Prüfling tatsächlich für ein Review geeignet ist. Der Moderator versorgt dann die Gutachter mit dem Referenzdokument, dem Prüf-

ling und den Fragenkatalogen. Möglich ist in dieser Phase eine Einführungssitzung, in der den Gutachtern das Umfeld und der Prüfling vorgestellt wird.

Die Gutachter untersuchen in der Vorbereitung den Prüfling nach den ihnen individuell zugeordneten Aspekten anhand der Fragenkataloge und des Referenzdokuments. Sie notieren sich ihre Befunde. In der Sitzung werden die Befunde besprochen und bewertet. Der Moderator leitet die Sitzung und sorgt für die Einhaltung von Review-Regeln, beispielsweise dass alle Gutachter zu Wort kommen, dass der Autor nicht persönlich angegriffen wird und dass die Diskussion nicht abschweift.

Abschließend einigen sich die Gutachter auf eine Empfehlung, die an den Projektleiter gegeben wird: Er sollte den Prüfling *akzeptieren*, *akzeptieren mit Überarbeitung* oder *nicht akzeptieren*. Soll der Prüfling überarbeitet werden, kann entweder eine neue Reviewsitzung verlangt werden oder die Gutachter empfehlen, dass eine Prüfung durch einen Gutachter oder den Moderator ausreicht. In der dritten Stunde können Lösungsansätze oder der Verlauf des Reviews diskutiert werden. Der Projektleiter entscheidet über die Freigabe des Prüflings oder die notwendige Nacharbeit.

Abbildung 16 zeigt die einzelnen Phasen des technischen Reviews nach Frühauf et al. (2000).



**Abb. 16:** Phasen des technischen Reviews

## 5.3 Effekte und Zusammenhänge der feingranularen Variante

### 5.3.1 Modellierung des Reviews

In diesem Abschnitt werden Effekte und Zusammenhänge des feingranularen Reviewmodells gezeigt. Das bisherige Reviewmodell ist im Abschnitt 1.2.3 bereits kurz beschrieben.

#### 5.3.1.1 Ablauf und Eingriffe des Spielers

Der Spieler bleibt auch bei einem feingranularen Reviewmodell in der Rolle des Projektleiters. Ein Wechsel zur Rolle des Moderators würde einen Bruch für den Spieler bedeuten. Die Leitung der Sitzung kann nicht modelliert werden: Gestik, Mimik und Ausdrucksweise sind nicht darstellbar, der persönliche Stil bei der Leitung kann durch Kommandos schlecht nachgebildet werden, Diskussionen können mit dem SESAM-Simulator nicht unterbrochen werden.

Wie im QS-Modell liegt die Planung außerhalb der Systemgrenzen des Modells. Sie muss vom Spieler durchgeführt werden: Er muss bereits im Projektplan berücksichtigen, wann die Prüfung in einzelnen Sitzungen stattfindet, welche Entwickler die Rollen des Reviews übernehmen und wie eine systematische Prüfung erreicht wird.

Der Spieler bestimmt für die Initialisierung das zu prüfende Dokument, den Moderator, welcher Mitarbeiter als Autor eingeladen wird und den Zeitpunkt, an dem die Sitzung stattfindet. Dokumente sind typisch zu groß, um sie in einer Sitzung mit höchstens zwei Stunden Dauer zu prüfen. Der Projektleiter muss das Dokument in einzelne Teile zerlegen, jeder Teil wird in einer Sitzung geprüft. Er legt dazu den Umfang des Prüflings bei der Initialisierung fest und soll damit sicherstellen, dass die Sitzung nicht länger als zwei Stunden dauert. Zur Initialisierung verwendet er folgendes Kommando:

```
lasse_Reviewstattfinden(Dokument,Umfang,Moderator,Autor,Datum)
```

Der Spieler erhält daraufhin eine Meldung, die eine Identifikation der Reviewsitzung erlaubt, beispielsweise `ReviewMeeting_1`.

Er legt fest, welche Gutachter eingesetzt werden und nach welchen Aspekten die Gutachter das Dokument prüfen sollen. Es werden abhängig von der Anzahl der Aspekte mehrere Kommandos angeboten. Das Review kann über die Identifikation oder über das Datum der Sitzung benannt werden. Einige der Kommandos für das Spezifikationsreview sind:

```
lasse_an_Spez_Review_teilnehmen(Entwickler,Review)
lasse_an_Spez_Review_teilnehmen_1Aspekt(Entwickler, Review, Aspekt)
...
lasse_an_Spez_Review_teilnehmen_5Aspekte(Entwickler, Review, Aspekt1, ...,
Aspekt5)
```

```
lasse_an_Spez_Review_teilnehmen_Termin(Entwickler,Datum)  
...  
lasse_an_Spez_Review_teilnehmen_5Aspekte_Termin(Entwickler,Datum, Aspekt1,  
..., Aspekt5)
```

Die Gutachter bereiten sich einzeln vor. Der Spieler kann kontrollieren, wieviel Zeit die Gutachter für die Vorbereitung des Reviews aufgewendet haben. Er kann das Review abbrechen. In der Regel bricht aber der Moderator das Review ab, wenn Gutachter in der Sitzung nicht erscheinen oder nicht vorbereitet sind oder wenn der Autor nicht erscheint. In der Sitzung werden die Befunde aus der Vorbereitung diskutiert und besprochen. Der Moderator soll dafür sorgen, dass die Sitzung nicht länger als zwei Stunden dauert. Die Dauer der Sitzung beeinflusst der Spieler mit dem Umfang des Prüflings.

Zur Nachbereitung durch den Moderator gehört, dass er die Ergebnisse der Sitzung verteilt. Nach der Sitzung erhält der Spieler eine Nachricht mit der Empfehlung. Verkürzt habe ich, dass der Autor vom Moderator die Liste mit den Befunden bekommt, da im QS-Modell der Projektleiter bestimmt, wer den Prüfling korrigiert. Im Idealfall wählt der Spieler den Autor.

An den Eingriffen des Spielers zeigen sich die Möglichkeiten einer feineren Zeitgranularität: Der Spieler kann zum Beispiel noch am Tag der Sitzung Gutachter kontrollieren oder kurzfristig weitere Gutachter einladen.

### 5.3.1.2 Modellierung der Fragenkataloge und Aspekte

Fragenkataloge sind ein wichtiges Hilfsmittel (Frühauf et al., 2000). Sie enthalten Richtlinien und Fragen, um Schwachstellen aufzudecken. Die Fragenkataloge werden durch Aspekte strukturiert. Durch die Zuordnung der Aspekte zu verschiedenen Gutachtern wird mit dem Review ein systematischer Ansatz verfolgt (Porter et al., 1995). Einzelne Fragen werden nicht modelliert. Der Projektleiter ordnet den Gutachtern Aspekte für die Prüfung zu, eine detailliertere Modellierung der Aspekte ist nicht notwendig.

Die Darstellungsgüte ist ein wichtiges Kriterium, wenn ein Dokument beurteilt wird. Eine schlechte Struktur oder eine schwammige Darstellung führen zu Fehlern, wenn das Dokument als Vorgabe dient. Die Darstellungsgüte wird aber im QS-Modell nicht modelliert, das Fehlermodell kennt nur Fehler und fehlende oder unvollständige Anforderungen (Verluste). Damit kann nur nach inhaltlichen Kriterien geprüft werden. Eine Erweiterung des Fehlermodells um Darstellungsmängel kann im Rahmen dieser Arbeit nicht geleistet werden.

#### Fragenkatalog für die Spezifikation

Ludewig (1997) beschreibt die erwünschten Eigenschaften der Spezifikation, in Freedman und Weinberg (1982) wird auf S. 294 ein Fragenkatalog vorgestellt. Die Aspekte des Fragenkatalogs können den angestrebten Eigenschaften der Spezifikation zugeordnet werden (siehe Abbildung 17).

Eigenschaften (Ludewig, 1997)	BCS Requirements Checklist (Freedman, Weinberg, 1982)
Zutreffend	Correct, Traceable
Vollständig	Complete, Relevant
Konsistent	Consistent
Abstrakt	Free of Unwarranted Design Detail
Präzise, leicht verständlich	Precise, Unambiguous and Clear
Objektivierbar	Testable
leicht erstellbar, leicht verwaltbar	Managable
	Feasible

**Abb. 17:** Aspekte für die Spezifikation

Feasible bedeutet, dass die Anforderung mit den verfügbaren Ressourcen in der verfügbaren Zeit mit den vorgegeben Kosten implementiert werden kann. Ich gehe davon aus, dass diese Voraussetzungen im QS-Modell gegeben sind. Die Spezifikation kann unter den Aspekten Vollständigkeit, zutreffend, Konsistenz, Abstraktheit und Objektivierbarkeit geprüft werden. Bei der Objektivierbarkeit handelt es sich als einzigen Aspekt nicht um ein inhaltliches Kriterium. Sie kann dennoch überprüft werden, da Anforderungen, die nicht objektivierbar beschrieben sind, nicht vollständig in der Spezifikation wiedergegeben werden. Damit werden durch die Prüfung der Objektivierbarkeit zumindest Verluste festgestellt.

### Fragenkatalog für den Grobentwurf

Der Fragenkatalog für den Entwurf in Freedman und Weinberg (1982, S. 313) befasst sich mit Richtlinien für den Grobentwurf und mit der Vollständigkeit der Anforderungen. Dieser Fragebogen wird auch in Würthele (1995) empfohlen. Eine Checkliste in Hollocker (1990) ist durch Aspekte zur Designmethode, Datenstrukturen, Schnittstellen, Komplexität, die geforderte Genauigkeit, Vollständigkeit und Modularität gegliedert. Die Checkliste in Mandl-Striegnitz (1998) enthält die Aspekte Verfolgbarkeit zur Spezifikation, Konsistenz, Vollständigkeit und Effizienz.

Für den Fragenkatalog für den Grobentwurf orientiere ich mich an Hollocker (1990) und Mandl-Striegnitz (1998). Der Fragenkatalog wird durch die Aspekte Komplexität, Schnittstellen, Vollständigkeit, Modularität, Erweiterbarkeit, Konsistenz und Effizienz gegliedert. Die Designmethode ist durch das Modell oder den Projektleiter vorgegeben und sollte deshalb im Review nicht diskutiert werden. Den Aspekt Datenstrukturen ordne ich der Modularität zu, die Verfolgbarkeit gehört meiner Meinung nach zur Darstellungsgüte.

### 5.3.1.3 Erweiterung der Mitarbeiter

Damit wichtige Effekte des Reviews modelliert werden können, werden die Entitäten zur Modellierung der Mitarbeiter erweitert. Zu den Erfahrungen eines Mitarbeiters kommt die Erfahrung als Reviewmoderator hinzu. Sie wird durch die diskreten Werte KEINE, GERING, MITTEL und HOCH beschrieben. Dazu kommen noch drei Charaktereigenschaften:

- **Kritikfähigkeit:** Kritische Gutachter prüfen genau, sie sind als Gutachter geeignet.
- **Diplomatie:** In der Reviewsitzung ist wichtig, dass Befunde sachlich, prägnant und leicht verständlich vorgetragen werden (Freedman, Weinberg, 1982; Frühauf et al., 2000). Es soll verhindert werden, dass sich der Autor persönlich angegriffen fühlt. Bei einem diplomatischen Gutachter ist diese Fähigkeit stark ausgeprägt.
- **Selbstbewusstsein:** Wie stark sich der Autor auf persönliche Kritik einlässt, ist abhängig von seinem Selbstbewusstsein. Wenig selbstbewusste Mitarbeiter lassen sich leicht durch persönliche Angriffe frustrieren.

Diese Modellierung ist hypothetisch, ich habe darum die Einteilung in Klassen aus dem QS-Modell übernommen, die Charaktereigenschaften werden also durch die diskreten Werte KEINE, GERING, MITTEL und HOCH beschrieben.

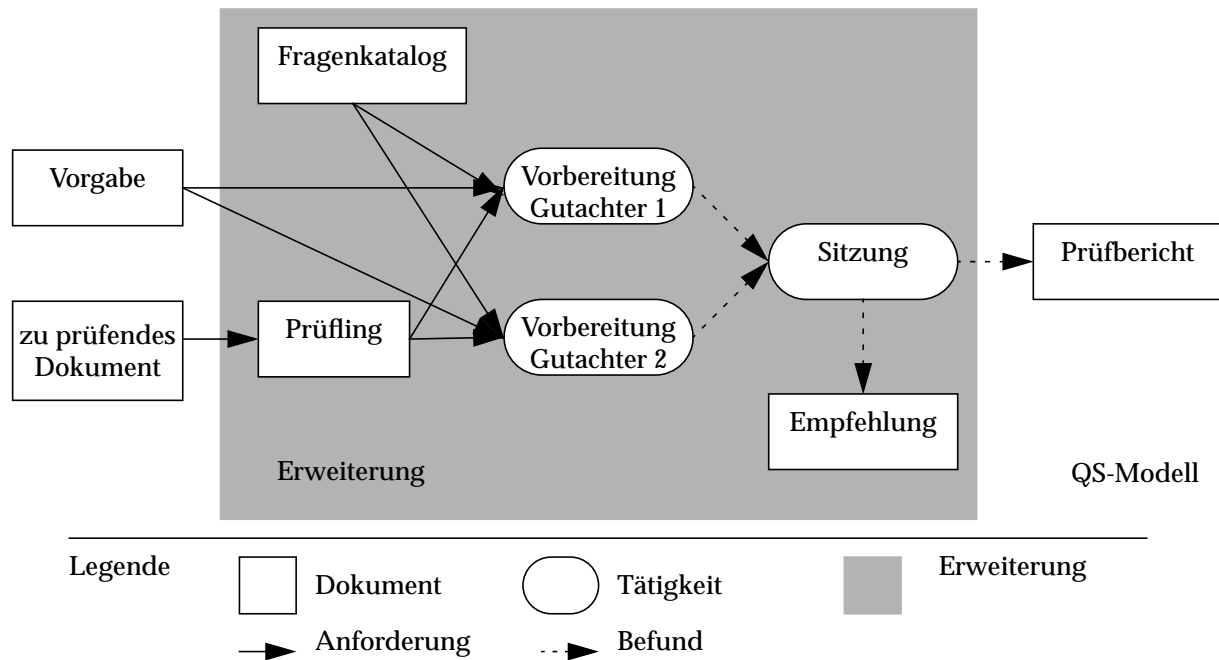
Um zu zeigen, welche Auswirkungen die Frustration des Autors hat, habe ich in Anlehnung an das QSVA-Modell von Dudler (2000) die Motivation der Entwickler eingeführt. Die Motivation bewegt sich auf einer Skala zwischen den Werten -5 und +5. Je höher der Wert, desto motivierter ist der Mitarbeiter. Im QSVA-Modell wird das QS-Modell um Verhaltensaspekte erweitert, die Motivation ist der zentrale Verhaltensaspekt. Das feingranulare Reviewmodell verfolgt ein anderes Ziel, die Motivation soll nur Auswirkungen einer Reviewsitzung zeigen und hat deshalb keine weiteren Einflüsse.

Da der Kunde am Review teilnehmen kann, könnte die Motivation des Kunden wie im QSVA-Modell dargestellt werden. Aus den folgenden Gründen modelliere ich die Motivation des Kunden nicht:

- Der Kunde befindet sich prinzipiell außerhalb der Systemgrenze. Er tritt im Modell nur für die Analyse, die Abnahme und im Spezifikations- und Handbuchreview in Erscheinung.
- Die Auswirkung der Teilnahme am Review wird bereits durch Beschwerden oder Lob des Kunden gezeigt (Abschnitt 5.3.1.8).
- Die Motivation des Kunden zu modellieren übersteigt den Rahmen der Arbeit.

### 5.3.1.4 Einbettung des Reviews

Die feinere Modellierung des Reviews ersetzt das bisherige Review im QS-Modell. Als Schnittstelle dienen die Entitäten des QS-Modells, hauptsächlich die Dokumente für die Vorgabe und den Prüfling und die beteiligten Entwickler.



**Abb. 18:** Einbettung des Reviews

Die Abbildung zeigt am Beispiel eines Reviews mit zwei Gutachtern, dass die Anforderungen für die Prüfung aus den vorhandenen Dokumenten stammen. Der Prüfling besteht aus einem Teil des Dokuments, den zu prüfenden Anforderungen. Diese Anforderungen werden mit den entsprechenden Anforderungen in der Vorgabe verglichen. In der Sitzung werden Befunde in den Prüfbericht geschrieben. Der Prüfbericht enthält die Befunde aus Reviews eines Dokuments so lange, bis das Dokument anhand des Befunds korrigiert wird. Damit das Modell nicht verändert werden muss, bleiben die Dokumente, Anforderungen und Befunde im Prüfbericht unverändert. Einschränkungen für die Variante ergeben sich aus der Modellierung der Anforderungen und der Mängel im QS-Modell.

Mängel werden im QS-Modell durch inhaltliche Fehler und Verluste modelliert. Fehler werden nach Fehlerarten unterschieden, die durch die Entstehungszeit der Fehler definiert sind. Es wird zwischen Analysefehlern, Grob- und Feinentwurfsfehlern, Implementierungsfehlern und Handbuchfehlern unterschieden. Zusätzlich zu Fehlern im Prüfling werden in Reviews im QS-Modell Referenzfehler, also Fehler im Vorgabedokument, entdeckt. Verluste sind als Differenz des Umfangs in AFP zwischen Vorgabeanforderung und Anforderung im Prüfling definiert. Im Review wird über Entdeckungsquoten berechnet, welche Anteile der Fehler und Verluste entdeckt werden können.

Im QS-Modell können im Test vermeintliche Fehler entdeckt werden. Wenn in der Vorgabe mehr Fehler als im Prüfling sind, erscheinen richtig implementierte Anforderungen falsch. Im Review im QS-Modell können Anforderungen in der Vorgabe erst nach dem

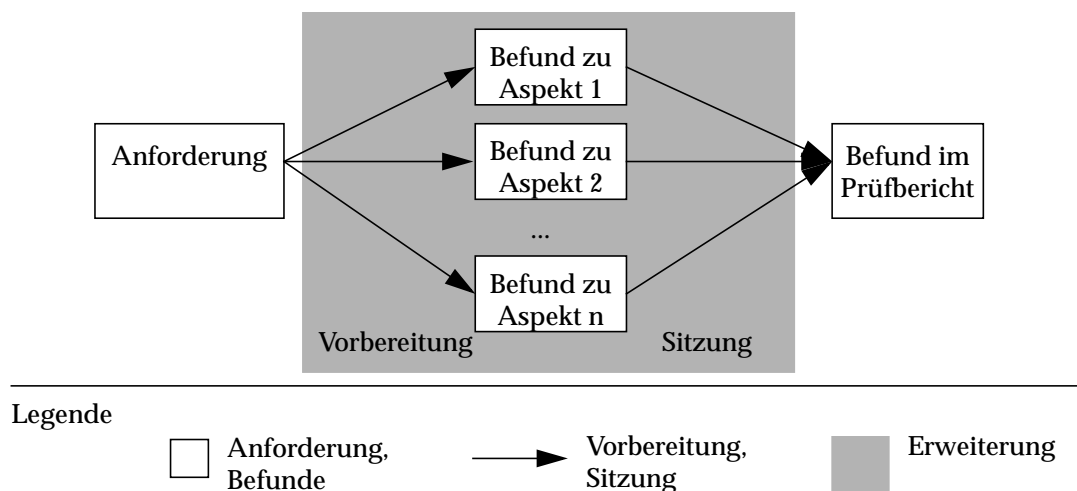


zweiten Review mehr Fehler enthalten. Da ein Dokument kaum durch drei Reviews geprüft wird, werden bisher in Reviews keine vermeintlichen Fehler gefunden. In der feingranularen Variante wird diese Modellierung beibehalten. Zusätzlich können aber die Gutachter den Prüfling falsch beurteilen und finden dadurch unabhängig von der Vorgabe vermeintliche Fehler. Die vermeintlichen Fehler sind nach Fehlerarten gegliedert. Sie spielen eine wichtige Rolle in der Reviewsitzung. Vermeintliche Fehler werden als Fehler in das Dokument eingefügt, sie erhöhen damit die Fehleranzahl des Dokuments.

Vermeintliche Verluste können im QS-Modell nicht modelliert werden - sie würden als nicht geforderte Funktionalität nur den Umfang des Dokuments steigern, sind aber vom Kunden nicht verlangt. Eine Unterscheidung zwischen nicht verlangter Funktionalität und der Funktionalität, die der Kunde fordert, ist mit der Modellierung des Umfangs im QS-Modell nicht möglich.

In der Reviewsitzung sollen die Gutachter jeden Befund bewerten, Frühauf et al. (2000) schlägt eine Klassifizierung in kritische Fehler, Haupt- und Nebenfehler vor. Das Fehlermodell lässt eine solche Bewertung aber nicht zu. Fehler werden im QS-Modell durch ihre Anzahl beschrieben. Nach Joos (1997, Seite 13) sind alle Fehler einer Fehlerart gleich schwerwiegend. Darum wird in der Variante erst am Schluss der Sitzung eine Bewertung für die Empfehlung durchgeführt.

Den Prüfbericht habe ich nicht verändert. Für jede geprüfte Anforderung wird ein Befund in den Prüfbericht geschrieben. In einem Befund werden alle entdeckten Fehler der Anforderung als Anteil der Fehler gespeichert, Verluste werden durch den Anteil an der Differenz des Umfangs zwischen Vorgabe und geprüfter Anforderung dargestellt.



**Abb. 19:** Anforderungen und Befunde

Um eine Prüfung nach Aspekten zu ermöglichen, werden die Befunde in der Vorbereitung nicht nur nach Anforderungen, sondern zusätzlich nach Aspekten unterschieden (Abbildung 19). Es entsteht daher nicht nur ein Befund für eine Anforderung, stattdes-

sen wird von einem Gutachter ein Befund für jeden Aspekt erzeugt. Der Befund enthält die in der Anforderung zu diesem Aspekt entdeckten Fehler und Verluste.

Diese Modellierung ermöglicht, dass in der Sitzung die Befunde der Gutachter nach den verschiedenen Aspekten unterschieden werden können. In der Sitzung werden die Befunde für jede Anforderung zuerst getrennt nach Aspekten zusammengefasst, bevor dann aus der Summe der Fehler und Verluste der Befund für den Prüfbericht erstellt wird. Abbildung 19 (Seite 51) soll diesen Zusammenhang verdeutlichen.

Die Befunde für einzelne Aspekte enthalten die absolute Anzahl Fehler und Verluste. Damit wird das Modell einfacher, da sonst in der Sitzung mit Anteilen von Anteilen gerechnet werden müsste.

### **5.3.1.5 Effekte bei der Initialisierung des Reviews**

Der Projektleiter bestimmt durch die Angabe des Umfangs in Seiten, wie groß der Prüfling in der Sitzung sein soll. Wie im QS-Modell wird ein Arbeitspaket (der Prüfling) anhand des Vorgabedokuments geschnürt. Das Arbeitspaket soll den vorgegebenen Umfang erreichen. Die Vorgabe wird verwendet, damit Anforderungen geprüft werden können, die beim Übertragen aus der Vorgabe in das zu prüfende Dokument vollständig verloren gegangen sind. Es werden wie im QS-Modell nur Anforderungen in den Prüfling gepackt, die nicht bereits geprüft wurden. Erst wenn ein Dokument vollständig geprüft und korrigiert wurde, können die Anforderungen einer weiteren Prüfung unterzogen werden.

Im QS-Modell werden Anforderungen, die ein Gutachter geschrieben hat, nicht in das Arbeitspaket übernommen und damit nicht geprüft. Dieser Effekt kann bei der feineren Modellierung entfallen, da sich Gutachter einzeln vorbereiten. Außerdem ist es möglich, dass bei der Initialisierung noch nicht bekannt ist, welche Gutachter eingeladen werden sollen. Der Spieler kann Gutachter noch zu einem späteren Zeitpunkt einladen, sie bereiten sich individuell vor. Er hat dadurch mehr Freiheit beim Vorgehen. Hier zeigt sich die größere Variabilität des feingranularen Modells (Hypothese H 8, Seite 18).

Durch den eingeladenen Autor wird bestimmt, welche Anforderungen bevorzugt werden. Können von ihm geschriebene Anforderungen verwendet werden, kommen diese Anforderungen in den Prüfling.

### **5.3.1.6 Effekte in der Vorbereitung**

Die Gutachter lesen den Prüfling mit einer vorgegebenen Geschwindigkeit, der Prüfleistung. Die Prüfleistung ist der Begriff aus dem QS-Modell, die Prüfleistung wird in der Literatur (Frühauf et al., 2000 oder Gilb, Graham, 1993) auch als Inspektionsrate bezeichnet und in Seiten pro Stunde angegeben. Sie wird abhängig von der verwendeten Notation auf AFP pro Stunde umgerechnet, da der Spieler den Umfang in Seiten angibt. Auf die Prüfleistung wirken sich die in diesem Abschnitt vorgestellten Effekte aus. Die Entdeckungsquoten für Fehler im Prüfling und im Referenzdokument, Verluste und ver-

meintliche Fehler sind durch das Modell vorgegeben. Sie definieren den Anteil der Fehler oder Verluste, die im Vergleich der Vorgabe mit dem Prüfling entdeckt werden. Die Entdeckungsquoten werden wie die Prüfleistung durch Effekte in der Vorbereitung beeinflusst.

Gilb und Graham (1993) beschreiben einen Zusammenhang zwischen Prüfleistung und Entdeckungsquote. Steigt die Prüfleistung, sinkt die Entdeckungsquote. Damit könnten die Entdeckungsquoten abhängig von der Prüfleistung modelliert werden. Ein Nachteil dieser Lösung ist die erschwerte Anpassung des Modells, etwa bei der Validierung. Das Modell würde unübersichtlich werden. Effekte wie die Erfahrung eines Entwicklers, die Prüfleistung und Entdeckungsquote in die gleiche Richtung beeinflussen, wirken sich zuerst positiv auf die Prüfleistung aus und damit negativ auf die Entdeckungsquote. Für die Entdeckungsquote muss dieser Effekt durch einen positiven Einfluss der Erfahrung überlagert werden. Ich habe mich deshalb entschieden, Prüfleistung und Entdeckungsquoten unabhängig voneinander zu modellieren.

### Der Einfluss des Umfangs

Der Umfang des Prüflings spielt im Review eine wichtige Rolle (Gilb, Graham, 1993; Kuhnle, 1998). Je größer der Prüfling ist, desto schneller lesen die Gutachter. Dafür wird aber weniger gründlich geprüft und deshalb werden weniger Fehler und Verluste entdeckt. Die folgende Abbildung zeigt Prüfleistung (P) und Entdeckungsquote (EQ) in Abhängigkeit vom Umfang.

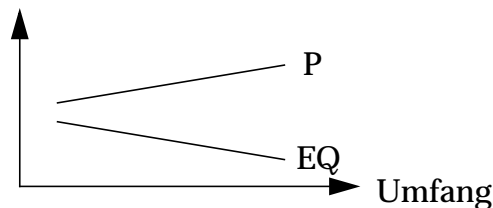
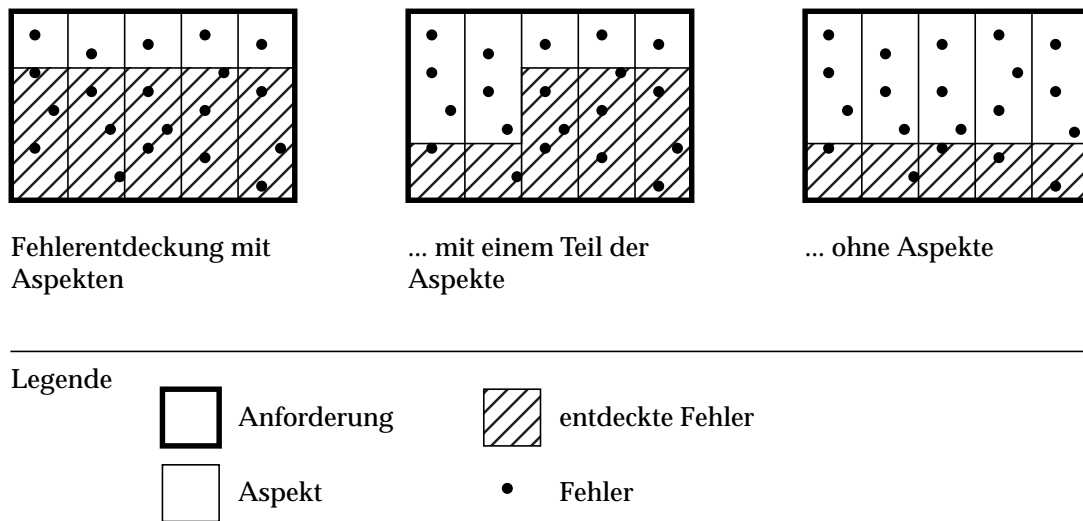


Abb. 20: Prüfleistung, Entdeckungsquoten und Umfang

### Die Prüfung mit Aspekten

Mit einer systematischen Prüfung durch Aspekte werden mehr Mängel im Prüfling entdeckt, jeder Fehler wird durch einen Aspekt gefunden (Porter et al., 1995). Abbildung 21 zeigt Fehler in einer Anforderung. Ein großer Anteil der Fehler wird bei der Prüfung mit Aspekten entdeckt. Wird ein Aspekt nicht untersucht, findet der Gutachter trotzdem einen kleinen Anteil der Fehler zu diesem Aspekt.



**Abb. 21:** Fehlerentdeckung mit und ohne Aspekte

Vereinfachend wird jedem Aspekt der gleiche Anteil Mängel aus der Anforderung zugeordnet. Nach diesem Prinzip werden Fehler, Verluste, vermeintliche Fehler und Fehler im Vorgabedokument entdeckt.

Die Prüfung nach Aspekten beeinflusst die Prüfleistung. Ich vermute, dass sich ein Gutachter um so länger vorbereitet, je mehr Aspekte betrachtet werden müssen. Obwohl dieser Einfluss nicht durch Literatur belegt werden kann, halte ich ihn für wichtig. Der Aufwand kann durch eine Verteilung der Aspekte auf Gutachter gesteuert werden.

### Der Einfluss der Gutachter

Neben dem Umfang des Prüflings und den betrachteten Aspekten spielen individuelle Eigenschaften der Gutachter eine Rolle. Die Erfahrung der Gutachter wird aus dem QS-Modell übernommen: Mit hoher Erfahrung lesen die Gutachter schneller und finden trotzdem mehr Fehler. Neben der Erfahrung mit der Notation und der Erfahrung mit Reviews spielt für das Spezifikationsreview die Erfahrung im Anwendungsbereich und für das Entwurfsreview die Entwurfserfahrung eine Rolle. Aus dem QS-Modell stammt auch der Einfluss der Einarbeitung. Gutachter, die sich noch einarbeiten müssen, brauchen länger und entdecken weniger Befunde.

Zusätzlich wird modelliert, dass kritische Gutachter mehr Mängel entdecken. Sie lesen den Prüfling aber auch langsamer. Kritische Gutachter sind daher geeigneter für Reviews.

### Weitere Effekte

Aus dem QS-Modell werden weitere Effekte übernommen. Mängel werden nur aus der Differenz zwischen Vorgabe und Prüfling entdeckt. Ohne Vorgabe werden keine Verluste und nur ein geringer Teil der Fehler gefunden. Bei einer mehrfachen Prüfung des

Dokuments wird es schwieriger, Mängel zu entdecken. Die Prüfungseffizienz und damit der Anteil der entdeckten Fehler, Verluste und vermeintlichen Fehler wird für jede durchgeführte Prüfung gesenkt.

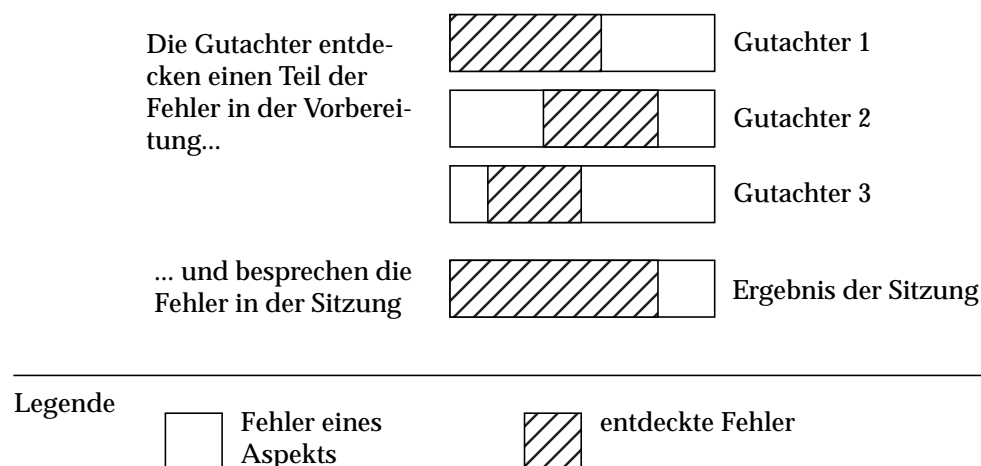
Im Gegensatz zum QS-Modell bereiten sich die Gutachter in der Variante 2 einzeln vor, die Anzahl der Gutachter hat keinen Einfluss auf die Vorbereitung.

### 5.3.1.7 Effekte in der Sitzung

#### Befunde aus der Vorbereitung

Die Dauer der Sitzung ist von der Anzahl der Befunde abhängig (Seaman, Basili, 1998; Gilb, Graham, 1993), im Modell von den entdeckten Fehlern und Verlusten. Die Diskussion selber wird nicht modelliert, ich habe verkürzt. Die Dauer ist von den Fehlern und Verlusten abhängig, die in der Sitzung besprochen werden. In der Vorbereitung mehrfach entdeckte Befunde zählen für die Dauer nicht doppelt, sondern fließen nur mit einem geringen Anteil in die Dauer ein. Die Leistung der Reviewsitzung wird in Fehler pro Minute und AFP pro Minute für entdeckte Fehler beziehungsweise Verluste angegeben.

Der größte Teil der Befunde im Review stammt aus der Vorbereitung. Dabei finden die Gutachter teilweise die gleichen Mängel. Die Mängel können durch das Fehlermodell aber nicht unterschieden werden, es wird daher mit Anteilen gerechnet. Nach Porter et al. (1997) findet bereits ein dritter Gutachter keine weiteren Mängel in der Vorbereitung. Nach Frühauf et al. (2000) sollen mindestens zwei Gutachter den gleichen Aspekt prüfen. Ich habe mich daher für die folgende Modellierung entschieden: Zwei Gutachter sind für die Prüfung eines Aspekts ausreichend. Ein weiterer Gutachter findet in der Vorbereitung keine neuen Fehler und Verluste, ein Gutachter alleine kann nur einen Teil der Mängel entdecken. Bereiten mehr als zwei Gutachter einen Aspekt vor, zählen die beiden Gutachter mit den meisten Befunden.



**Abb. 22:** Fehler aus der Vorbereitung

Im QS-Modell hängt dagegen der Anteil der entdeckten Fehler und Verluste von der Anzahl der Gutachter ab. Drei Gutachter finden 4% mehr Fehler als zwei Gutachter. Die Modellierung stützt sich auf Weller (1993). Für die Prüfung nach Aspekten habe ich mich an Porter et al. (1997) orientiert.

### **Befunde in der Sitzung**

Die Reviewsitzung bietet nach Porter und Johnson (1997) vor allem den Vorteil, dass vermeintliche Fehler erkannt werden. Ich vermute, dass abhängig von der Anzahl der teilnehmenden Gutachter mehr oder weniger vermeintliche Fehler aussortiert werden. Je mehr Gutachter an der Sitzung teilnehmen, desto weniger vermeintliche Fehler werden im Prüfbericht festgehalten.

Als weiteren Vorteil der Sitzung werden zusätzlich Mängel im Prüfling entdeckt (Porter et al., 1995). Allerdings gehen Befunde verloren, etwa wenn Gutachter nicht zu Wort kommen oder einen Befund vergessen (Porter et al., 1995; Freedman, Weinberg, 1982).

Wieviele Fehler und Verluste verloren gehen oder zusätzlich entdeckt werden, wird durch einen festen Anteil bestimmt. Möglich wäre hier noch ein Einfluss des Moderators – mit viel Erfahrung sorgt er dafür, dass alle Gutachter zu Wort kommen und für eine Atmosphäre, in der viele neue Mängel entdeckt werden. Diese Effekte werden verkürzt, um etwas einfachere Zusammenhänge zu modellieren (Kalajzic, 2001). Eine weitere Möglichkeit sind Zufallseffekte. Die Sitzung würde damit realistischer werden, insbesondere die Diskussion der Befunde. Da das Modell damit sehr feingranular wird, habe ich die Diskussion jedoch verkürzt.

### **Der Kunde im Spezifikationsreview**

Die Teilnahme des Kunden am Spezifikationsreview habe ich wie im QS-Modell verkürzt modelliert. Die Vorbereitung wird nicht modelliert, da Tätigkeiten des Kunden außerhalb der Systemgrenzen liegen. Der Kunde entdeckt einen Teil der Fehler und Verluste, die nicht von den Gutachtern gesehen wurden. Er identifiziert alle vermeintlichen Fehler, da er seine eigenen Anforderungen kennt. Die vom Kunden entdeckten Befunde fließen in die Dauer der Sitzung ein.

### **Die Eigenschaften der Entwickler**

Die Eigenschaften der Gutachter bestimmen die Dauer der Sitzung. Gutachter, die Erfahrung in Reviews gesammelt haben, neigen zu weniger unnötigen Diskussionen. Arbeitet sich ein Gutachter ein, verzögert er die Sitzung. Diese beiden Einflüsse stammen aus dem QS-Modell, dort wirken sie sich auf das gesamte Review aus. Eine weitere Eigenschaft der Gutachter beeinflusst die Dauer: Diplomatische Gutachter können sich besser ausdrücken und diskutieren, deshalb dauert die Sitzung nicht so lange.

Die Erfahrung des Moderators im Leiten der Sitzung hat einen starken Einfluss auf die Dauer der Sitzung. Er sorgt für einen geordneten und zügigen Ablauf der Diskussion in der Sitzung.

### **Der Aktuar in der Sitzung**

Die Dauer der Sitzung ist abhängig vom Einsatz des Aktuars. Ein einzelner Gutachter ist mit dieser weiteren Aufgabe überfordert. Der Moderator ist als Aktuar ebenfalls nur wenig geeignet, da er die Sitzung leitet. Der Moderator oder ein einzelner Gutachter als Aktuar verzögern deshalb die Sitzung (Frühauf, Ludewig, Sandmayr, 2000, Seite 85). Möglich ist der Einsatz des Autors oder eines unbeteiligten Entwicklers als Aktuar.

### **Der Autor in der Sitzung**

Die Diskussion der Befunde in der Sitzung soll sachlich bleiben (Frühauf et al., 2000; Freedman, Weinberg, 1982). Undiplomatische Gutachter greifen aber den Autor persönlich an und frustrieren ihn, wenn er wenig Selbstbewusstsein hat. Ein Moderator mit viel Erfahrung als Leiter der Sitzung lässt eine unsachliche und persönliche Kritik nicht zu. Bei persönlichen Angriffen sinkt die Motivation des Autors, abhängig von der Anzahl der besprochenen Fehler und Verluste und dem Selbstbewusstsein des Autors. Damit die Motivation aber nicht nur absinkt, vergisst der Autor wie in Dudler (2000) mit der Zeit die Kränkung.

### **Abbruch der Sitzung**

Am Anfang der Sitzung bricht der Moderator das Review ab, wenn mehr als die Hälfte der eingeladenen Gutachter nicht erscheint oder unvorbereitet ist. Gutachter erscheinen nicht, wenn sie bereits in einer anderen Sitzung sind. Ein Gutachter bereitet sich nicht vor, wenn er zu wenig Zeit für die Vorbereitung hat. Hat ein Gutachter eine Anforderung selber geschrieben, prüft er diese nicht. Der Moderator prüft im Modell die Vorbereitung anhand der investierten Zeit. Vorbereitete Gutachter bleiben nach einem Abbruch vorbereitet. Sie haben sich die Befunde notiert. Wird das Review wiederholt, bereiten sie sich nicht noch einmal vor.

Wenn der eingeladene Autor nicht am Dokument, aus dem der Prüfling stammt, beteiligt war, nimmt er nicht an der Sitzung teil. Die Sitzung wird abgebrochen. Fehlt der Moderator, weil er gleichzeitig eine andere Sitzung leitet, findet die Sitzung nicht statt.

Im QS-Modell findet eine Sitzung immer statt, selbst wenn alle Gutachter nicht vorbereitet sind. Im QS-Modell können sich Gutachter nicht vorbereiten, wenn sie zuviel Zeit für das Review aufwenden müssen oder alle Anforderungen von einem Gutachter geschrieben wurden. Die Gutachter nehmen im QS-Modell auf jeden Fall an der Sitzung teil.

### **Das Ende der Sitzung**

Wenn alle Befunde besprochen wurden, wird die Empfehlung aufgrund der Fehlerdichte im Prüfling gegeben. Es zählen die im Review entdeckten Fehler. Die Empfehlung kann nur verkürzt modelliert werden. "*Nicht akzeptieren*" bedeutet für die Nacharbeit eine vollständige Überarbeitung des Prüflings. Im QS-Modell ist nur eine Korrektur modelliert, der Prüfling kann nicht neu geschrieben werden. Bei der Empfehlung "*Überarbeiten*" kann eine Prüfung der Korrektur durch einen Gutachter oder den Moderator des Reviews erfolgen oder durch ein weiteres Review. Die Prüfung durch einen Entwickler ist im QS-Modell nicht modelliert.

Ich habe mich an den allgemeinen Regeln für die Reviewsitzung (Frühauf, 2000) orientiert. Der Spieler bekommt drei Empfehlungen.

Empfehlung	Nacharbeit
akzeptieren	(Korrektur)
überarbeiten	Korrektur (und Review)
nicht akzeptieren	Korrektur und Review

**Abb. 23:** Empfehlungen des Reviewmodells

Dauert die Sitzung länger als zwei Stunden, wird sie von einem erfahrenen Moderator vertagt. Die Sitzung wird um mindestens einen Tag auf einen Termin verschoben, an dem keine weitere Sitzung stattfindet. Wird die Sitzung nicht vertagt, sinkt die Konzentration der Teilnehmer: Es werden weniger Befunde der Vorbereitung besprochen, weniger neue Befunde entdeckt und weniger vermeintliche Fehler identifiziert. Dieser Effekt trifft auch auf den Kunden zu.

#### **Nachbereitung des Moderators**

Die Nachbereitung des Moderators, etwa um die Reviewsitzung zu protokollieren, die Empfehlung weiterzugeben oder die Kontrolle der Nacharbeit habe ich stark verkürzt (siehe Frühauf et al., 2000, Seite 104). Es wird nur die Dauer von 30 Minuten zum Aufwand für das Review berechnet.

#### **5.3.1.8 Effekte auf den Tagesablauf des Projektleiters**

Nach der Sitzung bringt der Moderator dem Projektleiter die Empfehlung vorbei. Im Modell wird eine Nachricht ausgegeben und Zeit des Projektleiters verbraucht. Konnte die Sitzung durchgeführt werden, wird an den Spieler eine entsprechende Nachricht ausgegeben. Der Spieler bekommt den geprüften Umfang, die Anzahl der gefundenen Fehler und Verluste, die Dauer der Sitzung und die Empfehlung mitgeteilt. Im QS-Modell wird nur eine Nachricht mit dem Umfang, der Anzahl der gefundenen Fehler und gefundenen Verlusten angezeigt. Musste der Moderator die Sitzung abbrechen oder vertagen, wird diese Tatsache und der Grund an den Spieler weitergegeben. Die Störung dauert nach einem Abbruch länger als bei einem erfolgreichen Review, da der Grund des Abbruchs mit dem Projektleiter diskutiert wird.

Wurde der Autor in der Sitzung stark persönlich angegriffen, dann beschwert er sich darüber beim Projektleiter.

Der Kunde kann sich nach einem Review beim Projektleiter melden. Wenn er am Spezifikationsreview teilgenommen hat, kann er die Spezifikation mit seinen Ansprüchen an das Dokument vergleichen. Im Modell können dazu die Zielvorgaben, die bereits im QS-Modell definiert werden können, verwendet werden.



### 5.3.2 Tagesablauf des Projektleiters

Der Tagesablauf des Projektleiters wird durch den Zeitverbrauch der Eingriffe zur Steuerung des Projekts und des Reviewprozesses bestimmt. Neben den Meldungen aus dem Review werden zwei weitere Einflüsse auf den Tagesablauf modelliert: Der Projektleiter kann ein Treffen mit einem Mitarbeiter vereinbaren. Der Mitarbeiter berichtet, was er gerade macht, gemeinsam beurteilen sie den Fortschritt der Dokumente, die der Mitarbeiter bearbeitet.

Erstellt oder korrigiert ein unerfahrener Entwickler ein Dokument, dann hat er dazu Fragen und stört er den Projektleiter ungefähr einmal pro Woche. Der Zeitpunkt der Störung wird durch Zufall bestimmt. Muss ein unerfahrener Entwickler ein Review leiten, dann hat er ebenfalls Fragen. Er kommt an einem zufälligen Zeitpunkt zwischen der Initialisierung des Reviews und der Reviewsitzung vorbei.

Der Zeitverbrauch der Meldungen der Sitzung, der Störungen und der Sitzungen und damit der Einfluss auf den Tagesablauf kann erst bei einer feinen Zeitgranularität beobachtet werden.

### 5.3.3 Verhalten der Mitarbeiter

#### Dringende und parallele Tätigkeiten

Im QS-Modell kann ein Entwickler gleichzeitig nur eine Tätigkeit durchführen, aber an mehreren Reviews teilnehmen. Es ist nicht möglich, dass ein Entwickler gleichzeitig entwirft und die Spezifikation korrigiert. In der feingranularen Variante ist möglich, dass ein Entwickler mehrere Aufgaben hat und alle zwei Stunden durch Zufall eine Aufgabe auswählt und durchführt. Die Modellierung von parallelen Tätigkeiten entspricht eher der Realität und lässt ein weniger sequentielles Vorgehen des Spielers zu. Es entstehen stochastische Prozesse, das Spiel wird weniger vorhersehbar.

Zusätzlich kann der Spieler in der Variante verlangen, dass eine Tätigkeit sofort und dringend durchgeführt werden soll. Der Entwickler macht dann nichts anderes mehr, selbst die Vorbereitung für ein Review unterbleibt. Grundsätzlich sind die Entwickler aber bereit, sich vorzubereiten und bevorzugen daher die Prüfung eines Dokuments vor Erstellung oder Korrektur. Damit wird die Reihenfolge für die einzelnen Tätigkeiten der Entwickler festgelegt: Die Entwickler erscheinen auf jeden Fall in der Sitzung. Einzige Ausnahme sind mehrere Sitzungen für denselben Entwickler zur gleichen Zeit. Die übrige Zeit widmen sie dringenden Tätigkeiten, müssen sie nichts dringendes erledigen, bereiten sie sich auf Reviews vor. Erst dann kommen andere Aufgaben an die Reihe.

#### Arbeitszeit der Mitarbeiter

Feiner eingreifen kann der Spieler, indem er für einzelne Entwickler Überstunden anordnet. Ein Entwickler kann eine oder zwei Überstunden leisten, er arbeitet dann länger. Überstunden sollen aber nur für kurze Zeit eingesetzt werden. Metzger und Boddie (1996) beschreiben, dass höchstens drei Wochen lang mehr gearbeitet wird. Werden

Überstunden für eine zu lange Zeit angeordnet, wird sogar weniger gearbeitet als ohne Überstunden.

## 5.4 Quantifizierung

In diesem Abschnitt werden die Effekte und Zusammenhänge der feingranularen Variante quantifiziert.

### 5.4.1 Quantifizierung des Reviews

#### 5.4.1.1 Voraussetzungen

Für die Quantifizierung gelten die folgenden Voraussetzungen:

- Die Quantifizierung des QS-Modells gilt als realistisch und wurde validiert. Die feingranulare Variante soll insgesamt die gleichen Ergebnisse ergeben wie Reviews im QS-Modell und orientiert sich daher an diesen Werten.
- In den Reviews im QS-Modell wird bereits nach Aspekten geprüft, das Modell ist an dieser Stelle aber verkürzt (Drappa, 2000, Seite 67).
- Im QS-Modell wird die Rolle des Autors, des Aktuars und des Moderators im Review nicht modelliert. Ich gehe davon aus, dass der Aufwand im QS-Modell nur für die Gutachter berechnet wird.
- Für die Quantifizierung wird als Notation der Dokumente deutsch angenommen. Die Quantifizierung wird an Projekte mit einem Umfang von 200 AFP angepasst.
- Einflüsse werden häufig durch Funktionen dargestellt. In den meisten Fällen sind diese Einflüsse kaum beschrieben, sie werden deshalb durch lineare Funktionen angenähert. Damit wird die Modellierung etwas einfacher (Kalajzic, 2001).

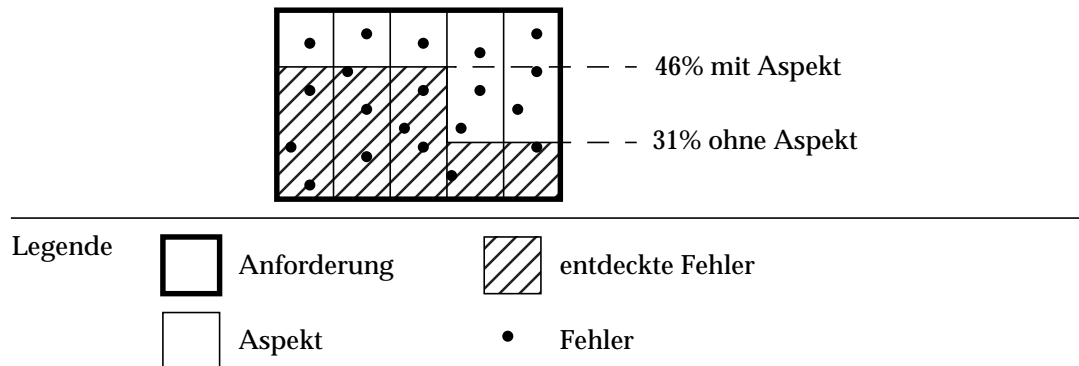
#### 5.4.1.2 Entdeckung von Fehlern und Verlusten

Fehler und Verluste werden nicht nur in der Vorbereitung entdeckt, sondern auch in der Sitzung. In der Sitzung erfolgt auch die Besprechung der in der Vorbereitung entdeckten Befunde. Die Quantifizierung wird daher in diesem Abschnitt für Vorbereitung und Sitzung zusammen vorgestellt.

Porter et al. (1995) haben einen systematischen Ansatz für die Prüfung mit Reviews untersucht. Dabei werden Szenarien verwendet, die dem Gutachter einen bestimmten Blickwinkel vorgeben. Miller et al. (1998) haben dieses Experiment wiederholt und die Vorbereitung der Gutachter genauer untersucht. Ich übertrage für die Quantifizierung die Ergebnisse auf die Prüfung mit Aspekten und betrachte vorerst die Vorbereitung einzelner Gutachter: Im Experiment werden drei Gutachter eingesetzt, jeder Gutachter prüft ein Szenario. Zum Vergleich finden im Experiment Reviews anhand von Checklisten oder ohne Hilfsmittel statt. Ohne Prüfung durch Szenarien deckt ein Gutachter 31% der im Prüfling enthaltenen Fehler auf. Durch die Prüfung mit einem Szenario findet ein Gutachter insgesamt 36% der Fehler, also 5% mehr. Porter et al. (1995) haben nachgewie-

sen, dass bei der Prüfung mit einem Szenario trotzdem Fehler entdeckt werden, die anderen Szenarien zugeordnet sind.

In Abschnitt 5.3.1.6 (Seite 53) ist beschrieben, dass jeder Fehler im Prüfling einem Aspekt zugeordnet ist - die Fehler werden gleichmäßig auf die Aspekte verteilt. Prüft ein Gutachter einen Aspekt nicht, so findet er 31% der diesem Aspekt zugeordneten Fehler. Damit bei drei Aspekten wie in Miller et al. (1998) bei der Prüfung eines Aspekts 36% der Fehler insgesamt entdeckt werden, findet der Gutachter zu einem Aspekt 46% der Fehler.

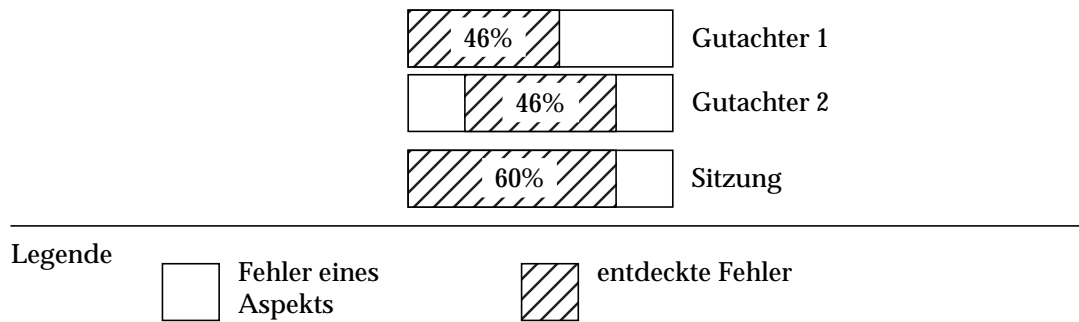


**Abb. 24:** Entdeckungsquote mit und ohne Aspekte

Ein Beispiel soll diesen Zusammenhang verdeutlichen. Eine Anforderung der Spezifikation enthält 10 Fehler. In der Vorgabe, den Analysenotizen, sind - für das Beispiel vereinfacht - keine Fehler. Für jeden Aspekt könnte der Gutachter 10 Fehler / 5 Aspekte = 2 Fehler entdecken. Er prüft unter einem Aspekt und entdeckt dafür 2 Fehler \* 46% = 0,92 Fehler. Ein anderer Aspekt wird nicht untersucht, der Gutachter entdeckt nur 2 Fehler \* 31% = 0,62 Fehler.

In der Sitzung werden die in der Vorbereitung entdeckten Fehler für jede Anforderung aspektweise zusammengefasst. Die in der Vorbereitung entdeckten Fehler werden durch ihre Anzahl dargestellt. Es werden die beiden Gutachter mit den meisten Befunden zu einem Aspekt berücksichtigt, ein dritter Gutachter entdeckt in der Vorbereitung keinen neuen Fehler.

Einzelne Fehler können nicht unterschieden werden, doppelte Fehler werden über Anteile berechnet. Da ich davon ausgehe, dass auch im QS-Modell mit Aspekten geprüft wird, sollen insgesamt 60% der Fehler entdeckt werden. Für die Quantifizierung des Anteils, den der zweite Gutachter zusätzlich findet, betrachte ich die Anteile aus der Quantifizierung der Vorbereitung: Finden zwei Gutachter jeweils 46% der Fehler, überschneiden sich ihre entdeckten Befunde um 70%.



**Abb. 25:** Fehler in der Vorbereitung und in der Sitzung

Vom zweiten Gutachter fließen also 30% der Fehler ein. Wieder soll ein Beispiel den Zusammenhang zeigen: Es haben zwei Gutachter einen Aspekt einer Anforderung vorbereitet. Dabei hat jeder Gutachter zu diesem Aspekt 0.92 Fehler gefunden. In der Sitzung werden die Befunde für diesen Aspekt zusammengerechnet. Für den Aspekt zählen insgesamt  $0.92 \text{ Fehler} + 0.92 \text{ Fehler} * 30\% = 1.196 \text{ Fehler}$ , ungefähr 60% von den 2 Fehlern des Aspekts. Hat ein dritter Gutachter weniger oder gleich viel Fehler zum Aspekt in der Anforderung entdeckt, wird er nicht betrachtet. Hat der dritte Gutachter dagegen mehr Fehler entdeckt, wird anstatt einem der beiden anderen Gutachter der dritte Gutachter berücksichtigt.

Diese Quantifizierung gilt für die verschiedenen Fehlerarten, die in den Reviews der unterschiedlichen Dokumente entdeckt werden. Verluste werden mit den gleichen Zahlen wie die Fehler entdeckt, da die Verlustentdeckung bereits im QS-Modell wie die Fehlerentdeckung quantifiziert ist.

Die Entdeckung von Referenzfehlern orientiert sich am QS-Modell. Dort werden insgesamt 15% der Fehler im Referenzdokument identifiziert. Referenzfehler werden nach dem gleichen Prinzip wie andere Fehler entdeckt, in der Vorbereitung findet ein Gutachter daher mit Aspekt 11%, ohne Aspekt 7% der Referenzfehler. Referenzfehler von zwei Gutachtern überdecken sich zu 70%. Vermeintliche Fehler werden nach einem Experiment von Porter und Johnson (1997) quantifiziert. In diesem Experiment hat sich gezeigt, dass 22% der Befunde eines Gutachters als vermeintliche Befunde eingestuft werden müssen. In der Sitzung wird ein Teil der vermeintlichen Fehler von den Teilnehmern als solche erkannt und aussortiert (siehe Abschnitt 5.4.1.5).

### 5.4.1.3 Dauer der Vorbereitung und der Sitzung

Die Dauer der Vorbereitung und der Sitzung ist so quantifiziert, dass der Aufwand der Gutachter insgesamt dem Aufwand des Reviews im QS-Modell entspricht. Die beiden Parameter, die die Dauer bestimmen, werden darum zusammen quantifiziert. Die Prüfleistung für die Vorbereitung stammt aus Frühauf et al. (2000) und ist für das Spezifikationsreview auf 10 Seiten pro Stunde gesetzt. Für das Entwurfsreview wurde der Wert an das QS-Modell angepasst, der Aufwand ist höher, es ergibt sich eine Prüfleistung von 8.67 Seiten.

Die Dauer der Reviewsitzung wird durch die Reviewleistung festgelegt. Die Reviewleistung für Spezifikations- und Entwurfsreviews ergibt sich aus dem Aufwand insgesamt im QS-Modell und der Prüfleistung der Vorbereitung. In der Sitzung des Spezifikationsreviews ergeben sich für Fehler und Verluste die beiden folgenden Werte:

$$P_{\text{Spez,Fehler}} = 0.21 \text{ Fehler pro Minute}$$

$$P_{\text{Spez,Verluste}} = 0.21 \text{ Fehler pro Minute}$$

Da die Verlustentdeckungsquote auf die Fehlerentdeckungsquote gesetzt wird, habe ich die Reviewleistung für die Verluste auf die Leistung für die Fehler gesetzt. Im Entwurfsreview ergeben sich die folgenden Zahlen.

$$P_{\text{Entwurf,Fehler}} = 0.17 \text{ Fehler pro Minute}$$

$$P_{\text{Entwurf,Verluste}} = 0.17 \text{ Fehler pro Minute}$$

In Bourgeois (1996) wird für die Spezifikation 0.18 Fehler pro Minute gemessen, für den Entwurf 0.16 Fehler pro Minute. Die aus dem Aufwand im QS-Modell und der Prüfleistung errechneten Zahlen erscheinen also plausibel.

#### 5.4.1.4 Effekte in der Vorbereitung

##### Der Umfang des Prüflings

Der Umfang wirkt sich stark auf die Vorbereitung aus: Bei großem Umfang lesen die Gutachter schneller und finden weniger Fehler. Für den Umfang gibt es unterschiedliche Angaben. Frühauf et al. (2000) empfiehlt 50 Seiten, Weller (1993) weniger als 20 Seiten. Im QS-Modell werden in einer Sitzung des Spezifikationsreviews 60 AFP (26 Seiten in deutsch) und im Entwurfsreview 40 AFP (18 Seiten auf deutsch) betrachtet. Als Referenzwert für den Umfang des Prüflings verwende ich 26 Seiten aus den folgenden Gründen:

- Der Spieler muss bei der Initialisierung des Reviews den Umfang angeben. Unterschiedliche Größen für Spezifikation und Entwurf verwirren den Spieler vermutlich.
- Ein Umfang von 50 Seiten führt zu einer starken Verkürzung der Prüfung, bei Projekten mit 200 AFP werden anstatt vier Sitzungen nur noch zwei Sitzungen benötigt. Der Aufwand des QS-Modells kann so nicht erreicht werden, da die Dauer der Sitzungen auf zwei Stunden beschränkt ist.

In Gilb und Graham (1993) werden übliche Durchschnittswerte für die Prüfleistung zwischen 5 und 20 Seiten pro Stunde angegeben. Als Grenzwert für einen sehr kleinen Umfang halbiere ich darum die Prüfleistung von 10 Seiten pro Stunde auf 5 Seiten pro Stunde. Der Einfluss wird als Funktion über den Umfang dargestellt und vereinfacht als lineare Funktion modelliert. Abbildung 26 zeigt die Funktion und ihre Grenzwerte.

Einfluss: $P' = P * f(U)$	P: Prüfleistung P': angepasste Prüfleistung U: Umfang (in AFP)
Gegebene Werte: $f(0) = 0.5$ $f(60) = 1$	Funktion: $f = 0.5 + 0.0083^{1/_{AFP}} * U$

**Abb. 26:** Einfluss des Umfangs auf die Prüfleistung

Der Einfluss des Umfangs auf die Entdeckungsquoten konnte anhand eines Diagramms aus einem Praxisbericht in Gilb und Graham (1993) auf Seite 333 abgeschätzt werden. Bei ungefähr 220 Seiten (500 AFP) sinkt die Anzahl der entdeckten Fehler auf ein Drittel gegenüber einem Umfang von rund 26 Seiten (60 AFP). Wieder wurde vereinfachend eine lineare Funktion aufgestellt:

$EQ' = EQ * f(U)$	EQ: Entdeckungsquote EQ': angepasste Entdeckungsquote U: Umfang (in AFP)
Gegebene Werte: $f(60) = 1$ $f(500) = 0.3$	Funktion: $f = 1.0955 - 0.0016^{1/_{AFP}} * U$

**Abb. 27:** Einfluss des Umfangs auf die Entdeckungsquoten

### Einfluss der Aspekte

Die Prüfung mit Aspekten beeinflusst die Dauer der Vorbereitung, der Gutachter braucht um so länger, je mehr Aspekte zu prüfen sind. Im Referenzlauf werden zwei Gutachter in der Spezifikation eingesetzt. Jeder Gutachter sollte also alle Aspekte prüfen, damit jeder Aspekt von zwei Gutachtern betrachtet wird. Die Prüfleistung bleibt bei der Prüfung mit fünf Aspekten gleich, da ich voraussetze, dass auch im QS-Modell mit Aspekten begutachtet wird. Prüft ein Gutachter unter keinem Aspekt, unterstelle ich eine um den Faktor 1.5 höhere Prüfleistung:

$$P'_{\text{Spez}} = P_{\text{Spez}} * (1.5 - 0.1 * \#\text{Aspekte})$$

Im Entwurfsreview werden dagegen im Referenzlauf drei Gutachter eingesetzt. Jeder Gutachter muss von den sieben Aspekten für den Entwurf nur fünf betrachten. Wieder bleibt daher bei fünf Aspekten die Prüfleistung gleich:

$$P'_{\text{Entwurf}} = P_{\text{Entwurf}} * (1.5 - 0.1 * \#\text{Aspekte})$$

### Einfluss der Gutachter

Der Einfluss der Erfahrung der Gutachter auf die Prüfleistung und die Entdeckungsquote wird aus dem QS-Modell übernommen. Die Einarbeitung wirkt sich wie im QS-

Modell aus. Die Kritikfähigkeit wirkt sich auf die Prüfleistung und die Entdeckungsquote aus. Die Werte sind hypothetisch:

Kritikfähigkeit	KEINE	GERING	MITTEL	HOCH
Prüfleistung	0.9	0.95	1.0	1.05
Entdeckungsquote	0.8	0.9	1.0	1.1
Entdeckungsquote für vermeintliche Fehler	1.2	1.1	1.0	0.8

**Abb. 28:** Einfluss der kritischen Gutachter

### 5.4.1.5 Effekte in der Sitzung

#### Befunde in der Sitzung

In der Sitzung treten verschiedene Effekte auf. Ein bestimmter Anteil der Befunde geht verloren. Nach Porter et al. (1995) gehen zwischen 6.8% bis 7.7% der in der Vorbereitung entdeckten Fehler verloren. Da in den Sitzungen im diesem Experiment kein erfahrener Moderator anwesend war, habe ich den Anteil etwas verkleinert und auf 5% festgelegt. Dafür werden aber neue Befunde entdeckt. Im gleichen Experiment von Porter werden nur 3.9% bis 4.7% der Fehler neu entdeckt. In einer anderen Studie (Porter, Johnson, 1997) werden 10% angegeben, diesen Wert habe ich für die Quantifizierung übernommen.

Von den vermeintlichen Fehlern, die in der Vorbereitung entdeckt wurden, wird in der Sitzung ein Teil als solche erkannt und aussortiert. Nach Porter und Johnson (1997) sinkt der Anteil der vermeintlichen Fehler von 22% auf 5% bei einer Sitzung mit drei Teilnehmern. In der Quantifizierung entdeckt daher jeder Gutachter 25% der vermeintlichen Fehler und verhindert damit, dass diese in den Prüfbericht geschrieben werden.

#### Eigenschaften der Entwickler

Die Eigenschaften der Entwickler wirken sich auf die Dauer der Sitzung aus. Die Erfahrung mit Reviews übernehme ich aus dem QS-Modell, den Einfluss der diplomatischen Fähigkeiten eines Entwicklers wird abgeschätzt:

Einfluss auf die Reviewleistung	KEINE	GERING	MITTEL	HOCH
Diplomatie der Gutachter	0.8	0.9	1.0	1.1

**Abb. 29:** Einfluss diplomatischer Gutachter

Vor allem spielt aber die Erfahrung des Moderators eine wichtige Rolle, da er die Sitzung leitet (Abbildung 30).

Einfluss auf die Reviewleistung	KEINE	GERING	MITTEL	HOCH
Erfahrung als Moderator	0.6	0.8	1.0	1.25

**Abb. 30:** Einfluss des Moderators

Wird die Rolle des Aktuars dem Moderator oder einem einzigen Gutachter zugewiesen, verzögert das die Sitzung. Den Einfluss des Moderators habe ich stärker quantifiziert, da der Gutachter nur bei eigenen Befunden zuviel zu tun hat.

Aktuar:	Moderator	ein Gutachter
Reviewleistung	0.7	0.9

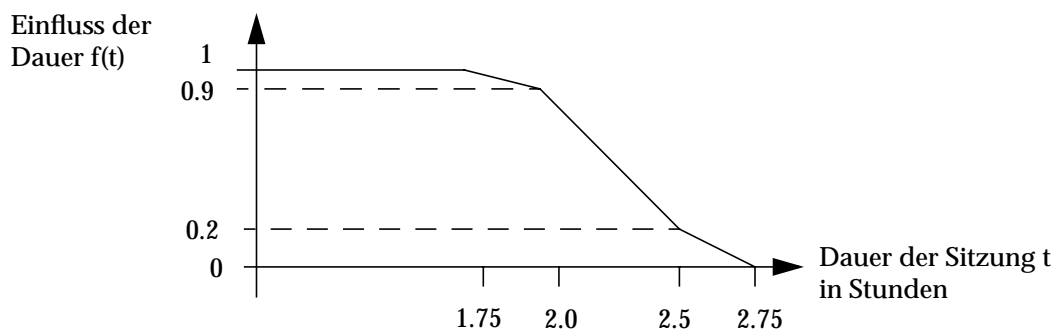
**Abb. 31:** Einfluss der Rolle des Aktuars

**Einfluss der Sitzungsdauer**

Die Dauer der Sitzung sollte auf zwei Stunden beschränkt werden. Ein Moderator mit mittlerer oder hoher Erfahrung wird daher die Sitzung nach zwei Stunden abbrechen. Ab zwei Stunden sinkt die Konzentration der Teilnehmer stark. Für die Quantifizierung verwende ich eine abschnittsweise definierte Funktion. Damit treten keine Sprünge auf, es soll keine Metaregel (Kalajzic, 2001) verletzt werden.

- Bis 1.75 Stunden gibt es keinen Einfluss auf die Ergebnisse der Reviews.
- Zwischen 1.75 Stunden und 2 Stunden ändern sich die Ergebnisse leicht.
- Zwischen 2 Stunden und 2.5 Stunden fällt die Konzentration stark ab.
- Zwischen 2.5 - 2.75 Stunden geht die Konzentration gegen Null.
- Ab 2.75 können sich die Entwickler überhaupt nicht mehr konzentrieren.

Die Konzentration wird indirekt modelliert, indem sich die Dauer der Sitzung auf die in der Sitzung zusätzlich entdeckten Fehler und Verluste und auf die in der Sitzung identifizierten vermeintlichen Fehler auswirkt. Die Funktionen habe ich mit den folgenden Grenzwerten festgelegt:



**Abb. 32:** Funktion für die Dauer der Sitzung



Zusätzlich zu diesen Auswirkung werden weniger Befunde besprochen. Die Anzahl der Fehler und Verluste halbiert sich ab 2.75 Stunden. Dazu wird der Einfluss der Funktion halbiert.

### Einfluss auf den Autor

Ein Fehler oder ein Verlust kann die Motivation des Autors beeinflussen. Jeder Fehler oder Verlust senkt die Motivation um 0.1 Motivationspunkte. Je nach Diplomatie des Gutachters wird diese Motivationsverringerung abgeschwächt. Der Moderator verhindert die Angriffe und ein selbstbewusster Autor wird sich nur wenig beeindrucken lassen.

	KEINE	GERING	MITTEL	HOCH
Diplomatie des Gutachters	1.0	0.7	0.4	0.1
Erfahrung als Moderator	1.0	0.7	0.4	0.1
Selbstbewusstsein des Autors	1.0	0.7	0.4	0.1

**Abb. 33:** Frustration des Autors

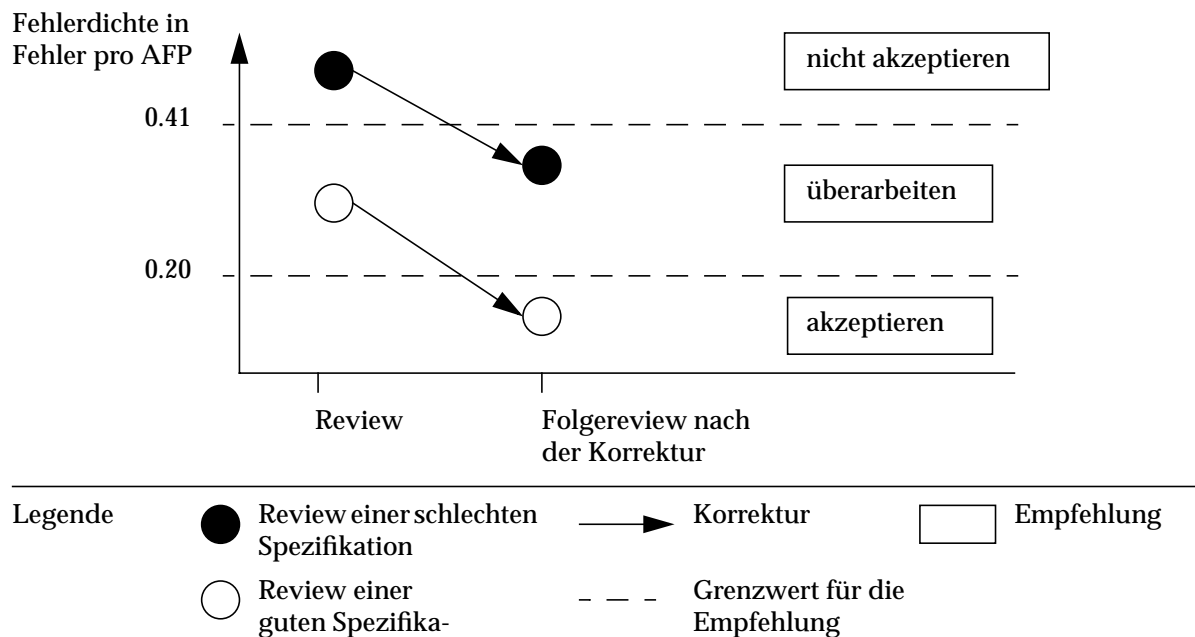
Der Autor beschwert sich, wenn während einer Sitzung eine Motivationsverringerung um 1.5 Punkte erfolgt. Typisch werden in einer Sitzung des Spezifikationsreviews etwa 15 Fehler entdeckt. Damit sich der Autor beschwert, müssen sehr ungünstige Teilnehmer gewählt werden. Die Motivation des Autors wird über eine Funktion wie im QSVA-Modell angepasst.

### Die Empfehlung an den Projektleiter

Die Bewertung der Gutachter erfolgt aufgrund der im Review entdeckten Fehlerdichte (Fehler pro AFP) im Prüfling. Entdecken die Gutachter eine hohe Fehlerdichte im Review, werden sie den Prüfling nicht akzeptieren, bei einer mittleren Fehlerdichte wird eine Überarbeitung empfohlen und bei einer niedrigen Fehlerdichte wird der Prüfling akzeptiert. Für die Empfehlung werden damit zwei Grenzwerte benötigt: ein Grenzwert für *Überarbeiten* oder *nicht akzeptieren* und ein Grenzwert für *Überarbeiten* oder *akzeptieren*. Verluste werden bei der Bewertung nicht berücksichtigt. Da Verluste und Fehler proportional zueinander modelliert sind, wird die Empfehlung nur aufgrund der im Review entdeckten Fehler gegeben.

Für die Quantifizierung wurden Testfälle mit dem QS-Modell durchgeführt, die unterschiedlichen Eigenschaften der Gutachter stammen aus einem Experiment.

Für das Spezifikationsreview wird der Einsatz des Kunden und der Autor der Spezifikation variiert. Zusätzlich wird das Folgereview nach der Korrektur betrachtet. Abbildung 34 zeigt die Grenzwerte für das Spezifikationsreview. In der Abbildung sind zwei Beispiele dargestellt. In beiden Beispielen wurde der Kunde im Review eingesetzt, in einem Fall war der Autor nicht für die Spezifikation geeignet, die Spezifikation wurde bei der ersten Prüfung nicht akzeptiert.



**Abb. 34:** Grenzwerte für das Spezifikationsreview

Wie die Abbildung zeigt, wird die Spezifikation bei einer Fehlerdichte von weniger als 0.20 Fehlern pro AFP akzeptiert. Bei mehr als 0.41 Fehlern pro AFP wird sie nicht akzeptiert, zwischen beiden Grenzwerten wird eine Überarbeitung empfohlen.

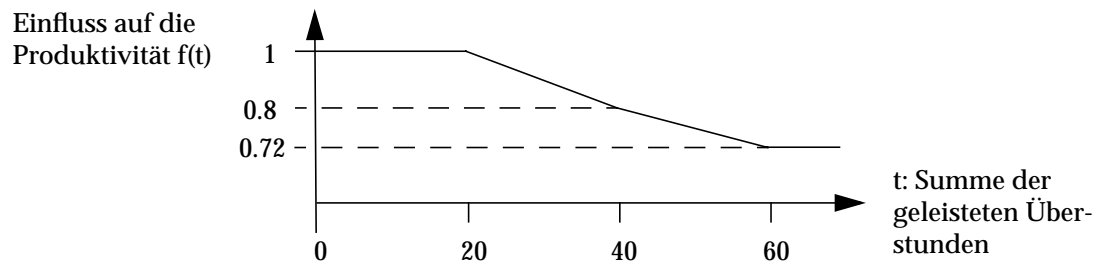
Beim Grobentwurf wird eine Überarbeitung mit erneutem Review bei mehr als 0.36 Fehlern pro AFP verlangt. Zwischen 0.36 und 0.16 Fehlern pro AFP wird eine Überarbeitung empfohlen, liegt die Fehlerdichte unter 0.16 Fehlern pro AFP, wird der Prüfling akzeptiert.

## 5.4.2 Verhalten der Entwickler

### 5.4.2.1 Überstunden der Entwickler

Die Auswirkung von zu vielen Überstunden wird für die Entwickler ähnlich quantifiziert wie in Dudler (2000). Dazu wird die Summe der geleisteten Überstunden betrachtet. Metzger und Boddie (1996) beschreiben, dass die Produktivität der Entwickler sinkt. Abbildung 35 zeigt, dass eine abschnittsweise definierte Funktion verwendet wird, die die Produktivität beeinflusst. Die Funktion hängt von der Anzahl der geleisteten Überstunden ab.

Nach 20 geleisteten Überstunden fängt der Entwickler an, sich an die längere Arbeitszeit anzupassen. Er nähert sich seiner sonst üblichen Arbeit an und wird bei 40 Überstunden gleich viel arbeiten wie ohne Überstunden. Dann sinkt seine Produktivität weiter, ab 60 Stunden arbeitet der Entwickler jeden Tag insgesamt 10% weniger.



**Abb. 35:** Einfluss der Überstunden auf die Produktivität

Wenn ein Entwickler keine Überstunden mehr leisten muss, dann vergisst er pro Tag eine Überstunde. Es wird einmal am Tag eine Stunde von der Summe der geleisteten Überstunden abgezogen. Überstunden werden mit Zuschlägen belohnt, ein Entwickler bekommt pro Überstunde einen um 25% höheren Stundenlohn.

#### 5.4.2.2 Motivationsanpassung der Entwickler

Damit die Motivation des Autors nicht nur verringert wird, verbessert sich die Motivation des Autors mit der Zeit. Dudler (2000) hat die Motivation jeden Tag mit dem Faktor 0.976 multipliziert. Die Regel wird nur an Werktagen ausgeführt. Damit vergisst der Entwickler nach 42 Arbeitstagen eine Änderung der Motivation. Pro Arbeitsstunde ergibt sich eine Motivationsänderung von  $0.976^{0.125} = 0.9970$ , die Motivation wird abhängig von der Arbeitszeit in jedem Zeitschritt verringert.

### 5.5 Hypothesen zur Modellierung

In Kapitel 3 wurden Hypothesen aufgestellt, die Probleme bei der Konzeption eines Modells vermuten. In diesem Abschnitt werden die Hypothesen anhand der feingranularen Variante geprüft.

#### Komplexität des Modells

Die Hypothese H 5 auf Seite 17 beschreibt, dass in einer feingranularen Variante die Komplexität steigt. Bereits das QS-Modell ist komplex. Es zeigt sich besonders bei der feineren Reviewmodellierung, dass für die Vorbereitung und die Sitzung viele Effekte ineinandergreifen. Das Modell wird damit noch komplexer.

#### Wechsel der Abstraktionsebene

Die Forderung F 1 auf Seite 17 verlangt, dass innerhalb eines Modells die Abstraktionsstufe nicht gewechselt werden soll. Die Probleme beim Wechsel zeigen sich etwa am Arbeitspaket. Im QS-Modell kann der Spieler keine einzelnen Arbeitspakete verwenden, die Pakete werden nur intern benutzt. In der feingranularen Variante definiert er durch die Angabe des Umfangs den Prüfling, der ein Arbeitspaket aus einem Dokument darstellt. Er hat aber keine Möglichkeit, einen bestimmten Prüfling zu benennen, diesen zu korrigieren und dann erneut zu prüfen. Die Einbettung in das QS-Modell erzwingt, dass

vor einer erneuten Prüfung das gesamte Dokument geprüft werden muss. Diese Modellierung kann den Spieler verwirren (Hypothese H 17 auf Seite 23).

### **5.5.1 Hypothese zur Quantifizierung**

Die Hypothese H 7 beschreibt, dass feingranulare Effekte schwer quantifiziert werden können. Obwohl Reviews sehr genau untersucht wurden, mussten manche Effekte abgeschätzt werden, zum Beispiel der Einfluss der Aspekte auf die Vorbereitungsdauer. Es gibt in der Literatur viele Reviewvarianten, die unterschiedlich gut untersucht wurden. Einige Daten habe ich aus anderen Varianten übernommen – beispielsweise die Zahlen für die Prüfung durch Aspekte, die aus der Prüfung mit Szenarien stammen. Die Daten mussten teilweise an das QS-Modell angepasst werden. Ein Beispiel dafür ist die Vorbereitungsdauer des Entwurfsreviews, die gegenüber dem Spezifikationsreview abgesenkt wurde, ein weiteres Beispiel ist der Umfang des Prüflings, der von 50 Seiten auf 26 Seiten gesenkt wurde. Die Hypothese H 7 wird damit bestätigt.

## Kapitel 6

### Realisierung einer feingranularen Variante

In diesem Kapitel wird die Realisierung der feingranularen Variante beschrieben. Der erste Abschnitt zeigt die Anforderungen an die Implementierung. Im Abschnitt 6.2 wird der Grobentwurf dargestellt. Der Abschnitt 6.3 erläutert die wichtigsten Strukturen der Variante. Im letzten Abschnitt werden Hypothesen zur Implementierung untersucht.

#### 6.1 Grundlegende Anforderungen

Die umgestellten Phasen des QS-Modells aus Kapitel 4, Spezifikation und Entwurf jeweils mit Reviews und Korrektur, werden um die in Kapitel 5 vorgestellten Konzepte erweitert. Die Schrittweite ist auf zwei Stunden eingestellt. Eine wichtige Anforderung an den Entwurf und die Implementierung ist eine mögliche Parametrisierung des Modells über die Schrittweite. Die feingranularen Effekte, etwa die Vorbereitung und die Reviewsitzung, sollen auch bei einer anderen Schrittweite des Modells zum Tragen kommen. Sie können dann aber möglicherweise nicht beobachtet werden.

Eine Parametrisierung soll auch für die unterschiedlichen Erweiterungen ermöglicht werden. Beispielsweise sollen Störungen abgeschaltet werden können, um Modellvarianten mit unterschiedlichem Schwierigkeitsgrad zu ermöglichen (Hypothese H 3 auf Seite 16 und Hypothese H 18 auf Seite 23).

Eine weitere Anforderung ist die Implementierung ohne Aktivitätsregeln, es sollen nur diskrete Regeln verwendet werden. Aktivitätsregeln erfüllen zwei Funktionen (siehe Abschnitt 2.2.5). Sie sind gedacht, um kontinuierliche Veränderungen über einen längeren Zeitraum zu modellieren und um das Regelmodell zu strukturieren. Ein Nachteil der Aktivitätsregeln ist, dass sie durch ihren Bedingungsteil die Voraussetzung für die eingeschachtelten Regeln schaffen. Damit wird für diese Regeln ein Zustand hergestellt, die Regeln sind nicht mehr vom gesamten Situationsmodell, sondern nur von einem Teil des Situationsmodells abhängig. Für die feingranulare Variante haben Aktivitäten zwei weitere Nachteile:

- Kontinuierliche Veränderungen werden in der Variante 2 nicht modelliert, Sitzungen etwa zählen auch bei einer kurzen Schrittweite zu den Ereignissen. Einziger Kandidat wäre die Vorbereitung, die jedoch nur wenige Stunden dauert. Damit wird die Hypothese H 9 (Seite 19) bestätigt.
- Die Strukturierung des Regelmodells ist für die Variante wünschenswert, da vermutlich viele Regeln für feingranulare Effekte entstehen. Diese Regeln könnten in Aktivitäten eingeschachtelt werden, um eine Strukturierung des Regelmodells zu erreichen. Soll das Modell aber unabhängig von der Schrittweite sein, muss es flexi-

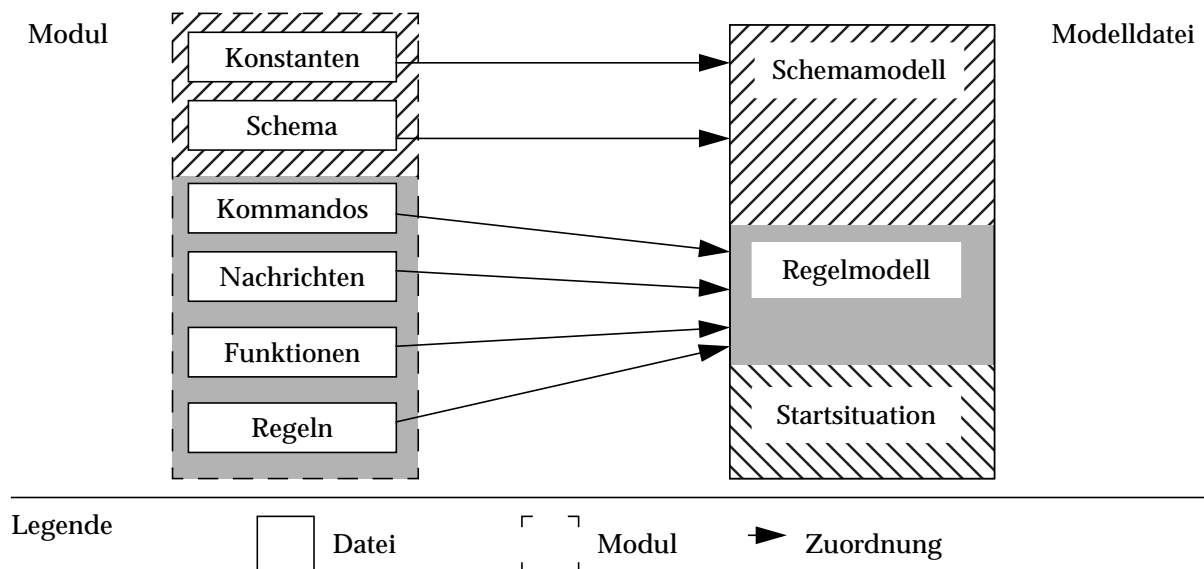
bel reagieren können: Findet etwa eine Sitzung statt, kann der teilnehmende Autor bei kurzer Schrittweite nichts anderes mehr machen. Bei langer Schrittweite beschwert er sich als Folge der Sitzung beim Projektleiter und widmet sich in der restlichen Zeit anderen Tätigkeiten. Durch die Strukturierung wird aber die Ausführungsreihenfolge der Regeln vorgegeben: Die Schachtelungstiefe und nicht das Situationsmodell würde bestimmen, wann der Bedingungsteil einer Regel geprüft wird. Wird die Sitzung tiefer eingeschachtelt als die Störung, kann die Störung nicht mehr im gleichen Zeitschritt nach der Sitzung durchgeführt werden.

Die einzige Aktivität, die in der feingranularen Variante benutzt wird, ist daher die vorgegebene Aktivität Initialize. Damit können zu Beginn des Zeitschritts Voraussetzungen für den Zeitschritt geschaffen werden, etwa dass im Zeitschritt eine Sitzung stattfindet. Die eigentlichen Effekte werden auf globaler Ebene durch diskrete Regeln realisiert.

## 6.2 Grobentwurf

### 6.2.1 Aufteilung in Module

Die einzelnen Teile der Erweiterung, etwa Reviews, Störungen oder die Arbeitszeit der Entwickler, sind durch einzelne Module implementiert. Dabei habe ich mich am SAM orientiert (siehe Eisenbarth, Rohrbach, 1998, Seite 89) und inhaltlich zusammengehörige Komponenten zu Modulen zusammengefasst. Jedes Modul besteht aus verschiedenen Dateien, die entweder dem Schemamodell oder dem Regelmodell zugeordnet werden. Abbildung 36 zeigt die Aufteilung eines Moduls und die Zuordnung in das Modell.



**Abb. 36:** Dateien, Modul und Modell

Die Entwicklungsumgebung SesamIDE erlaubt eine hierarchische Gliederung der Dateien und damit eine Modulsicht. Für die Übersetzung in die Basissprache müssen die

Dateien in einer Konfiguration in der richtigen Reihenfolge eingeordnet werden. Es können verschiedene Varianten erstellt werden, in dem das Regelmodell einzelner Module in das Modell übersetzt oder weggelassen wird. Der Teil des Moduls mit dem Schemamodell ist notwendig, wenn die Entitäten und Relationen in anderen Modulen benutzt werden.

## 6.2.2 Einbettung der Erweiterung in das Regelmodell

Die Struktur des Regelmodells ist durch die Beschränkung auf die Aktivität Initialize und auf globale Regeln vorgegeben.

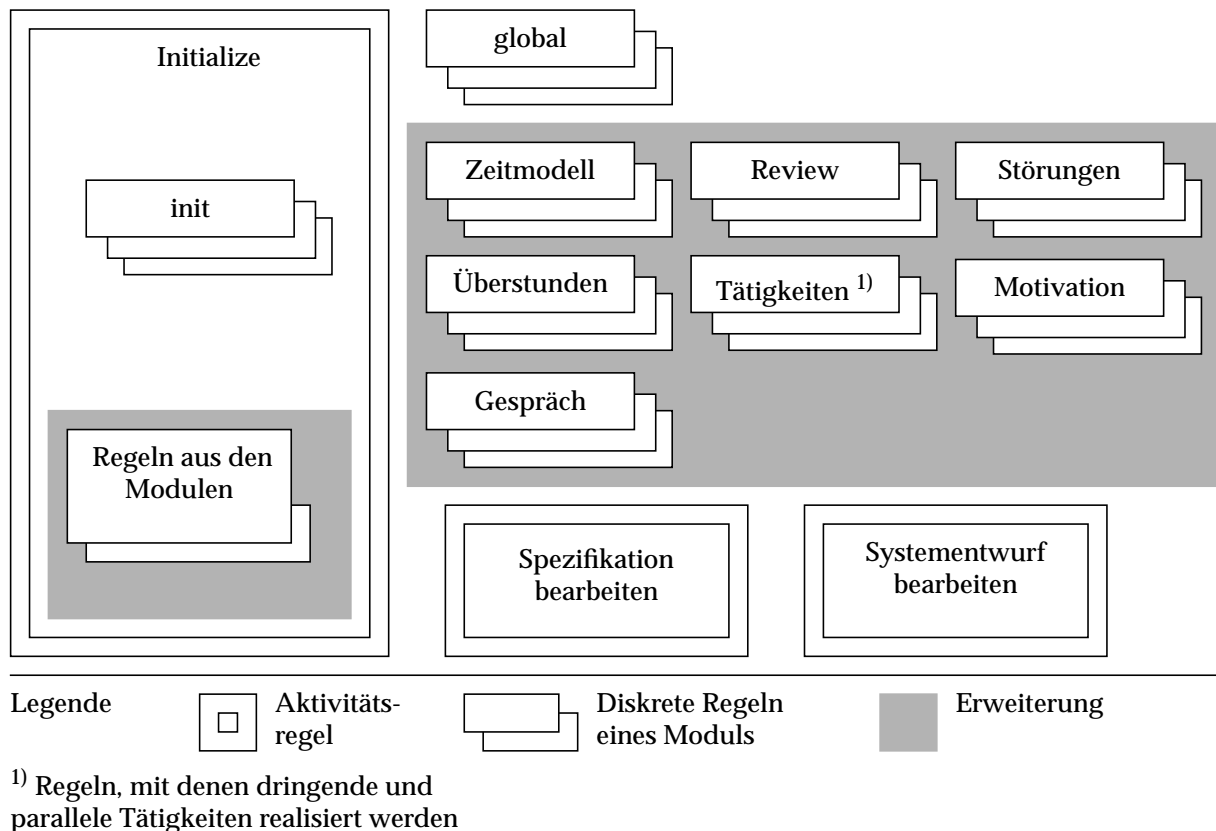


Abb. 37: Einbettung in das Regelmodell

## 6.2.3 Einbettung der Erweiterung in das Schemamodell

Die Einbettung der Erweiterung wurde für das Review bereits im Abschnitt 5.3.1.4 gezeigt. Um möglichst wenig an bestehenden Komponenten zu ändern, bilden die verwendeten Entitäten für Dokumente und Mitarbeiter die Schnittstellen. Die übrigen Erweiterungen sind durch zusätzliche Entitäten und Relationen realisiert, bestehende Komponenten des Schemamodells wurden möglichst nicht verändert. Ausnahmen sind die Mitarbeiter, die Entitäten wurden um die Motivation und die zusätzlichen Eigenschaften ergänzt.

## 6.3 Strukturen der Implementierung

In diesem Abschnitt werden die Strukturen der Modellvariante kurz beschrieben. Die wichtigsten Entitäts- und Relationsklassen werden vorgestellt und das Zusammenspiel der verschiedenen Regeln wird gezeigt.

### 6.3.1 Verhalten der Mitarbeiter

#### 6.3.1.1 Parallelarbeit im QS-Modell

Im QS-Modell ist die parallele Arbeit an Dokumenten nicht modelliert. Die feingranulare Variante erlaubt, dass ein Mitarbeiter alle zwei Stunden eine neue Tätigkeit auswählt und damit Dokumente quasi-parallel bearbeitet. Im QS-Modell existieren bereits zwei Relationsklassen für die Durchführung einer Tätigkeit. Die erste Klasse ist die Klasse "soll\_bearbeiten". Wenn der Spieler anweist, dass ein Entwickler ein Dokument bearbeiten soll, wird eine Relation aus dieser Klasse erzeugt. Führt der Entwickler die Tätigkeit daraufhin durch, wird eine Relation der zweiten Klasse erzeugt, eine "bearbeitet"-Relation. Diese Relationsklassen sind in Unterklassen für die verschiedenen Tätigkeiten gegliedert.

Beide Relationsklassen konnten nicht für die Parallelarbeit verwendet werden. Relationen der Klasse "soll\_bearbeiten" werden zu früh gelöscht, wenn der Entwickler die Tätigkeit beenden soll. Das aktuelle Arbeitspaket könnte dann nicht mehr zu Ende bearbeitet werden. Die Relation für die Tätigkeit kann nicht verwendet werden, da sie innerhalb einer Aktivität erzeugt wird. Für die globalen Regeln zur Auswahl der Tätigkeit wird die Relation zu spät erzeugt. Die Tätigkeit könnte im ersten Zeitschritt, in dem sie ausgeführt werden soll, nicht gewählt werden. Daher werden neue Relationen im QS-Modell eingeführt:

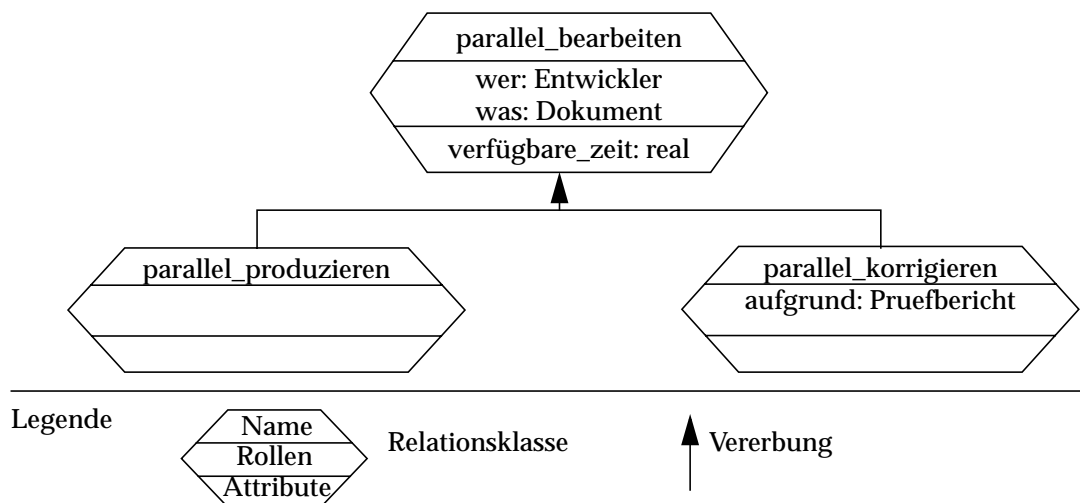


Abb. 38: Relationsklassen für die Parallelarbeit



Für jede Tätigkeit eines Entwicklers wird bei Beginn der Tätigkeit die entsprechende Relation erzeugt. Sie bleibt bis zum Ende der Tätigkeit bestehen. Hat der Entwickler Zeit für eine Tätigkeit, wird eine der Relationen ausgewählt und die verfügbare Zeit gesetzt. Die Tätigkeit wird erst gewechselt, wenn der Entwickler mindestens zwei Stunden für die Tätigkeit aufgewendet hat. Ist der Entwickler mit der Tätigkeit fertig, wird die Relation gelöscht. Die Regeln für die Durchführung der Tätigkeit binden diese Relation, um die verfügbare Zeit zu verwenden. An dieser Stelle mussten bestehende Regeln des QS-Modells verändert werden (Hypothese Q 3 auf Seite 28).

### **6.3.1.2 Benutzerkommandos und Tätigkeiten in der Erweiterung**

Für die Variante 2 habe ich mich an der Realisierung des QS-Modells orientiert. Das Zusammenspiel zwischen Benutzerkommandos und der Durchführung der Tätigkeit, etwa für die Vorbereitung auf Reviewsitzungen, wird durch zwei Relationsklassen realisiert. Eine Klasse dient der Anweisung durch den Projektleiter, etwa "Gutachter\_soll\_vorbereiten". Die andere Klasse wird für die Durchführung verwendet, zum Beispiel "Gutachter\_bereitet\_Review\_vor".

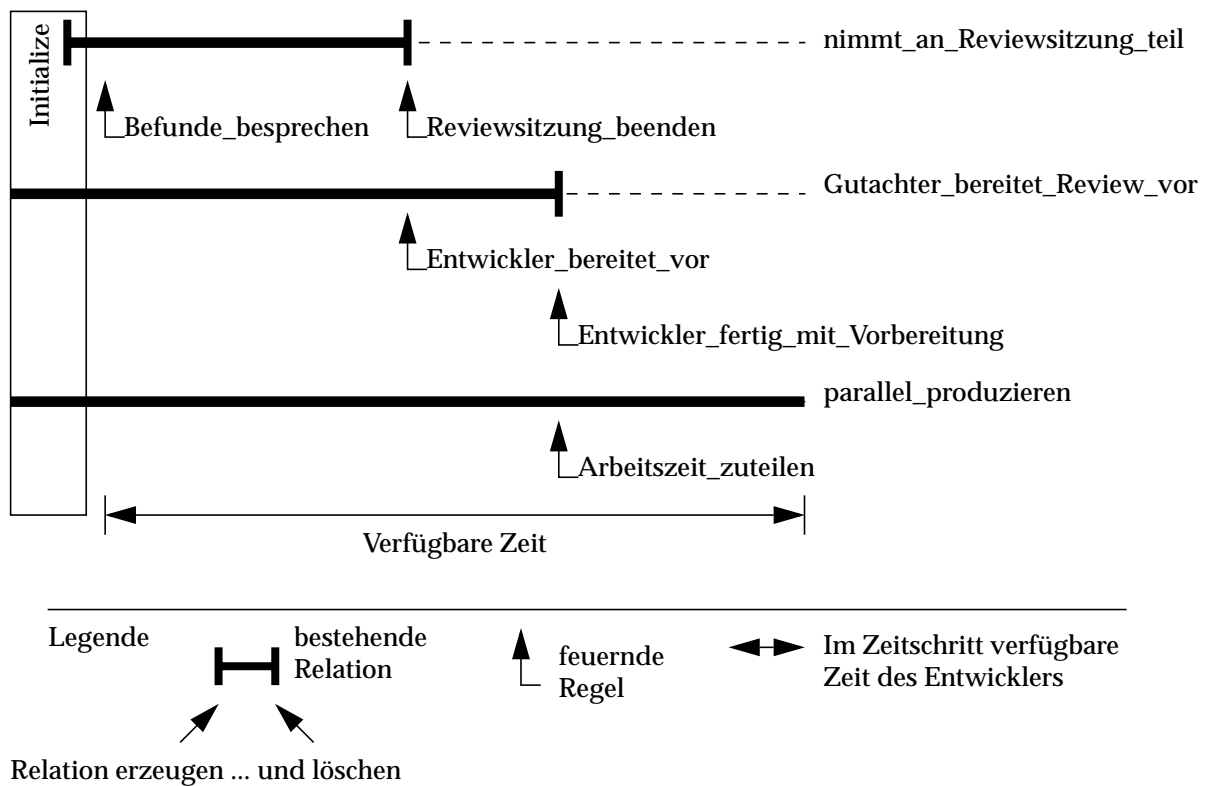
Für dringende Tätigkeiten wird genauso vorgegangen. Verlangt der Projektleiter, dass eine Tätigkeit dringend durchgeführt wird, wird eine Relation der Klasse "soll\_dringend\_bearbeiten" erzeugt. Die Tätigkeit, die ein Entwickler dringend durchführt, wird durch eine "dringend\_bearbeiten"-Relation gekennzeichnet. Bei parallelen Tätigkeiten ist dieses Vorgehen nicht notwendig, da die entsprechenden Relationen bereits im QS-Modell existieren (siehe Abschnitt 6.3.1.1).

### **6.3.1.3 Reihenfolge der Tätigkeiten**

Im Gegensatz zum ursprünglichen QS-Modell haben die Mitarbeiter sehr unterschiedliche Aufgaben zu erledigen. Sie bereiten sich für Reviews vor, nehmen an Reviewsitzungen teil, treffen sich mit dem Projektleiter oder erledigen eine Tätigkeit dringend.

Wie im Abschnitt 5.3.3 beschrieben, ziehen Mitarbeiter bestimmte Aufgaben vor. Sie versuchen, auf jeden Fall an Reviewsitzungen teilzunehmen. Die Vorbereitung ist wichtig und bleibt nur liegen, wenn eine andere Tätigkeit dringend bearbeitet werden soll. Die Variante 2 soll unabhängig von der Schrittweite sein, darum kann eine Tätigkeit mehrere Zeitschritte dauern oder nur einen Teil eines Zeitschritts verbrauchen. Das Verhalten wird über Relationen und die verfügbare Zeit eines Entwicklers gesteuert. Beginnt ein Entwickler mit einer Tätigkeit, wird eine Relation für die Tätigkeit in der Aktivität Initialize bei Beginn des Zeitschritts erzeugt, zum Beispiel die Relation "Gutachter\_bereitet\_Review\_vor" aus Abschnitt 6.3.1.2. Diese Relation zeigt an, was der Entwickler in diesem oder in den folgenden Zeitschritten macht. Es steht damit aber noch nicht fest, wieviel Zeit er für die Tätigkeit hat und wie lange er braucht. Das kann nur während des Zeitschritts festgestellt werden: Jede Regel, mit der eine Tätigkeit durchgeführt wird, kann erst feuern, wenn keine Relation einer wichtigeren Tätigkeit des Entwicklers existiert. Wieviel Zeit der Entwickler im aktuellen Zeitschritt hat, wird durch die individuelle verfügbare Zeit des Entwicklers festgestellt.

Findet zum Beispiel im Zeitschritt eine Sitzung statt, an der ein Mitarbeiter teilnimmt, wird dafür die Relation “nimmt\_an\_Reviewsitzung\_teil” angelegt. So lange diese Relation existiert, kann der Mitarbeiter nichts anderes machen. Für die Sitzung verbraucht der Mitarbeiter Arbeitszeit, die er nicht für andere Tätigkeiten verwenden kann. Am Ende der Sitzung wird die Relation gelöscht, erst dann kann der Mitarbeiter die übrige Zeit nutzen. Abbildung 39 zeigt beispielhaft das Zusammenspiel zwischen Relationen und Regeln für einen Zeitschritt von zwei Stunden. Der Mitarbeiter ist zuerst eine Stunde in einer Reviewsitzung und bereitet sich danach 30 Minuten für ein anderes Review vor. Da er im Zeitschritt mit der Vorbereitung fertig wird, kann er noch 30 Minuten an einem Dokument weiterschreiben.

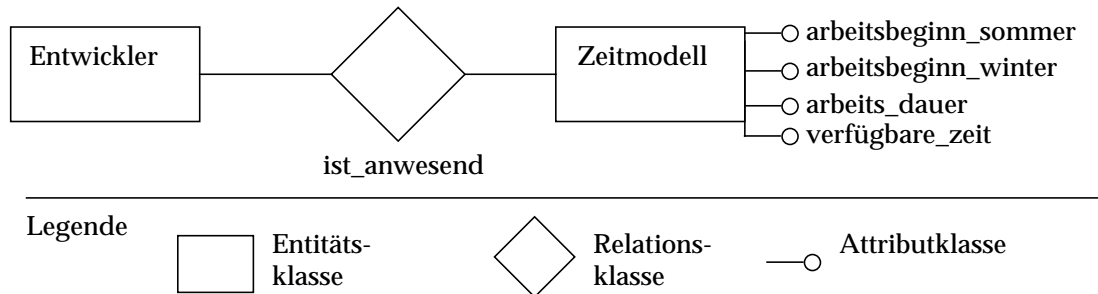


**Abb. 39:** Aufteilung zwischen verschiedenen Tätigkeiten

Dieses Vorgehen funktioniert unabhängig von der Schrittweite. Bei einer großen Schrittweite wird der Zeitschritt aufgeteilt, bei einer kleinen Schrittweite bleibt eine Relation über mehrere Zeitschritte bestehen. Der Nachteil dieser Lösung ist, dass bereits existierende Regeln verändert werden müssen: Im QS-Modell wird die verfügbare Zeit in einer Entität zentral gesetzt, die Aufteilung des Zeitschritts für Reviews und andere Tätigkeiten wird dezentral durch den Zeitanteil für Reviews modelliert. In der Variante 2 kann die verfügbare Zeit nicht mehr in einer einzigen Entität gespeichert werden. Die verfügbare Zeit muss für jeden Mitarbeiter individuell berechnet werden und jedem Mitarbeiter einzeln zugeordnet sein.

### 6.3.2 Arbeitszeit der Entwickler

Die Arbeitszeit eines Entwicklers wird in einer eigenen Entität realisiert. Die Entität ist über eine Relation dem Entwickler zugeordnet. Abbildung 40 zeigt den Ausschnitt aus dem Situationsmodell.



**Abb. 40:** Arbeitszeit eines Entwicklers

Die Abbildung zeigt die wichtigsten Attribute der Entitätsklasse Zeitmodell. Der Arbeitsbeginn muss für Sommer- und Winterzeit unterschiedlich eingestellt werden, da sich das Datum bei der Zeitemstellung von geraden auf ungerade Stunden ändert. Die Arbeitsdauer definiert, wie lange der Entwickler pro Tag arbeitet. Die verfügbare Zeit wird abhängig von der Modellzeit und der Arbeitsdauer gesetzt, bei einer Schrittweite von einem Tag auf acht Stunden, bei einer Schrittweite von zwei Stunden außerhalb der Arbeitszeit auf Null, innerhalb der Arbeitszeit auf zwei Stunden.

Überstunden werden realisiert, indem die Arbeitsdauer um eine oder um zwei Stunden hochgesetzt wird. Der Mitarbeiter arbeitet länger (Hypothese H 4.1, Seite 17).

### 6.3.3 Strukturen im Review

#### 6.3.3.1 Entitäten des Reviews

Für das Review werden neben den beteiligten Entwicklern, der Vorgabe und dem zu prüfenden Dokument drei neue Entitätsklassen verwendet:

- Sitzungen werden durch die Entitätsklasse "Review\_Sitzung" realisiert. Wichtige Attributtypen sind der Termin der Sitzung, der Prüfling und die zu besprechenden Befunde.
- Die Vorbereitung wird durch die Entitätsklasse "Befunde\_eines\_Gutachters" realisiert. Eine erzeugte Entität enthält die vom Gutachter vorbereiteten Befunde. Befunde werden als Menge realisiert, jeder Befund ist einer Anforderung und einem Aspekt über eindeutige Nummern zugeordnet (siehe auch Abbildung 19, Seite 51). Er enthält die entdeckten Verluste und Fehler und vermeintliche Fehler nach Fehlerarten.

- Die “Reviewzusammenfassung” enthält die Empfehlung an den Projektleiter und den Grund des Abbruchs, die Gesamtzahl der entdeckten Fehler und Verluste und wieviel Seiten in der Sitzung geprüft wurden.

### 6.3.3.2 Vorbereitung eines Entwicklers

Für die Vorbereitung wird eine Entität der Klasse “Befunde\_eines\_Gutachters” und eine Relation erzeugt, die den Entwickler, seine Befunde und die Sitzung verbindet. Abbildung 41 zeigt die Struktur im Situationsmodell.

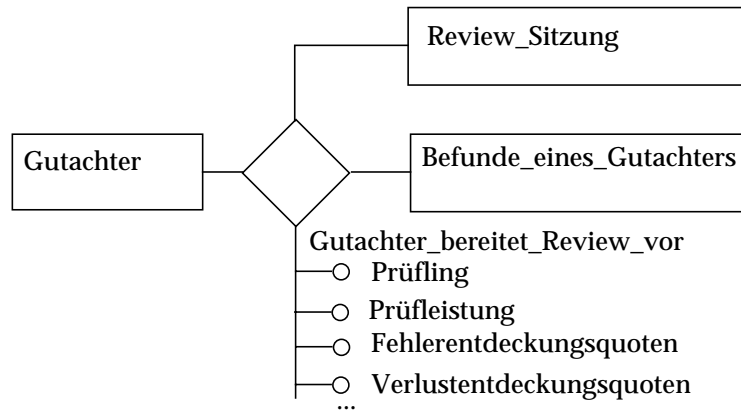


Abb. 41: Struktur für die Vorbereitung

Die Relation enthält lokale Daten für die Vorbereitung, beispielsweise die zu prüfenden Anforderungen oder die Prüfleistung. Für die unterschiedlichen Dokumente wird die Relation spezialisiert. Für das Spezifikationsreview wird die Relation “Gutachter\_bereitet\_Spezreview\_vor” angelegt, eine entsprechende Relation wird für das Entwurfsreview erzeugt.

### 6.3.3.3 Die Reviewsitzung

Findet eine Reviewsitzung statt, wird die Reviewzusammenfassung für die Empfehlung erzeugt. Eine Relation verbindet Moderator, Autor, den Prüfbericht für die Befunde und die Reviewzusammenfassung. Die Relation wird für die unterschiedlichen Dokumente spezialisiert, die Abbildung zeigt die abstrakte, übergeordnete Relation:

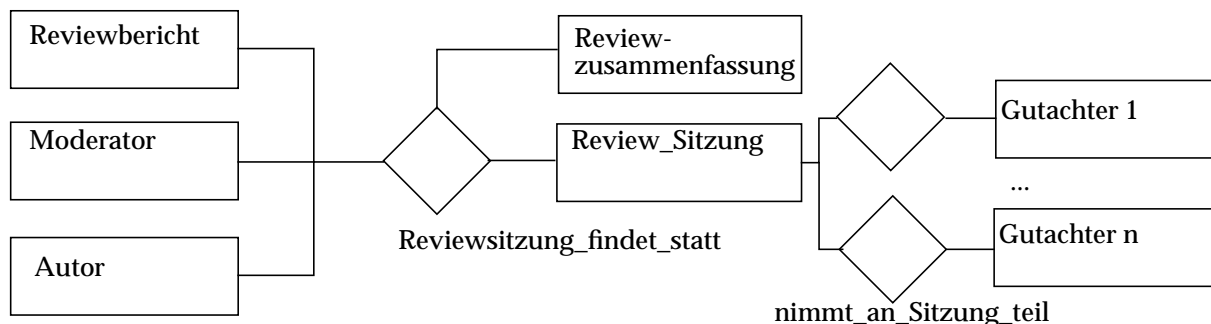


Abb. 42: Struktur der Reviewsitzung

Die teilnehmenden Gutachter werden der Reviewsitzung mit der Relation "nimmt\_an\_Sitzung\_teil" zugeordnet. Die Relation wird nur erzeugt, wenn der Gutachter tatsächlich in der Sitzung erscheint. Da die Hochsprache nicht erlaubt, beliebig viele Entitäten mit einer Regel zu binden, werden alle Befunde der teilnehmenden Gutachter zu Beginn der Sitzung in einem Attribut der Relation "Sitzung\_findetstatt" zusammengelegt. Die Einflüsse der Gutachter in der Reviewsitzung werden ähnlich behandelt und in einem gemeinsamen Attribut in der Relation gespeichert.

Am Ende der Sitzung werden diese Relationen gelöscht. Die Sitzung und die Empfehlung bleiben erhalten und sind über eine Relation verbunden, damit der Spieler die Ergebnisse einer bestimmten Sitzung erfragen kann.

## **6.4 Hypothesen zur Implementierung**

In diesem Abschnitt werden die aufgestellten Hypothesen zur Implementierung betrachtet.

### **6.4.1 Arbeitszeit der Entwickler und des Projektleiters**

In der Forderung F 2 auf Seite 19 wird vom Modellbauer verlangt, dass er Regeln für die Arbeitszeit implementiert. Die Regeln für die Arbeitszeit konnten aus der Variante 1 mit einer Schrittweite von zwei Stunden übernommen und an die individuelle Arbeitszeit der Entwickler angepasst werden.

Die Forderung an die Hochsprache F 3 (Seite 26), die ein flexibles Zeitmodell der Hochsprache fordert, kann nicht umgesetzt werden. Für ein Zeitmodell mit individueller Arbeitszeit der Entwickler, das von der Hochsprache bereitgestellt wird, müsste entweder der Hochsprachenübersetzer oder der Simulator die Bedeutung der Entitäten kennen, damit wird die Modellierung mit SESAM eingeschränkt. Das Zeitmodell muss deshalb wie in der feingranularen Variante vom Modellbauer erstellt werden.

### **6.4.2 Größe und Übersichtlichkeit**

Das Zusammenspiel der Regeln auf globaler Ebene ist, wie Abbildung 39 zeigt, komplex. Die Erweiterungen sind mit insgesamt 142 Regeln implementiert, davon gehören 89 zum Spezifikations- und Entwurfsreview. Das Situationsmodell wird größer, die Struktur der Sitzung oder der Vorbereitung zeigt, dass viele Entitäten gebunden werden müssen. Eine Reihe von Entitäten und Relationen werden erzeugt und gelöscht. Die Hypothesen H 6 (Seite 17) und H 5 (Seite 17), die eine Steigerung der Komplexität des Modells durch die Erweiterung um feingranulare Effekte vermuten, werden dadurch bestätigt.

# Kapitel 7

## Überprüfung der Ergebnisse

In diesem Kapitel wird der Test der feingranularen Variante beschrieben. Im ersten Abschnitt wird kurz das Vorgehen beim Test beschrieben, Abschnitt 7.2 zeigt die Abweichungen zwischen dem feingranularen Review und dem Review im QS-Modell. In Abschnitt 7.3 werden die Hypothesen betrachtet.

### 7.1 Vorgehen

Wie beim Test der Modellvariante in Kapitel 4 sind die Testfälle in einem Testprotokoll beschrieben.

Das Testprotokoll ist in verschiedene Kapitel gegliedert. Das erste Kapitel beschreibt Testfälle für den Vergleich der feingranularen Reviews aus der Variante 2 gegen die Reviews des QS-Modell. Im Test wird dazu die Variante 1 verwendet, damit Unterschiede, die nur durch die Änderung der Zeitgranularität entstehen, nicht zum Tragen kommen. Im zweiten Kapitel sind die Tests für ausgewählte Effekte der Variante 2 enthalten. Es werden Störungen, parallele und dringende Tätigkeiten, Reviews, Treffen mit dem Projektleiter und Überstunden getestet. Anschließend sind im Testprotokoll die Testfälle für die Prüfung der Hypothesen beschrieben, der letzte Teil untersucht die Parametrisierung durch die Schrittweite. In diesem Kapitel wird aus Platzgründen nur der Test der feingranularen Reviews gegen die Reviews im QS-Modell und der Test der Hypothesen gezeigt.

Die Quantifizierung des feingranularen Reviews wurde an die Quantifizierung der Reviews im QS-Modell angepasst. Dabei wurden die Entdeckungsquoten für Fehler um wenige Prozent nach unten korrigiert. Damit die Sitzung des Spezifikationsreviews mit dem Kunden bei der Prüfung von 26 Seiten nicht länger als zwei Stunden dauert, werden die Befunde etwas schneller besprochen. Damit der Aufwand insgesamt gleich bleibt (siehe 5.4.1.1), bereiten sich die Gutachter etwas langsamer vor.

Nach dem Test der feingranularen Variante wurden die Ergebnisse in einem Testbericht zusammengefasst.

### 7.2 Projektresultate im Vergleich mit dem QS-Modell

#### Testbeschreibung

Die Reviewergebnisse der feingranularen Variante werden über drei Testfälle mit den Reviewergebnissen aus dem QS-Modell verglichen. Die Testfälle stammen aus dem Test der Modellvariante in Abschnitt 4.3.1.1. Im Fall 1 prüfen zwei Gutachter die Spezifika-

tion und den Entwurf, im Fall 2 nimmt der Kunde am Spezifikationsreview teil. Drei Gutachter werden im Fall 3 für das Spezifikations- und das Entwurfsreview eingesetzt.

### Ergebnisse

Bei Reviews mit zwei Gutachtern unterscheidet sich die Anzahl der entdeckten Fehler und Verluste kaum. Auch beim Spezifikationsreview mit dem Kunden sind die Unterschiede gering (siehe Abbildung 43).

		Variante 1	Variante 2
Spezifikationsreview	Anzahl Fehler	51.7	51.5
	Verluste in AFP	4.6	4.8
Entwurfsreview	Anzahl Fehler	49.9	50.1
	Verluste in AFP	4.3	4.4

**Abb. 43:** Fehler und Verluste bei zwei Gutachtern

Werden drei Gutachter eingesetzt, tritt ein deutlicher Unterschied auf. Es werden weniger Fehler und Verluste als mit den Reviews im QS-Modell entdeckt, im Vergleich zu den Reviews mit zwei Gutachtern aber immer noch mehr Fehler:

		Variante 1	Variante 2
Spezifikationsreview	Anzahl Fehler	56.1	54.0
	Verluste in AFP	5.0	4.7
Entwurfsreview	Anzahl Fehler	52.5	51.0
	Verluste in AFP	4.6	4.4

**Abb. 44:** Fehler und Verluste bei drei Gutachtern

Der Unterschied entsteht durch die veränderte Modellierung. Im QS-Modell hat die Anzahl der Gutachter einen direkten Einfluss auf die Entdeckungsquoten, das Modell wurde nach Weller (1993) quantifiziert. Ich habe mich bei der Konzeption auf Porter et al. (1997) gestützt, der dritte Gutachter entdeckt in der Vorbereitung keine neuen Befunde, erst in der Sitzung werden durch den dritten Gutachter mehr vermeintliche Fehler aussortiert. Meiner Meinung nach sollte der Spieler in der feingranularen Variante einen etwas deutlicheren Vorteil durch den Einsatz des dritten Gutachters haben, da sonst der Einsatz des dritten Gutachters kaum sinnvoll ist. Die Variante 2 sollte deshalb Befunde des dritten Gutachters berücksichtigen. Damit werden zusätzlich Verluste durch den dritten Gutachter entdeckt. Die Abweichung der Fehler in den Dokumenten nach der Korrektur beträgt rund 3%, der Umfang unterscheidet sich um rund 0,2%.

Aufwand und Kosten in den beiden Varianten unterscheiden sich stark. Bei zwei Gutachtern entsteht mehr Aufwand durch den Moderator und den Autor, da deren Teilnahme an der Sitzung berechnet wird. Zusätzlich wird für die Nacharbeit durch den Moderator pro Sitzung eine halbe Stunde Aufwand berechnet.

Spezifikationsreviews	Variante 1	Variante 2
Aufwand bei zwei Gutachtern in MH	25.5	39.8
Aufwand bei drei Gutachtern in MH	44.3	48.4

**Abb. 45:** Aufwand im Spezifikationsreview

Obwohl die Werte der Quantifizierung für die Leistung in der Vorbereitung und im Review plausibel erscheinen (Abschnitt 5.4.1.3, Seite 62), sollten sie meiner Meinung nach stärker an das QS-Modell angepasst werden. Das Verhältnis des Aufwands für die Erstellung zum Reviewaufwand ist zu hoch, in der feingranularen Variante beispielsweise für das Spezifikationsreview 40% gegenüber rund 26% in der Variante 1. Frühauf et al. (2000) gibt 20% an. Der Phasenaufwand der Variante 2 nähert sich damit aber wieder an das QS-Modell mit der Schrittweite von einem Tag an: Für die Spezifikationsphase ergibt sich ein Aufwand von etwa 148 MH in der Variante 2, im QS-Modell ergeben sich rund 153 MH (siehe Abbildung 8 auf Seite 34).

Werden drei Gutachtern eingesetzt, spielen mehrere Effekte eine Rolle. In der Variante 2 hat jeder Gutachter weniger Aufwand, da er nicht mehr alle Aspekte betrachten muss. Zusätzlich zeigt sich ein Effekt aus dem QS-Modell: Dort bremst jeder zusätzliche Gutachter die anderen Entwickler. Das führt zu einem überproportionalen Anstieg des Aufwands, der in der feingranularen Variante nicht berücksichtigt wird.

In der Variante 2 entstehen höhere Kosten, weil ein zusätzlicher Mitarbeiter, der Moderator, in den frühen Phasen eingestellt werden muss. Im Testfall dauern Spezifikation und Entwurf mit Reviews und Korrektur insgesamt 69 Tage. Der Moderator kostet 720 DM pro Tag. Zu den Kosten von 154 680 DM kommen noch 49 680 DM dazu. Damit entstehen um ein Drittel höhere Kosten. Die Kosten des QS-Modells wurden validiert (Drappa, 2000, Seite 136) und gelten als realistisch, sie sind in der Variante 2 deutlich zu hoch. Eine Lösung ist die Verwendung von externen Mitarbeitern. Diese Möglichkeit wird im Abschnitt 8.3.4 genauer betrachtet.



## 7.3 Überprüfung der Hypothesen

In diesem Abschnitt werden Hypothesen aus Kapitel 3 untersucht, die sich auf eine feinere Zeitgranularität und eine feine Modellgranularität beziehen.

### 7.3.1 Auswirkungen auf die Modellierung

#### 7.3.1.1 Feingranulare Effekte

##### Testbeschreibung

Die Hypothese H 3 (Seite 16) beschreibt, dass bei kurzer Simulationsschrittweite feingranulare Effekte beobachtet werden können. Sie wird mit zwei Testfällen verdeutlicht. Im ersten Testfall kontrolliert der Spieler die Vorbereitungszeit eines Entwicklers. Im zweiten Fall finden zwei Reviewsitzungen am gleichen Tag statt. Der Autor wird in den Sitzungen angegriffen. Dieser Effekt kann bei kurzer Schrittweite besser analysiert werden.

##### Ergebnisse

Der Spieler kann erst bei einer Schrittweite von zwei Stunden beobachten, wie lange sich ein Entwickler vorbereitet. Der Auszug ist gekürzt:

```
1999/10/14/10:00
INVITE AXEL TO REVIEW THE SPECIFICATION WITH ABSTRACTNESS, ...

1999/10/14/12:00
ASK AXEL ABOUT THE TIME SPENT FOR PREPARING REVIEWMEETING_1
Axel: I've been preparing roughly 2.0 hours. ...

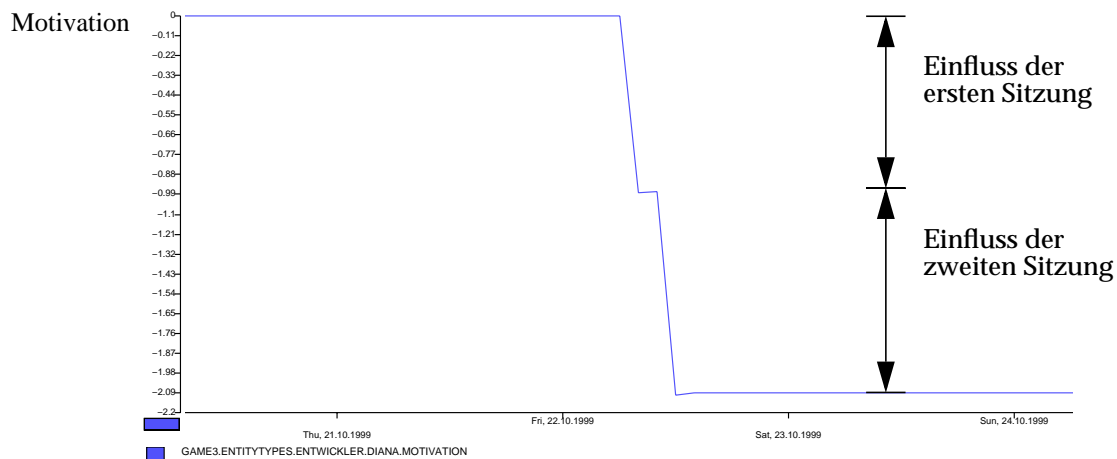
1999/10/14/14:00
Axel hopes, that he has found all errors in the Specification. He
finished the preparation for the ReviewMeeting_1

1999/10/15/08:00
ASK AXEL ABOUT THE TIME SPENT FOR REVIEWMEETING_1
I needed 4.25 hours to prepare, reports Axel. I was looking at
Abstractness, ... and Correctness.
```

**Abb. 46:** Kontrolle der Vorbereitungszeit

Die gesamte Vorbereitung kann an einem Tag durchgeführt werden. Bei einer Schrittweite von einem Tag kann der Spieler an der Nachricht für das Ende der Vorbereitung nicht erkennen, wie lange sich der Gutachter vorbereitet hat. Der Spieler kann den Fortschritt der Vorbereitung nicht beobachten, da er den Gutachter nur einmal am Tag kontrollieren kann.

Finden am gleichen Tag zwei Reviewsitzungen mit demselben Autor statt, kann nur bei einer Schrittweite von zwei Stunden festgestellt werden, in welcher Sitzung der Autor wie stark demotiviert wurde. Abbildung 47 zeigt die Motivation.



**Abb. 47:** Motivation nach zwei Sitzungen

Der Test zeigt eine weitere Hypothese (H 20, Seite 24). Bei der Analyse kann der Einfluss der beiden Sitzungen auf die Motivation getrennt betrachtet werden. Bei einer Schrittweite von einem Tag kann nur der gesamte Einfluss an diesem Tag betrachtet werden.

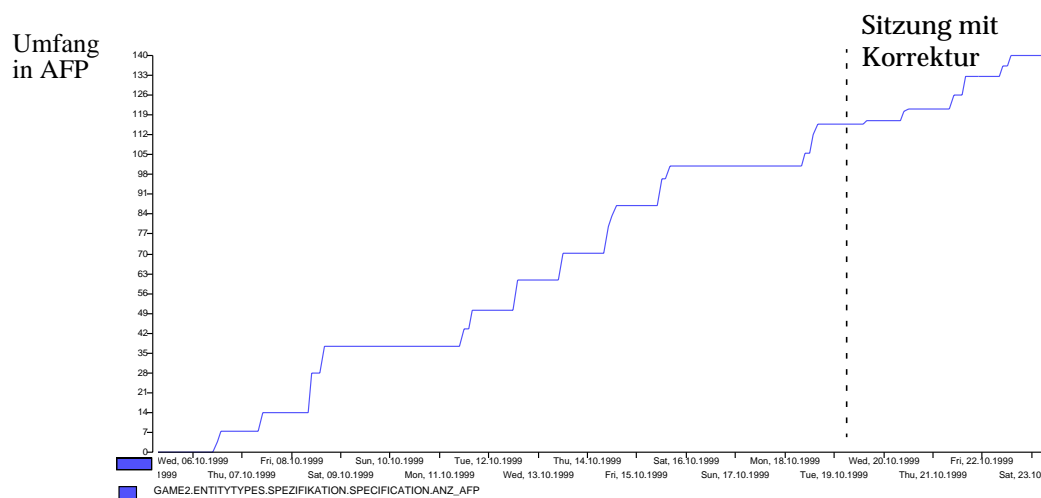
### 7.3.1.2 Verhalten der Mitarbeiter

#### Testbeschreibung

Das realistischere Verhalten eines Mitarbeiters (Hypothese H 4, Seite 16) wird im Testfall am Autor der Spezifikation gezeigt. Noch während der Mitarbeiter die Spezifikation erstellt, muss er an einer Reviewsitzung teilnehmen und die Spezifikation korrigieren.

#### Ergebnisse

Wenn der Entwickler nicht nur mit dem Schreiben der Spezifikation beschäftigt ist, kommt er langsamer voran. Die Abbildung zeigt, dass der Umfang der Spezifikation während der Sitzung und der Korrektur langsamer wächst.



**Abb. 48:** Sitzung und Korrektur während der Spezifikation

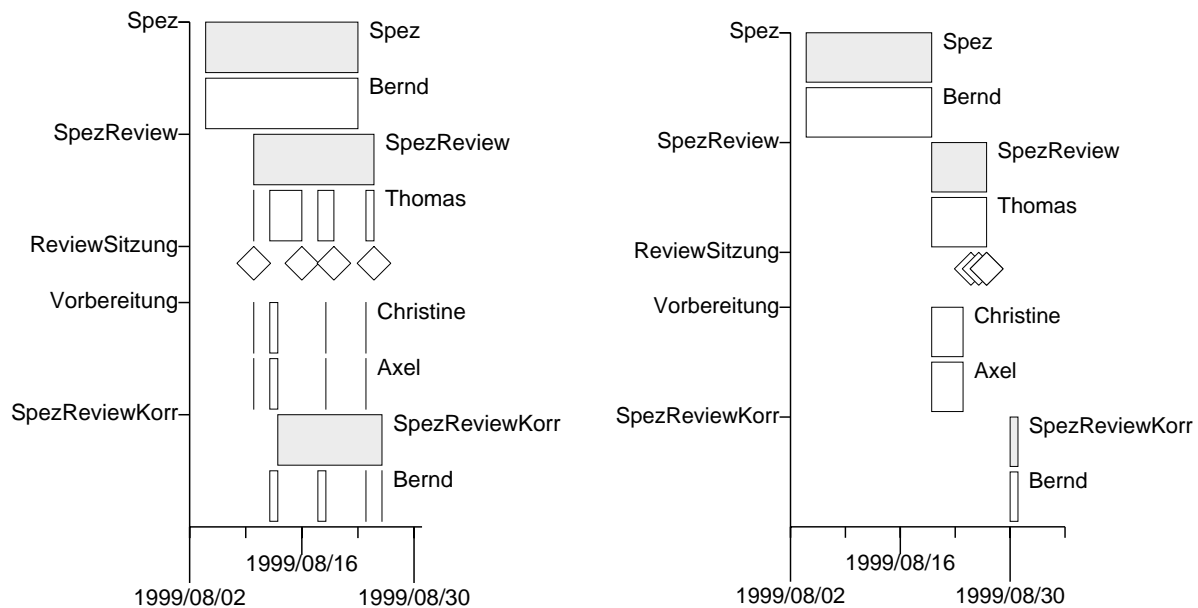
### 7.3.1.3 Variabilität

#### Testbeschreibung

Die größere Variabilität (Hypothese H 8, Seite 18) wird durch zwei unterschiedliche Spielverläufe gezeigt. Die Spezifikation wird im ersten Fall teilweise noch vor der Fertigstellung geprüft. Im zweiten Fall findet die Prüfung erst statt, wenn der Entwickler fertig spezifiziert hat. Die Sitzungen finden aber in einem sehr kurzen Abstand statt.

#### Ergebnisse

Die beiden Gantt-Diagramme zeigen den unterschiedlichen Projektverlauf. Als Moderator wurde in den Sitzungen der Entwickler Thomas eingesetzt, Christine und Axel begutachteten die Spezifikation. Nur wenn der Spieler die Sitzungen einzeln bestimmt, kann er den Abstand variieren.



**Abb. 49:** Unterschiedliche Vorgehensweisen

Weitere Beispiele für die Variabilität finden sich in den Abschnitten 7.3.2.2 und 7.3.2.5.

## 7.3.2 Auswirkungen auf die Ausbildung

### 7.3.2.1 Planung des Projekts

Als Hypothese H 10 wurde auf Seite 20 formuliert, dass der Spieler einer feingranularen Variante detaillierter planen muss. Dieser Effekt wird durch die Reviews deutlich: Der Spieler muss den Termin einzelner Sitzungen, den Umfang, den Moderator und den Autor festlegen. Er muss vorher planen, welche Mitarbeiter für welchen Zweck eingestellt werden, wann und in welchem Abstand Sitzungen stattfinden sollen und wer an den Sitzungen teilnehmen soll.

### 7.3.2.2 Eingriffe des Projektleiters

#### Testbeschreibung

Die Hypothese H 13 auf Seite 21 beschreibt, dass der Spieler feiner eingreifen kann. Als Beispiel wird die Vorbereitung gezeigt, ein Gutachter soll aber bereits dringend das System entwerfen. Der Spieler befreit den Gutachter erst kurz vor der Sitzung von der dringenden Tätigkeit.

#### Ergebnisse

Der gekürzte Ausschnitt in Abbildung 50 zeigt, dass der Spieler noch am Tag der Sitzung für die nötige Vorbereitungszeit sorgen kann. Bei einer Schrittweite von einem Tag ist das nicht mehr möglich.

1999/10/12/14:00

INVITE CHRISTINE TO REVIEW THE SPECIFICATION ...

INVITE STEFANIE TO REVIEW THE SPECIFICATION ...

1999/10/13/08:00:

Christine finished to examine the Specification for ReviewMeeting\_1.

1999/10/14/08:00

ASK STEFANIE ABOUT THE TIME SPENT FOR PREPARING REVIEWMEETING\_1

Stefanie: I've been preparing roughly 0.0 hours.

TELL STEFANIE THAT DESIGNING THE SYSTEM IS NOT URGENT.

1999/10/14/10:00

Stefanie finished to examine the Specification for ReviewMeeting\_1.

1999/10/14/14:00:

Thomas steps into your office to give you the results from ReviewMeeting\_1. The Specification contained 15 errors and 1.41293 missing AFP in 27 pages. The meeting took roughly 1.25 hours.

A correction should take place.

**Abb. 50:** Eingriffe bei feiner Zeitgranularität

Der Testfall ist auch ein Beispiel für die Hypothese H 19 (Seite 23). Der Zusammenhang zwischen der Kontrolle und der folgenden Anweisung kann erkannt werden. Im Beispiel kontrolliert der Spieler am 14.10 um 08:00 die Vorbereitungszeit der Entwicklerin Stefanie. Er stellt fest, dass sie sich noch nicht vorbereitet hat und befreit Stefanie direkt danach von der dringenden Entwurfstätigkeit.

### 7.3.2.3 Tagesablauf des Projektleiters

#### Testbeschreibung

Der Tagesablauf wird durch die Dauer der Tätigkeiten des Projektleiters und durch Störungen geprägt. Bei feiner Zeitgranularität kann der Spieler die Dauer wahrnehmen, er erfährt den Tagesablauf (siehe Hypothese H 14, Seite 22). Im Test hat der Projektleiter ein

Treffen mit einem Entwickler. Gleichzeitig findet eine Reviewsitzung statt, in der die Spezifikation geprüft wird. Der Spieler lässt die Spezifikation nach dem Review korrigieren und schaut sich die Liste mit den Befunden, den Prüfbericht, an.

## Ergebnisse

Der Auszug zeigt den Tag des Projektleiters:

**1999/10/14/08:00**

Christine comes into your office, she reminds you: We wanted to talk about the project today. So you discuss the documents Christine is writing on.

Christine is currently working on the Specification.

The Specification has a volume of 45 pages and between 42 percent to 53 percent of the functionality included.

Thomas steps into your office to give you the results from ReviewMeeting\_1. The Specification contained 10 errors and 0.0 missing AFP in 21 pages. The meeting took roughly 1.75 hours. A correction should take place.

**1999/10/14/10:00:**

**1999/10/14/12:00**

ASK DIANA TO CORRECT THE SPECIFICATION  
INSPECT THE REVIEW REPORT SPECIFICATIONREVIEWREPORT

The review report Specificationreviewreport shows, that 10 errors and 0.0 missing function points have been found.

Nobody likes correcting a document, same goes with Diana. But correcting the specification is an important work to do, everything is based on it.

**1999/10/14/14:00**

PROCEED 1 STEP

### Abb. 51: Ein Tag des Projektleiters

Es werden mehrere Effekte deutlich. In der ersten Hälfte des Tages kann der Projektleiter nicht eingreifen, er ist mit Störungen beschäftigt. Die Störungen treten bereits um 8 Uhr auf, verbrauchen aber so viel Zeit, dass der Simulator bis 12 Uhr weiterschaltet. Um 12 Uhr kann der Spieler endlich die Korrektur anweisen. Da er noch den Prüfbericht inspiziert, schaltet der Simulator in diesem Zeitschritt automatisch weiter. Erst für den letzten Zeitschritt schaltet der Projektleiter mit "proceed" die Modellzeit weiter.

#### 7.3.2.4 Fortschalten der Modellzeit

Die Hypothese H 15.1 (Seite 22) trifft vor allem bei einem grobgranularen Modell und feiner Zeitgranularität zu. In der Variante 1 etwa müssen über 50 Zeitschritte überbrückt werden, wenn die Spezifikation erstellt wird, im QS-Modell auf Tagesbasis dagegen nur 13 Zeitschritte. In der feingranularen Variante passiert mehr, vor allem während der

Reviews muss der Spieler viel mehr Aufgaben wahrnehmen. Er bekommt öfter Nachrichten und muss etwa aufgrund der Empfehlung sein weiteres Vorgehen planen. Er schaltet vermutlich die Modellzeit weniger oft weiter, und wenn, dann nur in kleinen Schritten, um eingreifen zu können (Hypothese H 15.2).

### **7.3.2.5 Detailliertes Feedback**

#### **Testbeschreibung**

In der Hypothese H 22 auf Seite 24 wird vermutet, dass der Tutor einer Schulung mit einer feingranularen Variante einen höheren Auswertungsaufwand hat. Die Hypothese H 21 beschreibt, dass der Spieler einer feingranularen Variante die erzielten Resultate und Folgen seiner Aktionen schlechter als mit einer grobgranularen Variante auf sein Vorgehen zurückführen kann.

Als Beispiel wird angenommen, dass der Spieler verschiedene Fehler bei der Durchführung des Spezifikationsreviews macht. Der Spieler hat in allen Reviews einen etwas zu großen Umfang geprüft. Im einem Review wurde nur ein Teil der Aspekte geprüft, einer der Autoren wurde als Gutachter verwendet. Für eine Reviewsitzung war die Vorbereitungszeit sehr knapp. Ein Gutachter konnte sich nicht vorbereiten, weil er in einer anderen Sitzung war. Das Beispiel wurde als Testfall durchgeführt und anschließend ausgewertet.

#### **Ergebnisse**

Um das Vorgehen des Spielers bei den Reviews zu analysieren, wird in der grobgranularen Variante ausgewertet, ob der Spieler Reviews eingesetzt hat und welche Gutachter eingesetzt wurden. Vor allem der Autor soll nicht als Gutachter verwendet werden. Für die feingranulare Variante muss der Tutor zusätzlich betrachten, ob der Umfang in Ordnung ist, ob alle Aspekte von zwei Gutachtern geprüft wurden und ob die Vorbereitungszeit ausgereicht hat.

Obwohl der Spieler Reviews eingesetzt hat, werden in den Reviews nur wenige Fehler gefunden. Dass ein Entwickler Zeit braucht, um sich vorzubereiten, kann ein Spieler vielleicht noch am Simulator erkennen. Für andere Effekte wird er vermutlich die Rückmeldungen des Tutors benötigen.

### **7.3.2.6 Schulungen mit einer feingranularen Variante**

Zwei Hypothesen beschäftigen sich mit Schulungen durch eine feingranulare Variante. Die Hypothese H 16 (Seite 22) vermutet, dass eine feingranulare Variante von einem Spieler nicht mehr an einem halben Tag durchgeführt werden kann. Diese Hypothese trifft vermutlich nur zu, wenn das gesamte QS-Modell um die feingranularen Reviews und die zusätzlichen Effekte erweitert wird. In der Hypothese H 18 wird beschrieben, dass ein feingranulares Modell für fortgeschrittene Spieler interessanter ist. Diese Hypothese kann nur schlecht mit Testfällen überprüft werden, vermutlich wird der Spieler aber stärker gefordert (siehe Abschnitt 8.1.2). Beide Hypothesen könnten erst durch Spiele am Simulator, etwa in einer Fallstudie, überprüft werden.

### 7.3.3 Technische Aspekte

#### 7.3.3.1 Performanz und Speicherbedarf des Simulators

##### Testbeschreibung

Performanz und Speicherbedarf werden mit den Testfällen aus Abschnitt 7.1 untersucht, es werden zwei oder drei Gutachter oder der Kunde als Gutachter eingesetzt. Wie bei den Tests der Modellvariante gegen das QS-Modell in Abschnitt 4.3.3 wurden die Simulationsläufe nachts ohne interaktive Eingriffe durchgeführt.

##### Ergebnisse

Die Abbildung 52 zeigt, dass die Ausführungszeit steigt, Abbildung 53 vergleicht den Speicherbedarf (Hypothesen H 25 und H 26 auf Seite 25). Durch die Erweiterung des Modells steigen im Vergleich zur Variante 1 die Ausführungszeit und der Speicherbedarf.

Testfall	Dauer (s)		Faktor
	Variante 1	Variante 2	
1	457.8	642.4	1.40
2	448.9	640.3	1.42
3	471.5	675.1	1.43

**Abb. 52:** Ausführungszeit

Testfall	Memory (MB)		Faktor
	Variante 1	Variante 2	
1	69	100	1.45
2	69	100	1.45
3	71	106	1.49

**Abb. 53:** Speicherbedarf

Der Vergleich zwischen Fall 1 mit zwei Gutachtern und Fall 3 mit drei Gutachtern zeigt, dass ein kleiner Teil der Dauer und des Speicherbedarfs durch den zusätzlichen Gutachter entsteht. Er bereitet zusätzliche Befunde vor, es werden etwas mehr Regeln ausgeführt.

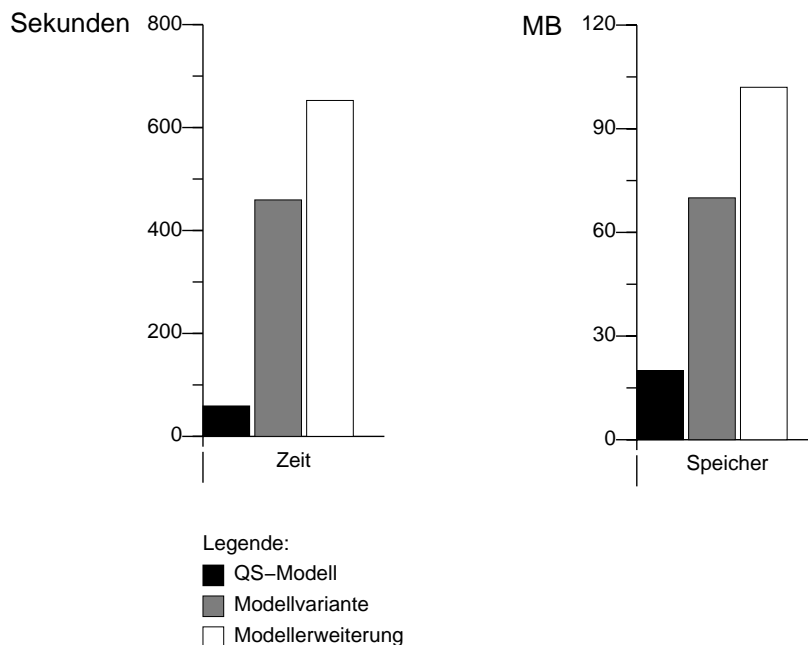
Die Ausführungszeit wird von der Ladezeit des Modells beeinflusst. Ein Vergleich zur Variante 1 zeigt, dass der Simulator für das erweiterte Modell anstatt 13 s rund 30 s benötigt, mehr als doppelt so lange (Hypothese H 25.2, Seite 26). Dabei spielt vermutlich die Größe des Modells eine entscheidende Rolle, die zu ladende Basissprachendatei der Variante 2 hat 19 035 ELOC (Executable Lines of Code), die Variante 1 ohne

Erweiterungen 12 230 ELOC. Der Einfluss der Modellgröße auf die Ladezeit ist also überproportional.

Ein weiterer Einfluss auf die Ausführungszeit ist die Dauer eines Simulationsschritts. Dazu wurde das Modell ausgeführt, ohne dass Entwickler Tätigkeiten wahrnehmen oder der Projektleiter eingreift. Ein Simulationsschritt der Variante 2 dauert rund 290 ms, während für einen Schritt Variante 1 (ohne zusätzliche Effekte) nur 170 ms gebraucht werden (Hypothese H 25.1, Seite 26). In der feingranularen Variante müssen mehr Regeln geprüft werden, die Regeln für die Arbeitszeit werden zusätzlich durchgeführt.

Das Antwortverhalten auf Kommandos, die in beiden Modellen zur Verfügung stehen, ändert sich kaum (Hypothese H 25.3). Für das Kommando "Lasse\_spezifizieren" benötigt der Simulator in der Modellvariante 1,7 ms, mit der Variante 2 mit dem komplexeren Situationsmodell 1,8 ms. In den zusätzlichen Kommandos der Variante 2 müssen mehr Parameter angegeben werden, vor allem Entitäten. Diese müssen gebunden werden. Kommandos für die Initialisierung des Reviews oder für die Einladung der Gutachter brauchen rund 6 ms.

Die beiden folgenden Balkendiagramme zeigen die durchschnittliche Ausführungszeit und den Speicherbedarf für die drei untersuchten Modelle. Sie zeigen, dass die Schrittweite den stärksten Einfluss auf die Ausführungszeit und den Speicherbedarf hat.



**Abb. 54:** Ausführungszeit und Speicherbedarf im Vergleich



# Kapitel 8

## Fazit und Ausblick

Das Kapitel teilt sich in drei Abschnitte. Im ersten Abschnitt beschreibe ich meine Erfahrungen, die ich mit der feingranularen Variante gemacht habe. Im zweiten Abschnitt wird ein Überblick über den Projektverlauf gegeben. Die erzielten Ergebnisse werden kurz zusammengefasst. Der dritte Abschnitt zeigt als Ausblick, welche Schwächen der feingranularen Variante noch behoben werden sollten und welche Möglichkeiten sich durch die Variante ergeben.

### 8.1 Erfahrung mit einer feingranularen Modellvariante

In diesem Abschnitt fasse ich meine Erfahrungen mit der Variante zusammen. Diese Erfahrungen sind stark subjektiv gefärbt.

#### 8.1.1 Konzeption und Realisierung der feingranularen Variante

##### Einbettung der Erweiterung

Bei der Definition der Effekte und Zusammenhänge für das Review hat sich gezeigt, dass die Einbettung bereits bei der Modellkonzeption berücksichtigt werden muss. Ein Beispiel dafür ist die Modellierung der Fehler. In der feingranularen Variante wäre eine Gewichtung der entdeckten Fehler durch die Gutachter sinnvoll, ohne eine Änderung der Fehlermodellierung im gesamten Modell ist diese Gewichtung aber nicht möglich.

##### Die Hochsprache

Prinzipiell hat sich die Hochsprache als ausreichend erwiesen. Nützlich sind aber Erweiterungen, die bereits Eisenbarth und Rohrbach (1998) bei der Erstellung des SAM vorgeschlagen haben. Abstrakte Datentypen für Attributtypen würden die Regeln verkürzen. Relationen und Entitäten als Funktionsparameter wären für die Einbettung hilfreich: Es werden Entitäten als Schnittstellen benutzt, durch Funktionen könnten definierte Schnittstellen geschaffen werden.

Die Aufteilung in einzelne Module wird durch die Hochsprache nicht unterstützt. Hier bietet der Einsatz der Entwicklungsumgebung SesamIDE Vorteile. Die Einbindung ganzer Module könnte aber noch besser unterstützt werden, zur Zeit muss der Modellbauer noch alle Dateien einzeln in die richtige Reihenfolge für Schema- und Regelmodell einsortieren.

## 8.1.2 Simulationen mit der feingranularen Variante

In der Testphase wurde die Modellvariante auch interaktiv am Simulator gespielt. Dabei macht sich die längere Ausführungszeit bemerkbar. Es tritt bereits beim Überspringen des Wochenendes eine Verzögerung von wenigen Sekunden auf.

Ein weiterer Nachteil der feingranularen Variante sind die vielen Parameter der Kommandos, etwa zur Initialisierung des Reviews oder um Gutachter einzuladen. Ich habe die Möglichkeiten zur Nachfrage durch den Dolmetscher ausgenutzt, um dem Spieler etwas Hilfe zu geben. Der Spieler kann aber auch alle Kommandos ohne Nachfragen eingeben.

Als nützlich hat sich der Kalender des Organizers bei der Planung und Durchführung der Reviews erwiesen. Mit dem Kalender kann man sich einen Überblick über die nächsten Tage verschaffen und erkennt auf einen Blick, wann beispielsweise Wochenenden sind. Wird der Kalender mitgeführt und enthält die Sitzungen, lässt sich ein Überblick über die verschiedenen Sitzungen und eingesetzten Mitarbeiter erreichen.

Auf mich haben die Spiele mit der feingranularen Variante realistisch gewirkt. Während der Reviews hat der Projektleiter viel zu tun, er muss den Überblick behalten und die Termine der Sitzung koordinieren. Obwohl ich die feingranulare Variante gut kenne, hat sich das manchmal als schwierig erwiesen. Die Empfehlung wirkte auf mich deutlicher als die Anzahl der entdeckten Fehler und Verluste. Ich vermute, dass sich die Spieler stark an der Empfehlung orientieren werden. Wie in realen Projekten müssen sie für eine sorgfältige und genaue Beurteilung des Prüflings sorgen (vgl. Freedman, Weinberg, 1982).

Durch die zufälligen Störungen und die geplanten Sitzungen wirkt das Spiel auf mich lebendiger und abwechslungsreicher. Durch die Störungen ist das Spiel weniger vorhersehbar. Die Ergebnisse der Reviewsitzung habe ich manchmal mit Spannung erwartet. Dazu trägt meiner Meinung nach auch die Schrittweite von zwei Stunden bei, die Effekte verteilen sich dadurch über den ganzen Tag.

## 8.2 Rückblick auf die Arbeit

### 8.2.1 Projektverlauf

Im ersten Schritt der Arbeit wurde der Projektplan erstellt, der den Zeitplan für die Arbeit vorgibt. Im Zeitplan war eine Woche Puffer eingeplant. Der Puffer war gedacht, um Verzögerungen oder umfangreiche Korrekturen der Dokumente aufzufangen. Dazu wurde der Puffer dann im Verlauf des Projekts auch gebraucht. Der Zeitplan musste für die Hypothesen und während der Konzeption des Modells angepasst werden.

- Die Hypothesen mussten nach der ersten Abgabe stark überarbeitet werden. Dafür konnte Puffer verwendet werden, die folgende Umstellung des QS-Modells auf zwei

Stunden Schrittweite war weniger aufwändig als geplant. Der hohe Aufwand für die Hypothesen war meiner Meinung nach notwendig. Sie sind die Grundlage der Arbeit und beeinflussen alle weiteren Phasen.

- Durch die Einbettung der Erweiterung in das QS-Modell sind die verwendeten Metriken weitgehend vorgegeben. Der Aufwand hat sich von der Auswahl der Metriken zur Definition der Effekte und Zusammenhänge verlagert.
- Die Quantifizierung wurde bei der Planung unterschätzt. Anstatt einer Woche dauerte die Quantifizierung zwei Wochen. Dafür wurde teilweise Puffer verwendet, die Implementierung hat sich verschoben.

Meilenstein	Termin Soll	Termin Ist
Projektplan	19.03.2001	19.03.2001
Hypothesen	10.04.2001	20.04.2001
Umstellung des QS-Modells und Test	11.05.2001	11.05.2001
Anforderungen an die feingranularen Variante	18.05.2001	18.05.2001
Effekte und Zusammenhänge	01.06.2001	05.06.2001
Definition der Metriken	15.06.2001	15.06.2001
Quantifizierung	27.06.2001	04.07.2001
Implementierung	27.07.2001	01.08.2001
Test der feingranularen Variante	10.08.2001	10.08.2001

**Abb. 55:** Wichtige Meilensteine und Termine

Die Abbildung zeigt die wichtigen Meilensteine der Arbeit, die geplanten Termine und die erreichten Termine. Nicht aufgeführt sind zwei Wochen Einarbeitung.

## 8.2.2 Erzielte Ergebnisse

### 8.2.2.1 Die feine Zeitgranularität

In dieser Diplomarbeit hat sich durch die Überprüfung der Hypothesen meiner Meinung nach gezeigt, dass eine feinere Zeitgranularität Vorteile bietet: Der Spieler kann den Tagesablauf eines Projektleiters erfahren. Er kann beim Spielen feststellen, wie lange kurze Tätigkeiten der Entwickler dauern und wann genau bestimmte Ereignisse stattfinden. Die kurze Schrittweite erlaubt, die Effekte von Ereignissen, die an einem Tag stattfinden, getrennt zu betrachten. Nachteilig wirken sich vor allem die erhöhte Ausführungszeit und der Speicherbedarf aus. Für den Einsatz einer kurzen Simulationsschrittweite wird daher das Ausbildungsziel eines Modells die entscheidende Rolle spielen.

### **8.2.2.2 Verhalten der Mitarbeiter**

Das Verhalten der Mitarbeiter wurde in dieser Arbeit neu modelliert. Dazu zählen nicht nur dringende oder parallele Tätigkeiten, sondern auch die Teilnahme an Sitzungen. Ein Nachteil der Modellierung ist vor allem die Komplexität. Das Verhalten wird durch Bedingungen in den verschiedenen Regeln einzelner Tätigkeiten realisiert. Sollen weitere Tätigkeiten eingeordnet werden, müssen die Regeln von bereits bestehenden Tätigkeiten angepasst werden.

Meiner Meinung nach überwiegen die Vorteile: Die Modellierung arbeitet nur mit dem Situationsmodell und nicht über Konstrukte wie den Kausalitätsfluss. Sie funktioniert unabhängig von der Schrittweite und erlaubt den Entwicklern eine flexible Aufteilung des Zeitschritts für die unterschiedlichen Tätigkeiten. Bei kurzer Schrittweite wird eine Tätigkeit über mehrere Zeitschritte durchgeführt, bei langer Schrittweite wird der Zeitschritt aufgeteilt. Allerdings ist das Verhalten der Entwickler erst bei einer kurzen Schrittweite realistisch, da sich nur dann die Entwickler mehrmals am Tag für eine andere Tätigkeit entscheiden können.

### **8.2.2.3 Modellierung des Reviews**

Ein weiteres Ergebnis der Arbeit ist die feinere Modellierung des Reviewprozesses für die Spezifikation und den Entwurf. Die Vorbereitung und die Sitzung sind getrennt modelliert, der Spieler muss Reviews genauer planen und sich um die Durchführung kümmern. An der Vorbereitung und der Sitzung konnte der Nutzen einer feineren Zeitgranularität gezeigt werden. Für diese kurzen Tätigkeiten wurde das Verhalten der Entwickler neu modelliert. Damit kann dieses Reviewmodell unabhängig von der Schrittweite eingesetzt werden. Es zeigt, dass unabhängig von der Schrittweite feingranulare Effekte modelliert werden können.

Die Prüfung der Quantifizierung zeigt im Vergleich zur grobgranularen Variante Unterschiede, etwa beim Aufwand oder beim Einsatz von drei Gutachtern. Die Quantifizierung konnte im Rahmen der Arbeit nur teilweise zufriedenstellend gelöst werden. Die Unterschiede werden in Abschnitt 7.2 kritisch betrachtet.

Ein weiteres Problem sind die Effekte, deren Quantifizierung nicht durch Literatur belegt werden konnte. Dazu zählen vor allem die Charaktereigenschaften der Mitarbeiter, damit die Frustration des Autors modelliert werden kann. Der Autor beschwert sich mit der gewählten Quantifizierung nur in extremen Fällen, in denen die Eigenschaften der Reviewteilnehmer stark verzerrt sind. Die Testfälle für diesen Effekt sind stark konstruiert, soll der Effekt bei Spielen mit weniger karikaturhaften Entwicklern auftreten, müsste an dieser Stelle die Quantifizierung angepasst werden.

## **8.3 Ausblick**

### **8.3.1 Fehlende Effekte und Kommandos**

In der Erweiterung fehlen noch zwei Effekte. Die Rolle des Aktuars konnte nicht mehr modelliert werden. Außerdem vergessen Gutachter die vorbereiteten Befunde, wenn eine Sitzung abgebrochen wurde. Diese beiden Effekte sollten noch implementiert werden. Außerdem kann die Schnittstelle zum Spieler noch stark verbessert werden: Es fehlen die Kommandos, mit denen das Review über den Termin identifiziert werden kann. Das Wörterbuch ist noch eingeschränkt.

### **8.3.2 Hypothesen zur Ausbildung**

Im Rahmen der Diplomarbeit konnten nicht alle Hypothesen überprüft werden. Nicht überprüfte Hypothesen betreffen die Ausbildung, sie können wie in Abschnitt 7.3.2.6 beschrieben nur durch Spiele am Simulator geprüft werden. Spiele mit der Variante, etwa im Rahmen einer Fallstudie, würden zeigen, welche Auswirkungen auf die Ausbildung tatsächlich auftreten.

### **8.3.3 Validierung der feingranularen Variante**

Das QS-Modell ist mit der Projektgröße parametrisierbar. Die feingranulare Variante wurde in dieser Arbeit nur gegen Projekte mit 200 AFP Umfang geprüft. Dabei hat sich gezeigt, dass der Aufwand und die Kosten stark abweichen. Der Aufwand sollte für Reviews noch stärker angepasst werden, er liegt über den typischen 20% Aufwand, die vom Erstellungsaufwand für die Prüfung verwendet werden (Frühauf et al., 2000). Die höheren Kosten entstehen durch zusätzliche Mitarbeiter. Durch den Einsatz von externen Mitarbeitern können diese Kosten gesenkt werden, diese Möglichkeit wird im Abschnitt 8.3.4 näher betrachtet.

Die Variante konnte im Rahmen dieser Arbeit nicht mehr mit anderen Projektgrößen verglichen werden. Ich vermute Unterschiede beim Aufwand, da im QS-Modell die Fehlerrate von der Projektgröße abhängt. Sitzungen in der Variante sind von der Anzahl der entdeckten Fehler und Verluste abhängig, die Sitzungen werden vermutlich länger dauern. Eine gründliche Validierung, auch gegen Daten aus der Literatur, ist sicher noch notwendig.

### **8.3.4 Externe Mitarbeiter**

Im QS-Modell ist die Rolle des Moderators verkürzt. In der feingranularen Variante muss der Spieler vor allem in den frühen Phasen für diese Rolle einen weiteren Mitarbeiter einstellen. Besonders deutlich schlägt sich der Mitarbeiter in den Projektkosten nieder. Bei kleinen Software-Projekten mit 200 AFP Umfang werden aber nur wenig Mitarbeiter benötigt. Eine Schätzung mit Cocomo (Boehm, 1981) ergibt für Projekte mit 200 AFP für die Spezifikationsphase nur 1,6 FSP (Full Software Personal) und 2,4 FSP für

den Entwurf. Es werden deutlich weniger Mitarbeiter geschätzt, als durch die Modellierung für die Reviews gebraucht werden.

In der Literatur werden für das Spezifikationsreview sogar fünf Gutachter angegeben (Frühauf et al., 2000, Seite 104). Es wird empfohlen, Teilzeitgutachtern aus anderen Projekten oder aus dem Qualitätswesen (Freedman, Weinberg, 1982) einzuladen. Diese Teilzeitgutachter müssen dann nur für die Prüfung bezahlt werden. Für den Moderator ist es sinnvoll, wenn er am Projekt nicht direkt beteiligt ist, er kann etwa aus dem Qualitätswesen kommen (vgl. Frühauf et al., 2000, Seite 106). Für Reviews wäre eine Erweiterung des Modells um Teilzeitmitarbeiter oder externe Mitarbeiter sinnvoll.

### **8.3.5 Einarbeitung der Gutachter**

Versucht der Spieler, Kosten zu sparen und stellt die Teilnehmer spät ein oder entlässt sie zwischen den Reviews wieder, müssen sich die Teilnehmer einarbeiten. Mitarbeiter lernen während der Einarbeitung die Projektumgebung, die Aufgabe oder verwendete Methoden kennen (Drappa, 2000, Seite 63). Arbeitet sich ein Gutachter ein, steigt der Aufwand für das Review und es werden weniger Befunde entdeckt. Im Zusammenhang mit externen Mitarbeitern entsteht dabei ein Widerspruch: Während für Projektmitarbeiter die Einarbeitung als Effekt modelliert wird, wird von externen Mitarbeitern eine Einarbeitung nicht erwartet.

In einer Einführungssitzung wird der Prüfling, das Umfeld und der Zweck des Reviews vorgestellt. Der Projektleiter könnte den Moderator veranlassen, eine Einführungssitzung durchzuführen. Wird die Einführungssitzung durchgeführt, ist eine Einarbeitung für das Review nicht mehr nötig. Die feingranulare Variante sollte daher um die Einführungssitzung erweitert werden.

### **8.3.6 Die feingranulare Erweiterung und das QSVA-Modell**

Der Nutzen einer kurzen Simulationsschrittweite zeigt sich vor allem bei Ereignissen oder kurzen Tätigkeiten. Das Modell wurde um wenige Störungen des Projektleiters und Treffen mit Mitarbeitern erweitert, der Schwerpunkt liegt auf dem feineren Review.

Im QSVA-Modell (Dudler, 2000) sind Störungen und Sitzungen des Projektleiters bereits umfassend enthalten: Mitarbeitergespräche, verschiedene Statusmeetings und unterschiedliche Motivationsmaßnahmen sind möglich. Eine Umstellung des QSVA-Modells auf eine feinere Zeitgranularität erlaubt dem Spieler, zu beobachten, wann diese feingranularen Effekte stattfinden und welchen Einfluss die Effekte auf seinen Tagesablauf haben. Dafür sind dort die Tätigkeiten der Mitarbeiter für die Arbeit an den Dokumenten unverändert. Wird das QSVA-Modell zusätzlich um feingranulare Reviews erweitert, würde ein Modell mit feinen Tätigkeiten der Entwickler und umfassenden Effekten auf den Tagesablauf des Projektleiters entstehen. Die Motivation des Autors wird eingebunden.

## Begriffslexikon

Das Begriffslexikon erklärt die grundlegenden Begriffe in dieser Arbeit. Das Begriffslexikon von SESAM (Reißing, 1996b) enthält die definierten Begriffe des SESAM-Systems.

### Abstraktionsstufe

Ein Softwareprozess kann auf unterschiedlichen *Abstraktionsstufen* betrachtet werden. Die *Abstraktionsstufen* bestimmen, wie detailliert die Modellierung erfolgt.

### Anforderung

In SESAM-Modellen wird der Inhalt aller *Dokumente* durch *Anforderungen* beschrieben. Eine *Anforderung* kann nicht mehr geteilt werden und wird durch ihre Komplexität in AFP beschrieben. Weitere wichtige Attribute sind *Fehler* getrennt nach *Fehlerarten* und der Autor der *Anforderung* in einem bestimmten *Dokument*.

### Arbeitspaket

Im QS-Modell enthalten *Arbeitspakete* eine Menge von *Anforderungen*. Diese *Anforderungen* werden von einem *Entwickler* bearbeitet. *Arbeitspakete* werden benutzt, damit mehrere *Entwickler* gleichzeitig ein *Dokument* bearbeiten können, ohne sich gegenseitig zu stören.

### Aspekt

*Fragenkataloge* sind nach *Aspekten* gegliedert. Mit jedem *Aspekt* wird eine bestimmte Eigenschaft des *Dokuments* geprüft.

### Ausführungszeit

Die *Ausführungszeit* eines Simulationslaufs ist der Abschnitt in der realen Zeit, über den der Animationsvorgang vom Start bis zum endgültigen Abbruch andauert. (Schneider, 1994)

### Befund

Im *Review* entdeckte Mängel im *Prüfling* werden als *Befund* formuliert. Im Modell ist ein *Befund* einer *Anforderung* zugeordnet und enthält die entdeckten *Fehler* und *Verluste* der *Anforderung*. In der Vorbereitung sind *Befunde* zusätzlich nach *Aspekten* aufgeteilt.

### Dokument

In SESAM-Modellen verwendete Entitätsklasse zur Modellierung der in Software-Projekten entstehenden und verwendeten *Dokumente*.

### Dynamik des Modells

Mögliche Veränderungen des Situationsmodells im Projektverlauf werden durch die *Dynamik des Modells* beschrieben. Die *Dynamik* wird durch Regeln und Kommandos modelliert.

**Empfehlung, synonym Reviewzusammenfassung**

In einer Reviewsitzung wird der *Prüfling* bewertet. Der Projektleiter bekommt eine *Empfehlung* mit der Bewertung des *Prüflings* (siehe Frühauf et al., 2000, Seite 93).

**Fehler**

*Fehler* werden im QS-Modell getrennt nach *Fehlerarten* durch ihre Anzahl beschrieben und sind einer *Anforderung* zugeordnet.

**Fehlerart**

Durch *Fehlerarten* wird unterschieden, bei welcher Tätigkeit ein *Fehler* in eine *Anforderung* eingefügt wurde. *Fehlerarten* sind Analysefehler, Grobentwurfsfehler, Feinentwurfsfehler, Implementierungsfehler und Handbuchfehler.

**Fragenkatalog**

*Fragenkataloge* sind ein wichtiges Hilfsmittel im *Review*. Sie enthalten Richtlinien und Fragen, unter denen der *Prüfling* begutachtet werden soll (vgl. Frühauf et al., 2000, Seite 100).

**Konfiguration des Modells**

Eine *Konfiguration des Modells* in der Hochsprache beschreibt, wie die einzelnen Teile zusammengesetzt werden sollen, damit das Modell in die Basissprache übersetzt werden kann. Durch *Konfigurationen* ist es möglich, verschiedene Startzustände für das gleiche Modell zu verwenden.

**Mitarbeiter, synonym Entwickler**

Entitätsklasse in einem SESAM-Modell zur Modellierung von *Entwicklern* in einem Software-Projekt. Der Projektleiter kann *Entwickler* einstellen, entlassen und zur Bearbeitung von *Dokumenten* verwenden.

**Modellgranularität**

Die *Modellgranularität* legt den Detaillierungsgrad und das Auflösungsvermögen einer Modellierung fest. (Schneider, 1994, Seite 74)

**Modellzeit**

Die *Modellzeit* ist eine Abbildung der Originalzeit des Projektes. Sie entspricht einer um den Startzeitpunkt nach hinten verschobenen Originalzeit. (Schneider et al., 1999)

**Produktivität**

Die *Produktivität* wird im QS-Modell als Eigenschaft eines *Entwicklers* modelliert. Sie beschreibt, wie viele AFP der *Entwickler* pro Stunde bearbeiten kann.

**Prüfbericht**

Eine Entitätsklasse im QS-Modell, in der *Befunde* aus *Reviews* gesammelt werden, bis sie korrigiert werden. Entspricht der Liste der *Befunde* aus Frühauf et al. (2000, Seite 93).



**Prüfling**

Als *Prüfling* wird der Teil eines *Dokuments* bezeichnet, der in der Vorbereitung für die Sitzung von den Gutachtern geprüft und in der Sitzung besprochen wird.

**Review, synonym technisches Review**

Ein *Review* ist eine Maßnahme zur Qualitätssicherung. Der *Prüfling*, ein *Dokument*, wird von *Entwicklern* nach *Vorgaben* begutachtet und in einer Sitzung beurteilt. Ein *Review* ermöglicht Fortschrittskontrolle und Prüfung in den frühen Phasen. In diesem Bericht wird damit das technische *Review* nach Frühauf et al. (2000) bezeichnet

**sesamIDE**

Das Werkzeug *sesamIDE* unterstützt den Modellbauer. Es bietet eine Entwicklungsumgebung für die Hochsprache und verwendet den Hochsprachencompiler.

**Tätigkeit**

*Entwickler* führen in einem simulierten Projekt eine *Tätigkeit* durch, wenn sie ein *Dokument* bearbeiten.

**Verlust**

Wenn eine *Anforderung* unvollständig von einem *Dokument* in ein anderes *Dokument* übertragen wird, wird der Teil der nicht übertragenen *Anforderung* als *Verlust* bezeichnet.

**Verschnitt, synonym Zeitverschnitt**

Als *Verschnitt* wird die Zeit bezeichnet, die ein *Entwickler* in einem Zeitschritt nicht verbrauchen kann, weil der *Entwickler* eine *Tätigkeit* in einem Zeitschritt beendet hat und dafür nicht den gesamten Zeitschritt braucht.

**Vorgabe, synonym Vorgabedokument**

Als *Vorgabe* wird das *Dokument* bezeichnet, gegen das geprüft wird. Gutachter vergleichen den *Prüfling* gegen die *Vorgabe*.

**Zeitgranularität**

Die *Zeitgranularität* bestimmt die kleinste sichtbare Zeiteinheit im Modell. Sichtbare Zeiteinheit bedeutet, dass Veränderungen des Zustandes nur im Abstand dieser Zeiteinheit außerhalb des Simulators beobachtet werden können. Die *Zeitgranularität* bestimmt die Schrittweite. Der Simulator schaltet in jedem Simulationsschritt die *Modellzeit* um die Schrittweite weiter. Innerhalb des Modells kann aber in kleineren Zeiteinheiten gerechnet werden.

**Zeitmodell**

Abbild der Arbeitszeit eines Software-Projektes in ein SESAM-Modell. Das *Zeitmodell* bestimmt, wann im Modell gearbeitet wird und wann der Simulator Eingaben des Spielers entgegennimmt.

## Literaturverzeichnis

Boehm (1981)

Boehm, B.W.: **Software Engineering Economics**. Prentice-Hall, Englewood Cliffs, 1981.

Bossel (1992)

Bossel, H.: **Modellbildung und Simulation**. 2. Auflage, Vieweg, Braunschweig, 1994.

Bourgeois (1996)

Bourgeois, K. V.: Process Insights from a Large-Scale Software Inspections Data Analysis. **Cross Talk**, Oct 1996.

<http://www.stsc.hill.af.mil/CrossTalk/1996/oct/xt96d10e.asp>

Drappa (2000)

Drappa, A.: **Quantitative Modellierung von Softwareprojekten**. Dissertation, Shaker, Aachen, 2000.

Deiningner (1995)

Deiningner, M.: **Quantitative Erfassung der Software und ihres Entstehungsprozesses**. Dissertation, Dr. Kovac, Hamburg, 1995.

Dudler (2000)

Dudler, A.: **Quantitative Modellierung von Verhaltensaspekten in Softwareprojekten**. Diplomarbeit Nr. 1819, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 2000.

Eisenbarth, Rohrbach (1998)

Eisenbarth, T.; Rohrbach, J.: **Evaluation und Erprobung des SESAM-2-Systems anhand eines strikt atomaren Modells**. Gruppdiplomarbeit Nr. 1633, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1998.

Frühauf et al. (2000)

Frühauf, K.; Ludewig, J.; Sandmayr, H.: **Software-Prüfung. Eine Anleitung zum Test und zur Inspektion**. 4. Auflage, vdf, Zürich, 2000.

Freedman, Weinberg (1982)

Freedman, D. P.; Weinberg, G. M.: **Handbook of Walkthroughs, Inspections, and Technical Reviews**. 3. Auflage, Little, Brown, and Company, Boston, 1982.

Gilb, Graham (1993)

Gilb, T.; Graham, D.: **Software Inspection**. Addison-Wesley, Wokingham, 1993.

Hollocker (1990)

Hollocker, C. P.: **Software Reviews and Audits Handbook**. John Wiley & Sons., New York, 1990.

Jones (1996)

Jones, C.: **Applied Software Measurement**. 2. Auflage, McGraw-Hill, New York, 1996.

Joos (1997)

Joos, M.: **Konzeption und Realisierung eines Simulationsmodells für kleine Softwareprojekte**. Diplomarbeit Nr. 1543, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1997.

Kalajzic (2001)

Kalajzic, S.: **Meta-Regeln und Meta-Regel-Verletzungen**. Internes Dokument, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 2001.

Kuhnle (1998)

Kuhnle, J.: **Eine Untersuchung zum Nutzen von Reviews und Inspektionen**. Diplomarbeit Nr. 1598, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1998.

Ludewig (1994)

Ludewig, J.: **SESAM: Software-Engineering-Simulation durch animierte Modelle**. Fakultätsbericht Nr. 5/1994, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1994

Ludewig (1997)

Ludewig, J.: **Software Engineering. Skript zur Vorlesung**. Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1997.

Mandl-Striegnitz (2000)

Mandl-Striegnitz, P. in: Dumke, R.; Lehner, F. (Hrsg.): **Software Metriken: Entwicklung, Werkzeuge und Anwendungsverfahren**. Deutscher Universitätsverlag, Wiesbaden, 2000, 111 - 134.

Mandl-Striegnitz (1998)

Mandl-Striegnitz, P.: **Checklist for Design Reviews**. Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1998.  
[http://www.informatik.uni-stuttgart.de/ifi/se/service/checklists/download/Review\\_Design.html](http://www.informatik.uni-stuttgart.de/ifi/se/service/checklists/download/Review_Design.html)

Metzger, Boddie (1996)

Metzger, P.; Boddie, J.: **Managing a Programming Project - Processes and People**. 3. Auflage, Prentice Hall, Upper Saddle River, New Jersey, 1996.

Miller et al. (1998)

Miller, J.; Wood, M.; Roper, M.; Brooks, A.: Further Experiences with Scenarios and Checklists. **Empirical Software Engineering: An International Journal**, Vol. 3, No. 1, 1998, 37 - 64.

Porter et al. (1995)

Porter, A. A.; Votta, L. G.; Basili, V. R.: Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment. **IEEE Transactions on Software Engineering**, Vol. 21, No. 6, 1995, 563 - 575.

Porter, Johnson (1997)

Porter, A. A.; Johnson, P. M.: Assessing Software Review Meetings: Results of a Comparative Analysis of Two Experimental Studies. **IEEE Transactions on Software Engineering**, Vol. 23, No. 3, 1997, 129 - 145.

Porter et al. (1997)

Porter, A. A.; Harvey, P. S.; Toman, C. A.; Votta, L. G.: An Experiment to Assess the Cost-Benefits of Code Inspections in Large Scale Software Development. **IEEE Transactions on Software Engineering**, Vol. 23, No. 6, 1997, 329 - 346.

Reißing (1996a)

Reißing, R.: **Konzeption und Realisierung einer Basismaschine für SESAM-2**. Diplomarbeit Nr. 1345, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1996.

Reißing (1996b)

Reißing, R.: **Begriffslexikon für SESAM**. Internes Dokument, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1996.

Schneider (1994)

Schneider, K.: **Ausführbare Modelle der Software-Entwicklung. Struktur und Realisierung eines Simulationssystems**. Dissertation, vdf, Zürich, 1994.

Schneider et al. (1999)

Schneider, B.; Messner, M.; Müller, U.: **Modellbeschreibungssprache für SESAM-2**. Version 3.1. Internes Dokument, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1999.

Seaman, Basili (1998)

Seaman, C. B.; Basili, V. R.: Communication and Organization: An Empirical Study of Discussion in Inspection Meetings. **IEEE Transactions on Software Engineering**, Vol. 24, No. 7, 1998, 559 - 572.

Stachowiak (1973)

Stachowiak, H.: **Allgemeine Modelltheorie**. Springer, Wien, 1973.

Weller (1993)

Weller, E. F.: Lessons from Three Years of Inspection Data. **IEEE Software**, Vol. 10, No. 5, 1993, 38 - 45.

Würthele (1995)

Würthele, V: **Checklisten für die Software-Bearbeitung**. Diplomarbeit Nr. 1299, Abteilung Software Engineering, Institut für Informatik, Universität Stuttgart, 1995.

## **Erklärung**

Hiermit versichere ich, diese Arbeit selbständig verfasst  
und nur die angegebenen Quellen benutzt zu haben.

---

Tilmann Hampp