

Seminar „Beiträge zum Software Engineering“

Die Softwareentwicklung simulieren - Einführung an Beispielen, Einsatz in der Lehre

Sebastian Jekutsch

Freie Universität Berlin
Institut für Informatik
AG Software Engineering

Mr. Tompkins' Personalproblem

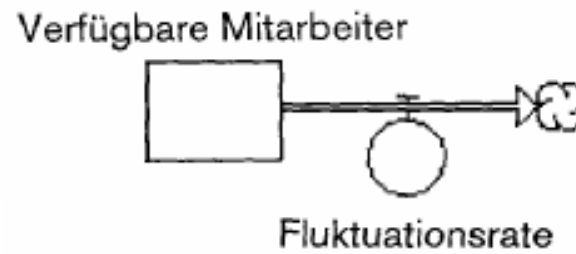
Modellierungsbeispiel (1/5)

Verfügbare Mitarbeiter



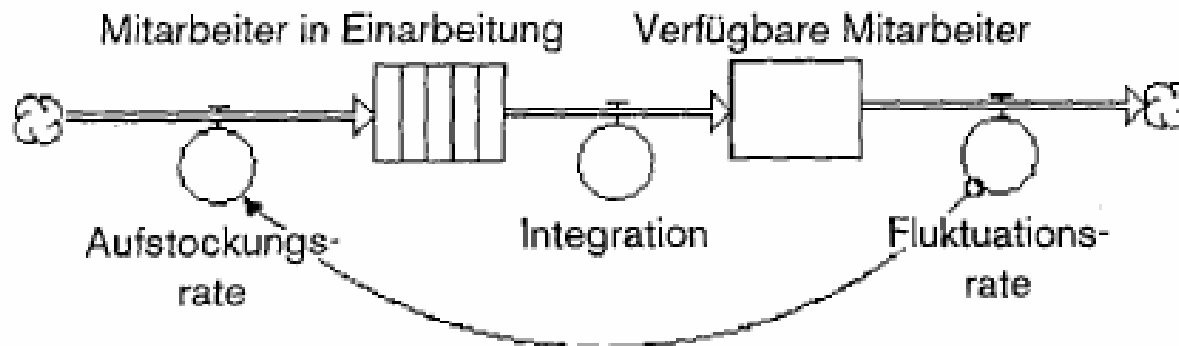
Mr. Tompkins' Personalproblem

Modellierungsbeispiel (1/5)



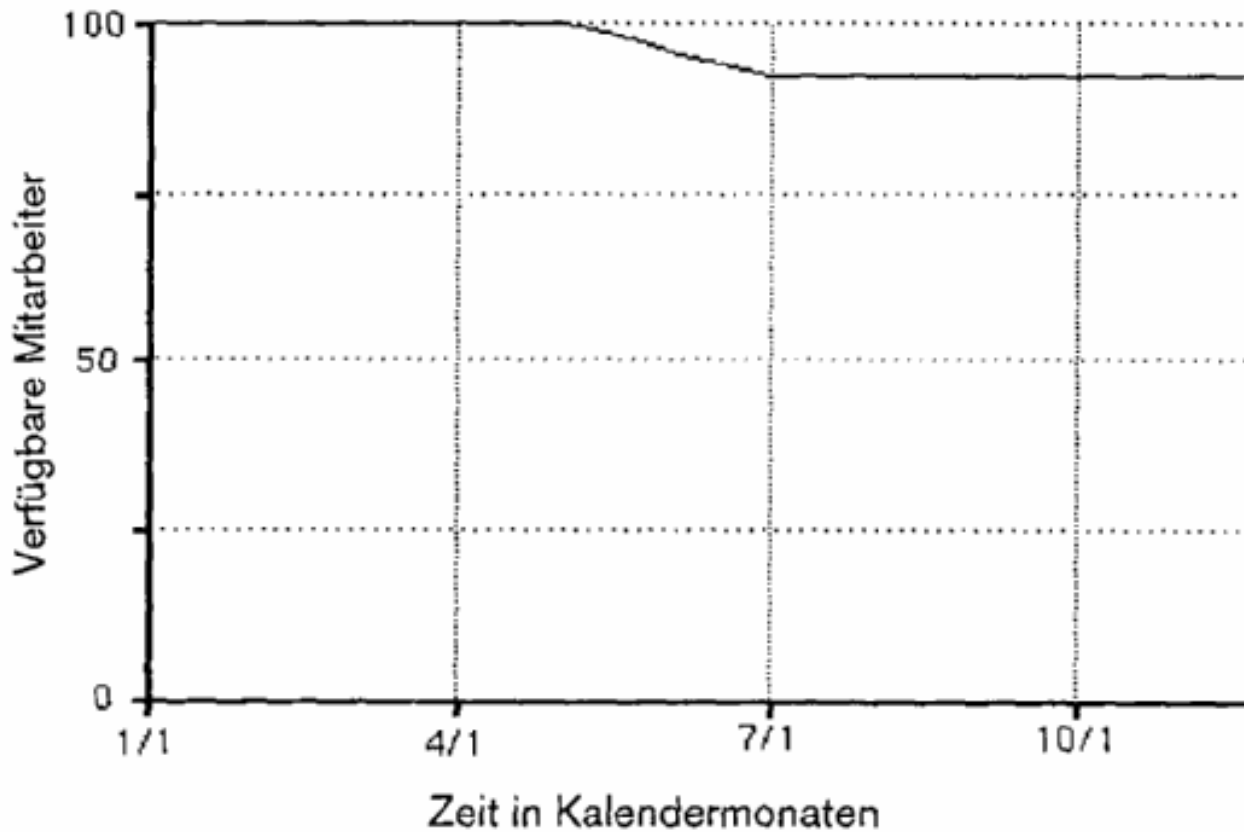
Mr. Tompkins' Personalproblem

Modellierungsbeispiel (1/5)



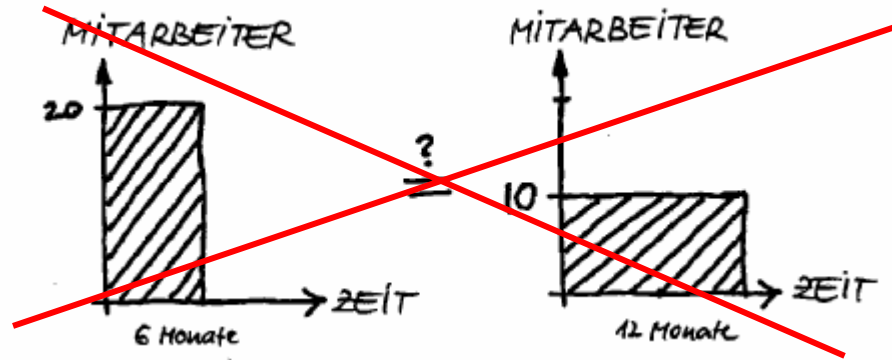
Erster Simulationsverlauf

Modellierungsbeispiel (2/5)

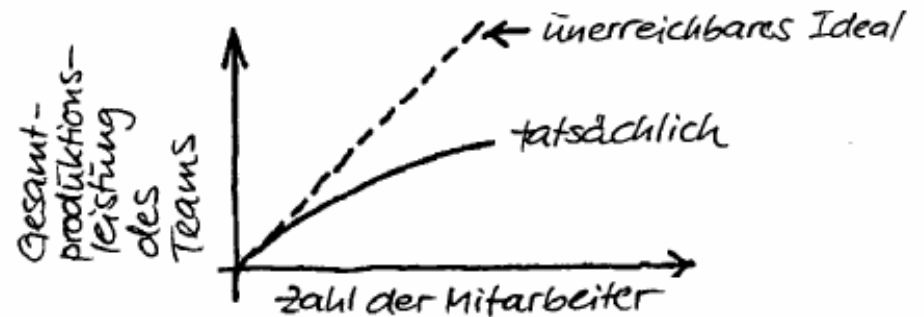
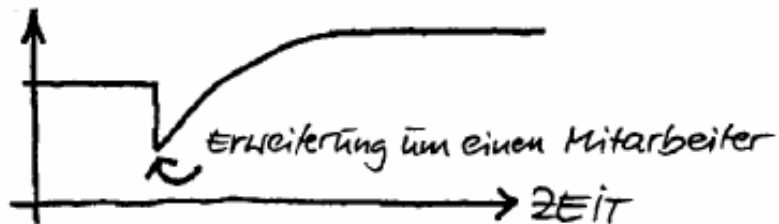


Verlauf der Teamproduktivität

Modellierungsbeispiel (3/5)

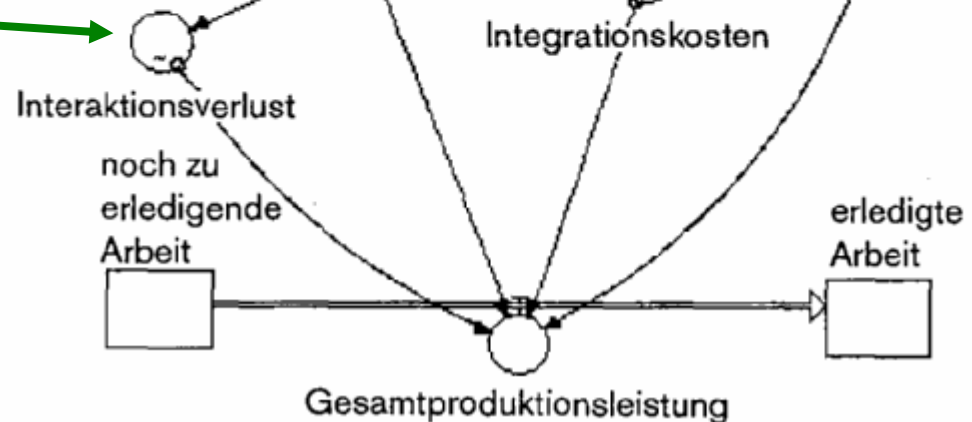
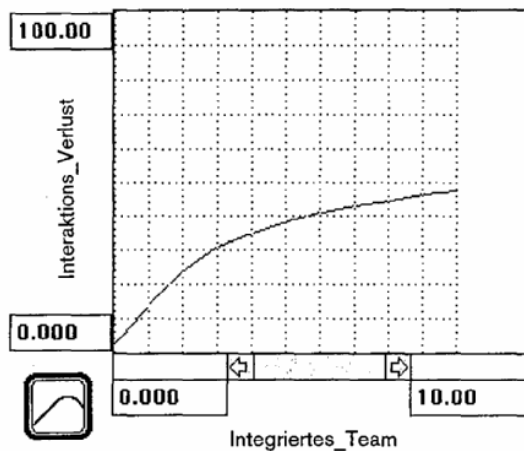
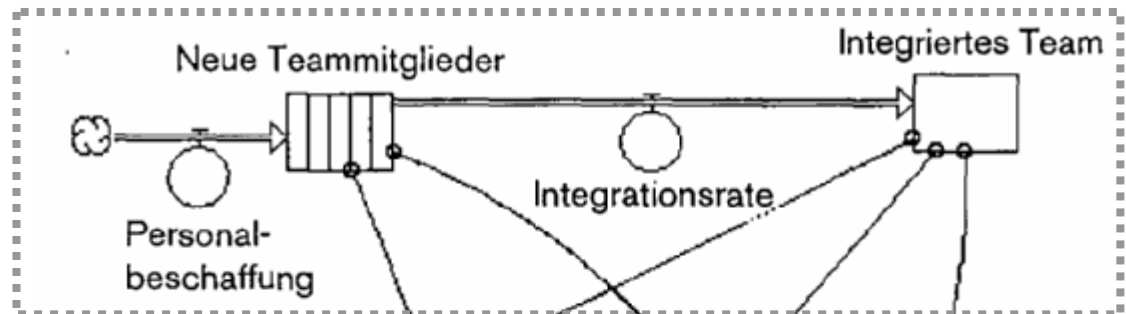
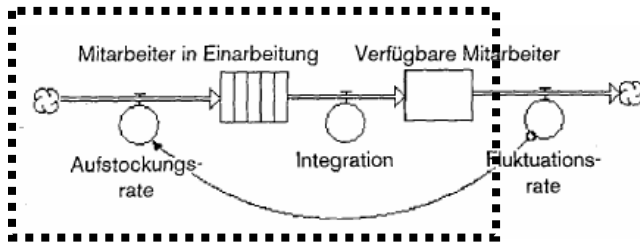


Produktionsleistung des Teams



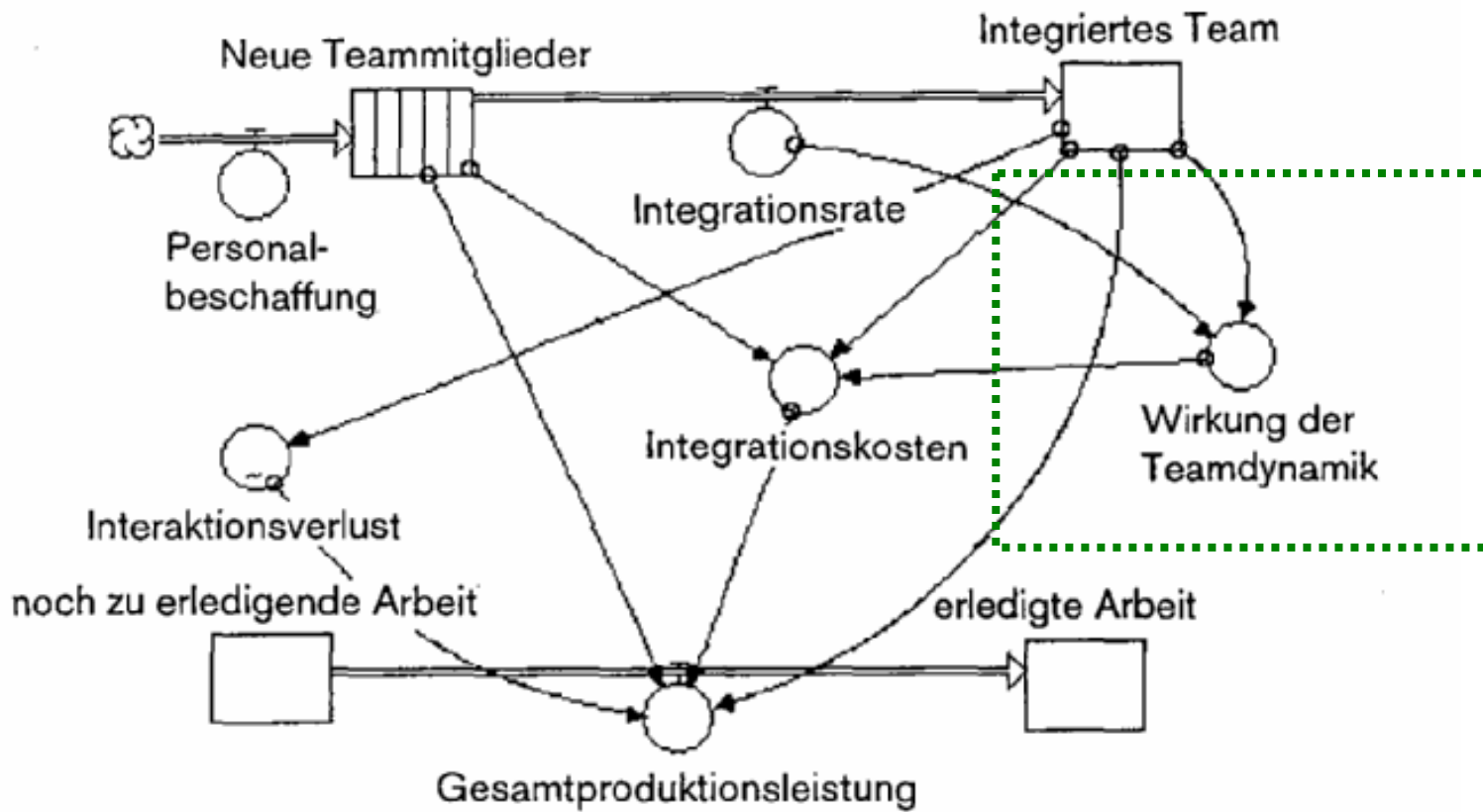
Erweiterung des Modells: Interaktionsverlust

Modellierungsbeispiel (4/5)



Anpassung des Modells: Teamdynamik

Modellierungsbeispiel (5/5)



1. *Modellierungsbeispiel*: Mr. Tompkins' Personalproblem
2. **Einführung: Was ist das? Was soll das?**
3. *Pionierarbeit*: Abdel-Hamid und andere
4. *Einsatzgebiete*: Chancen durch Simulation
5. *Ausbildung*: SESAM und AMEISE
6. *Systematik*: Was für Simulationen gibt es?
7. *Probleme*: Warum macht es nicht jeder?

- Zur Vorhersage
 - Was wird passieren?
 - z.B. Wetter
- Zur Evaluation
 - Wird es passieren?
 - z.B. Automotor
- Zur Erkundung
 - Was passiert, wenn?
 - z.B. Verkehrsplanung
- Zur Ausbildung
 - Was passiert?
 - z.B. Piloten



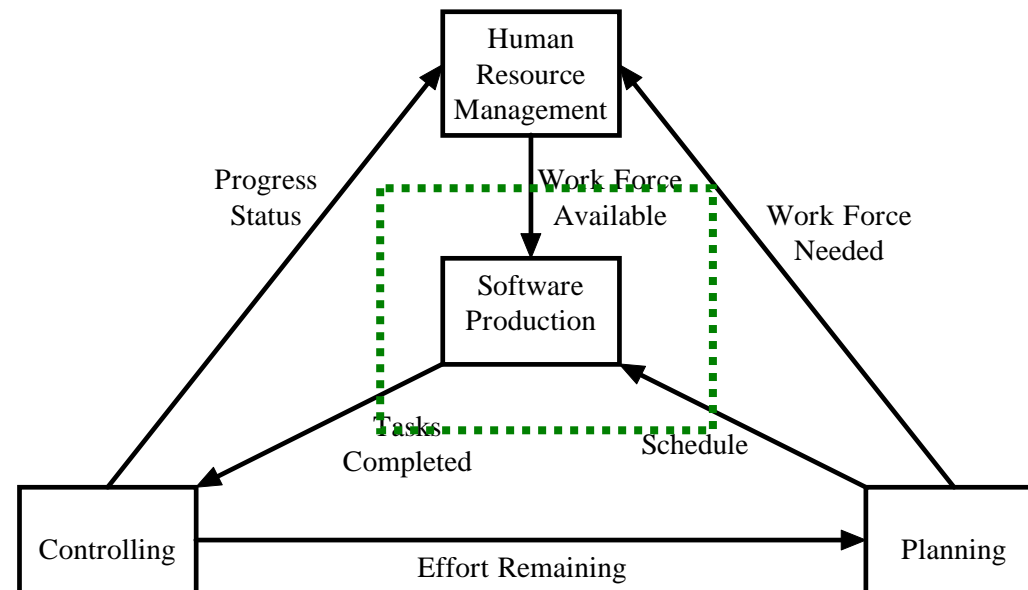
steigende
Interaktivität

Systeme, ...

- die sich in der Zeit verändern
 - automatisch, auch ohne Impulse von außen
 - fortschreitende Simulationszeit
- die (meist) von außen beeinflussbar sind
- die aus vielen, interagierenden Teilsystemen bestehen
 - komplex, stochastisch, mit Rückkopplungen
- deren Teilsysteme
 - ... selbst wieder simuliert werden
 - ... oder deren Verhalten ausreichend bekannt ist
 - durch Formeln, Expertenwissen oder Empirie
- die nicht rein analytisch beschreibbar sind
 - und zu komplex für Entscheidungsbäume

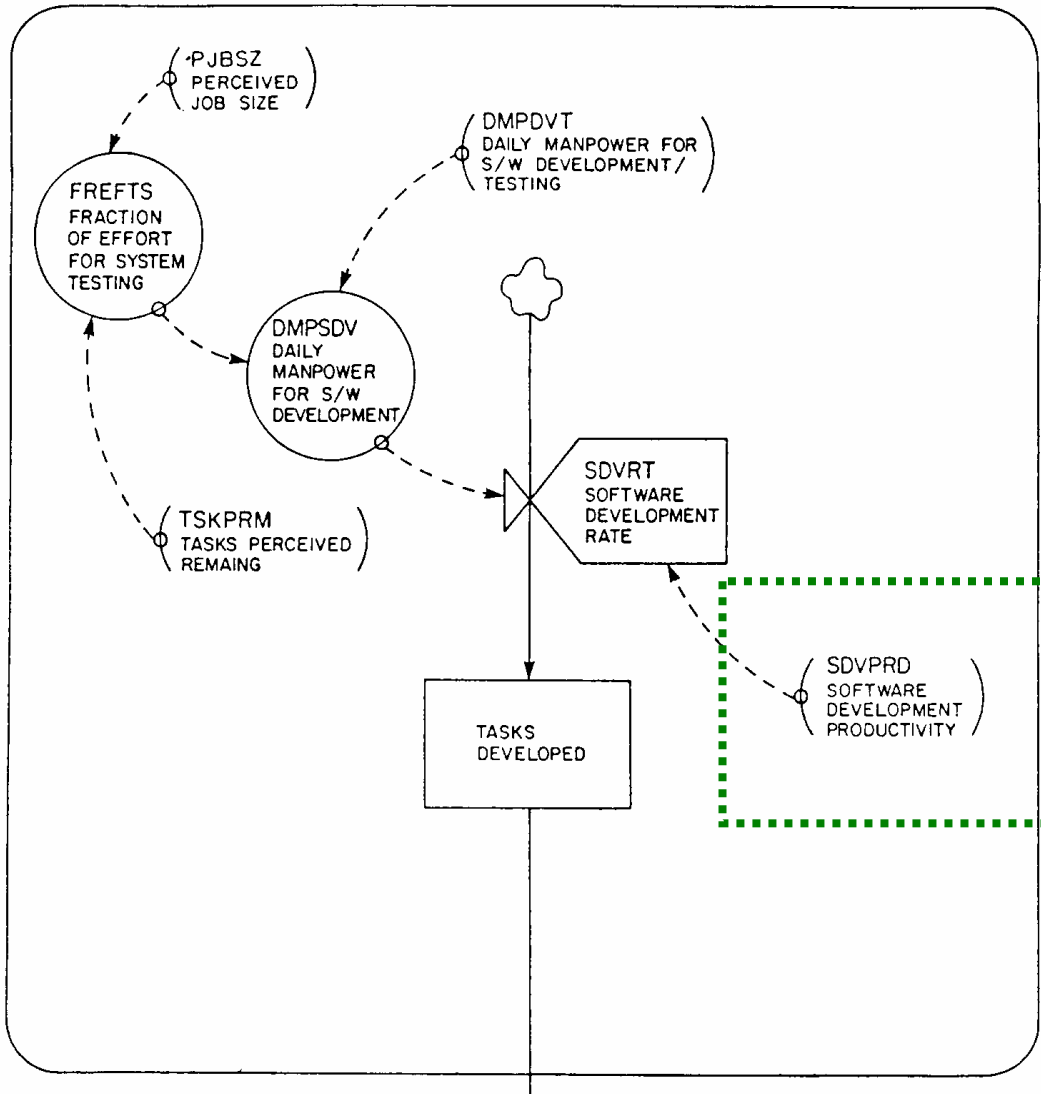
- Es wird inzwischen auch Software „simuliert“
 - Modellbasierter Ansatz (MDD)
 - also evaluierend oder erkundend
- Hier geht es um Software**prozess**simulation
 - also Dynamik der Softwareentwicklung, Projektplanung
 - meist erkundend und vorhersagend, auch lehrend
- Das „System der Softwareentwicklung“ besteht z.B. aus
 - Projekt, Qualitätsvorgaben
 - Mitarbeiter, Zeit, Geld
 - Techniken, Werkzeuge, Ausbildung
 - ... und deren Interaktion
- Es geht um kaum verstandene Systeme
 - COCOMO (Kostenschätzung) z.B. ist ein statisches Modell

- Abdel-Hamid (und Madnick) haben 1991 die erste umfangreiche Veröffentlichung zum Thema gehabt
 - Einfluss des Projektmanagements auf den Fortschritt der Softwareentwicklung (Anforderungen gegeben)
 - 270 Gleichungen
- Modell basiert auf umfangreiches Literaturstudium und Interviews



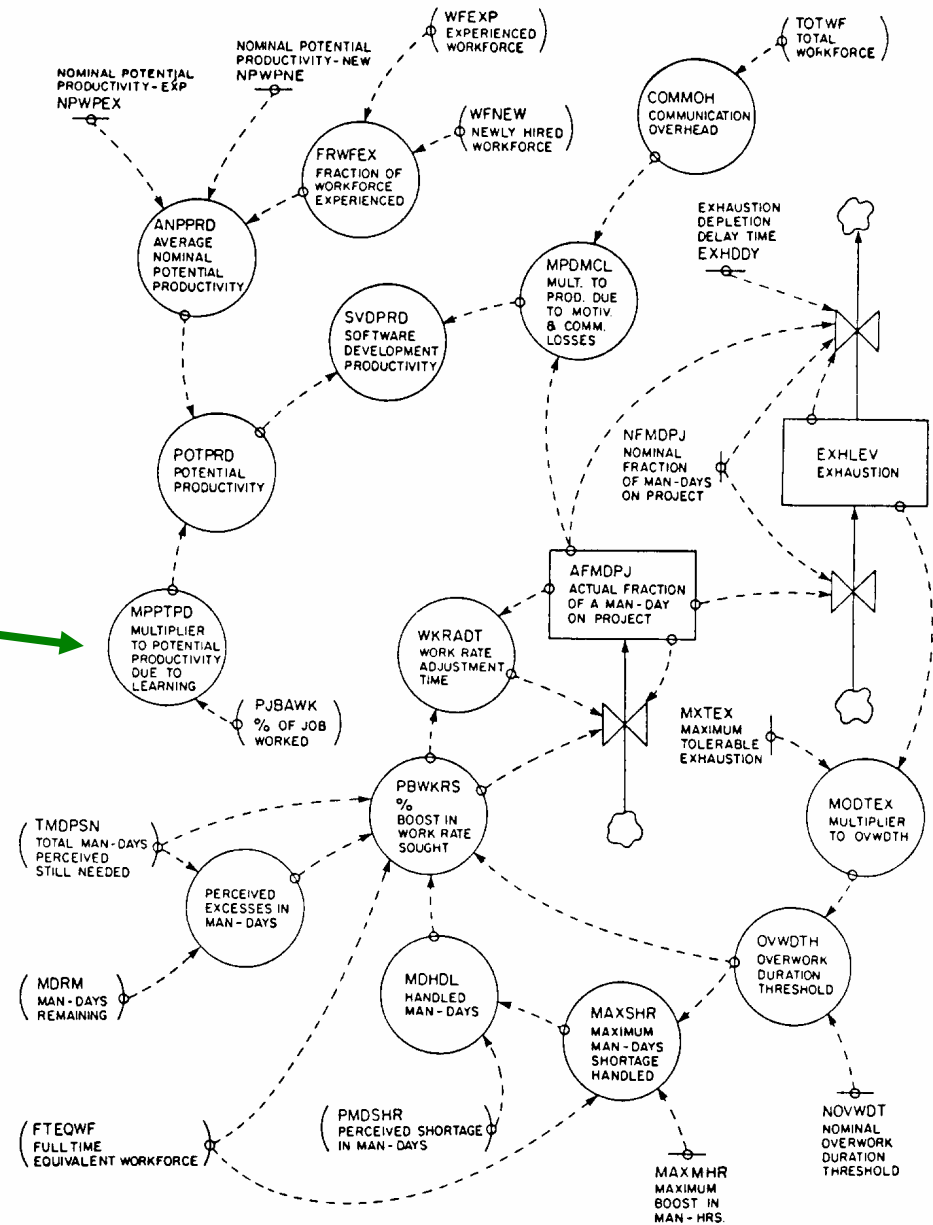
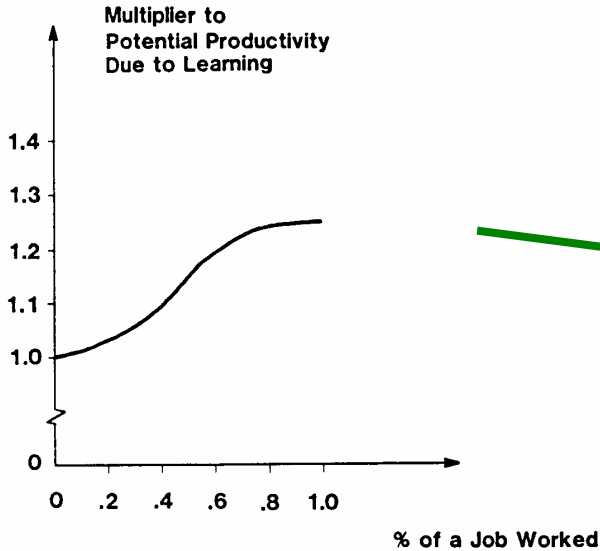
Modul 'Software Production'

Pionierarbeit (2/4)



Modul 'Productivity'

Pionierarbeit (3/4)



- Er hat die freien Parameter für ein abgelaufenes NASA-Projekt eingefügt und das Modell damit validiert
- Allgemeine Erkenntnisse darüber hinaus:
 - Es gibt eine Neigung, einen Planzeitrahmen voll auszunutzen (und darüber hinaus), egal wie der Plan aussieht (ähnlich Parkinson)
 - Zeitdruck kostet Geld (wegen Arbeitsstress)
 - Brooks Gesetz konnte nicht bestätigt werden
 - „Adding manpower to a late project makes it even later.“
- Andere Pionierarbeit von Kellner / Hansen (1989)

1. *Modellierungsbeispiel: Mr. Tompkins' Personalproblem*
2. *Einführung: Was ist das? Was soll das?*
3. *Pionierarbeit: Abdel-Hamid und andere*
4. ***Einsatzgebiete: Chancen durch SWT-Simulation***
5. *Ausbildung: SESAM und AMEISE*
6. *Systematik: Was für Simulationen gibt es?*
7. *Probleme: Warum macht es nicht jeder?*

- Probieren von Alternativen grundsätzlicher Art
 - *Sollte die Arbeit über mehrere Orte verteilt werden?*
 - *Sollte die Kodierarbeit an Dritte vergeben werden?*
 - *In welchem Umfang sollte COTS eingesetzt werden?*
 - *Sollten die Mitarbeiter mehr Fortbildung bekommen?*
- Kein Simulationsmodell kann (derzeit) all diese Fragen beantworten

- Dazu werden bei einem Modell projekt- und herstellerspezifische Parameter gesetzt
- Geeignet für Projektstart und Neuplanung
 - *Abschätzen der Kosten*
 - *Abschätzen der Produktqualität*
 - *Voraussichtliche Aufwendungen über die Zeit*
 - *Risikoanalyse*
 - *Anwenden verschiedener Entwicklungsprozesse*

- Durch stetige Anpassung der Parameter während der Projektlaufzeit
- Ohne umfangreiche Messung
 - *Vermutlicher Projektfortschritt*
 - *Aktuelle Aufwandsabschätzung*
 - *Voraussichtlicher Auslieferungstermin*
 - *Voraussichtliche Defektanzahl*

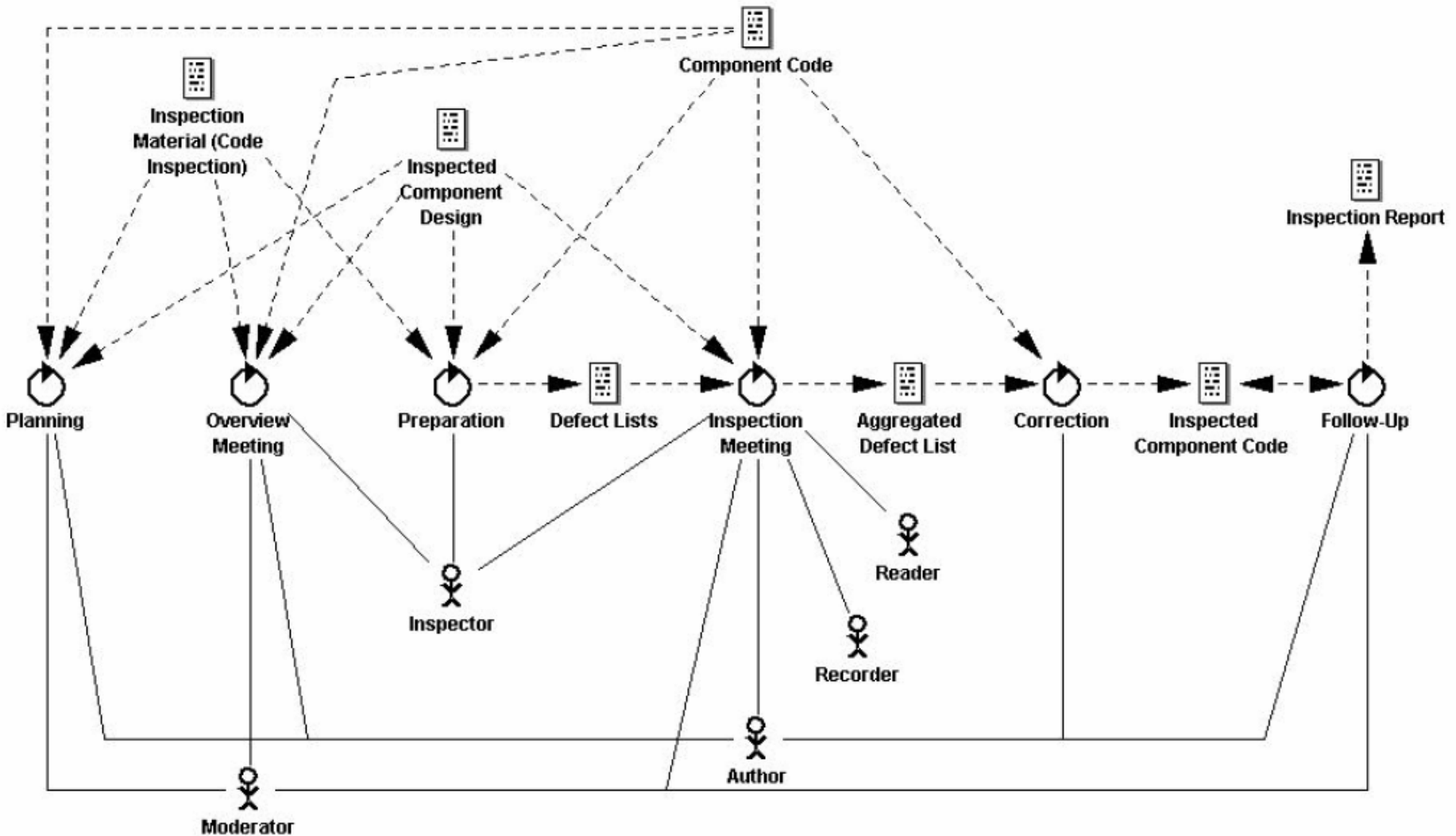
- Simulieren möglicher Anpassungsalternativen
- Ebenfalls durch stetige Anpassung der Parameter während der Projektlaufzeit
- Beurteilung anhand der Reaktionen
 - *Einsatz weiterer Mitarbeiter bestimmter Qualifikation*
 - *Änderung der Reihenfolge von Tätigkeiten*
 - *Einsatz qualitätssichernder Maßnahmen*
 - *Beispiel später*

- Nebeneffekt: Man „versteht“ das eigene Projekt
- ... oder deckt unverstandene Gebiete auf
- Visualisierungen helfen dabei
- Mittel der Kommunikation
 - *Nutzen von Inspektionen*
 - *Nutzen eines frühen Testens*
 - *Einsatz formaler Mittel*

- Softwaretechnikausbildung
 - Theoretisch: zu abstrakt, Auswendig lernen
 - Praktisch: zu unrealistisch oder zu zeitaufwändig
- ⇒ simulierte Praxis!
- Lehrhafter Vergleich zwischen Vermutung und „Realität“
 - Lernen (auch) aus Fehlern
 - Projekte beobachten
 - selbst Fehler machen dürfen (auch: Radikal sein!)
 - Macht Spaß
 - siehe SimCity, Die Sims, Flugsimulatoren,...
 - Zielgruppe meist: Projektleiter
 - aber keine Personalführung, Kundenkontakt,...
 - Oftmals vereinfachte (Lehr-)Modelle gewünscht

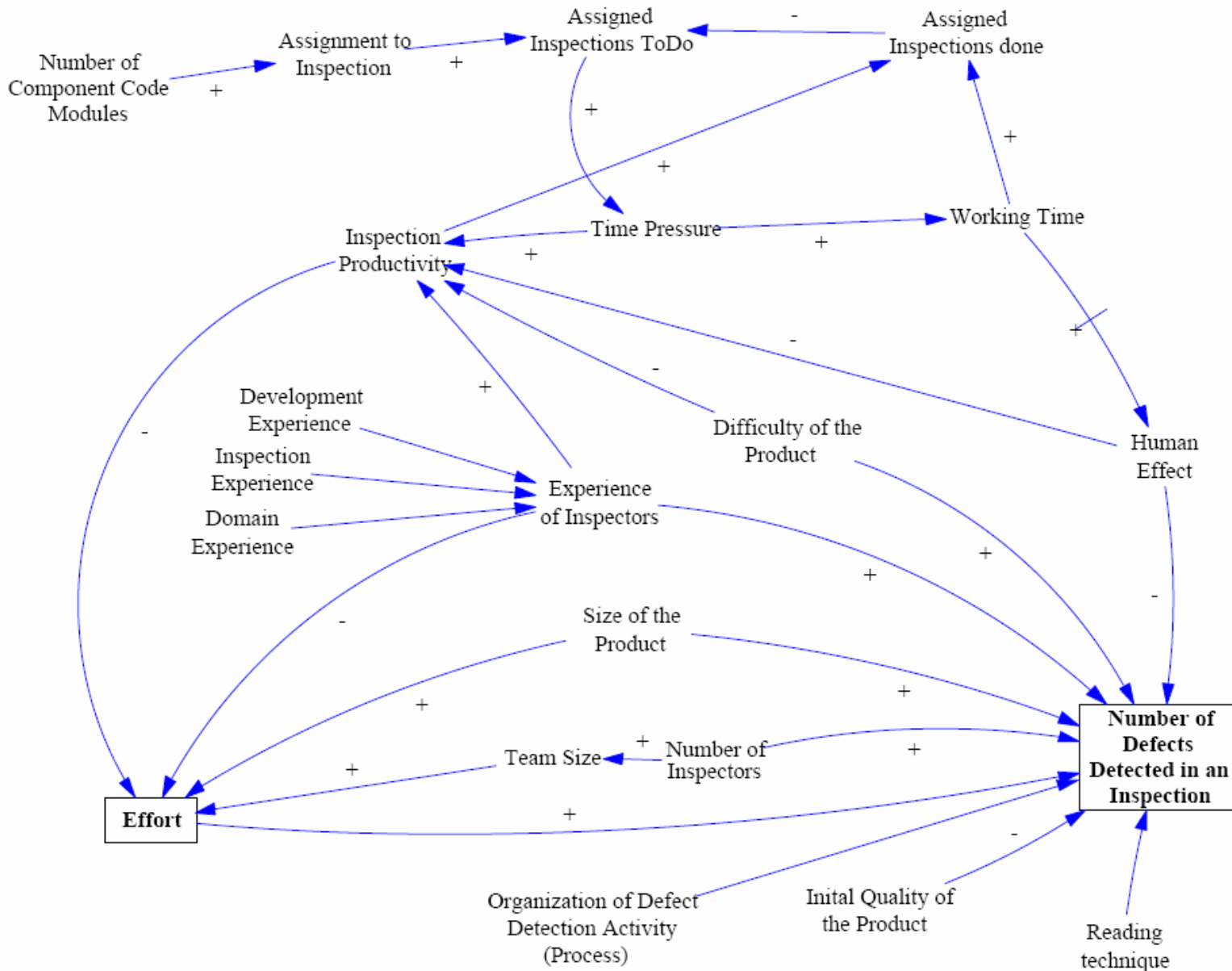
Hilfe bei der Ausbildung: Beispiel Inspektionen

Einsatzgebiete (6/6)



Hilfe bei der Ausbildung: Beispiel Inspektionen

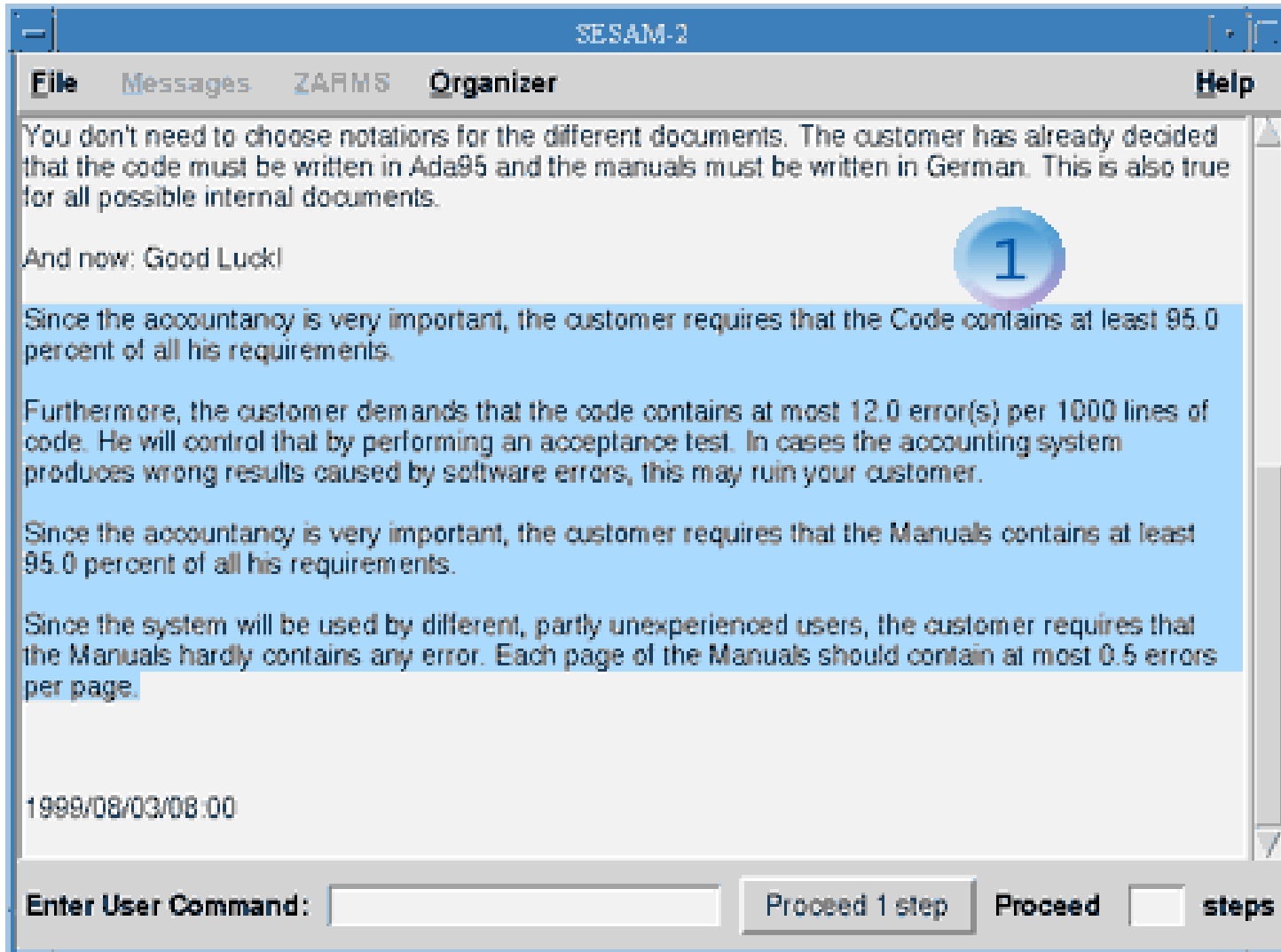
Einsatzgebiete (6/6)



- Uni Stuttgart, Lehrstuhl Ludewig (Anke Drappa, Diss.)
- Thema: Einfluss des Projektmanagements auf die Qualität eines Softwareprodukts
- Ähnlich einem Textadventure
 - Spieler ist der Projektleiter
 - Simulationszeit in Tagesschritten
 - Tägliche Entscheidungen möglich
- Beispiel folgt...

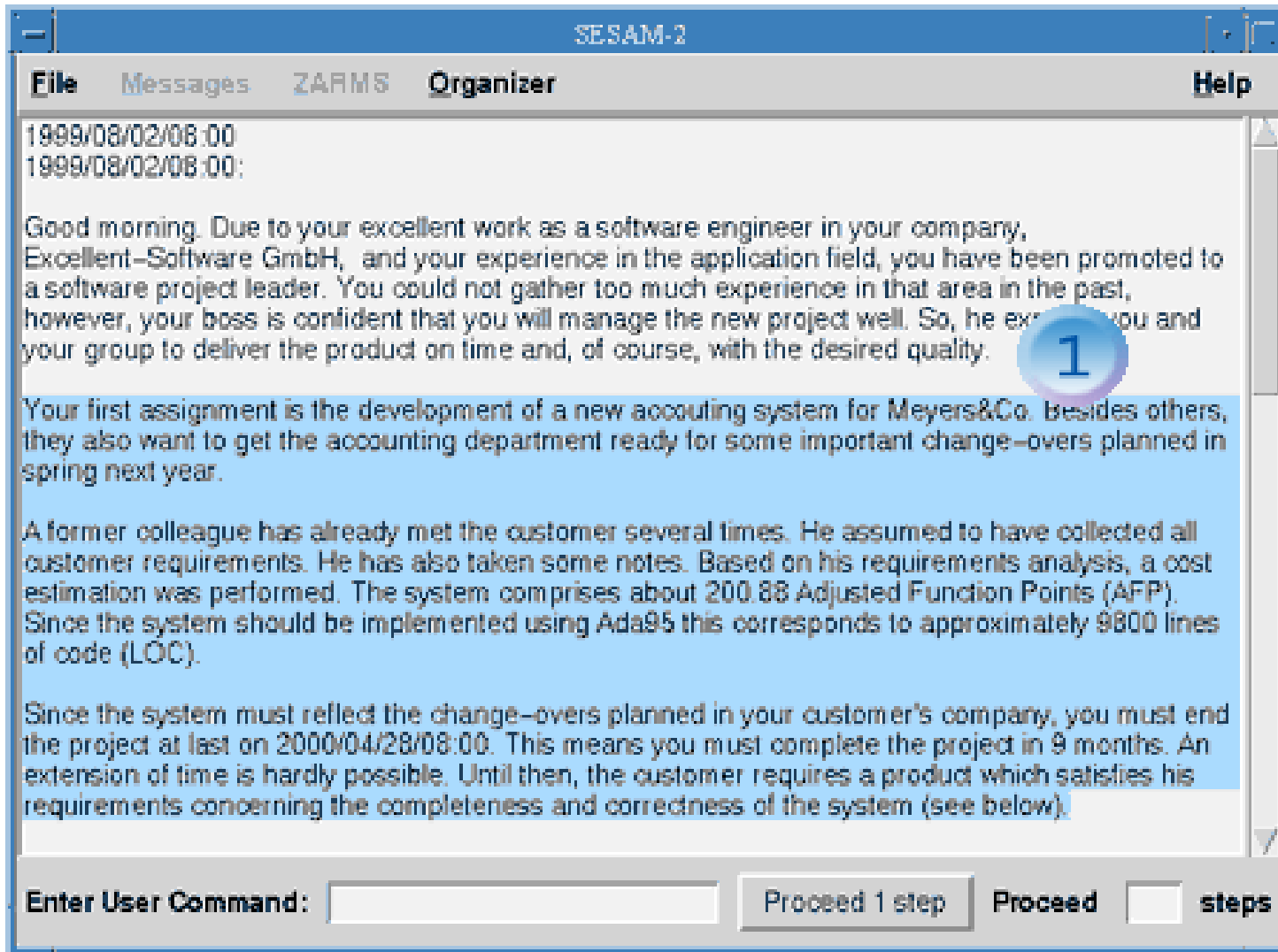
SESAM – Beispiel (1/8)

Ausbildung (2/13)



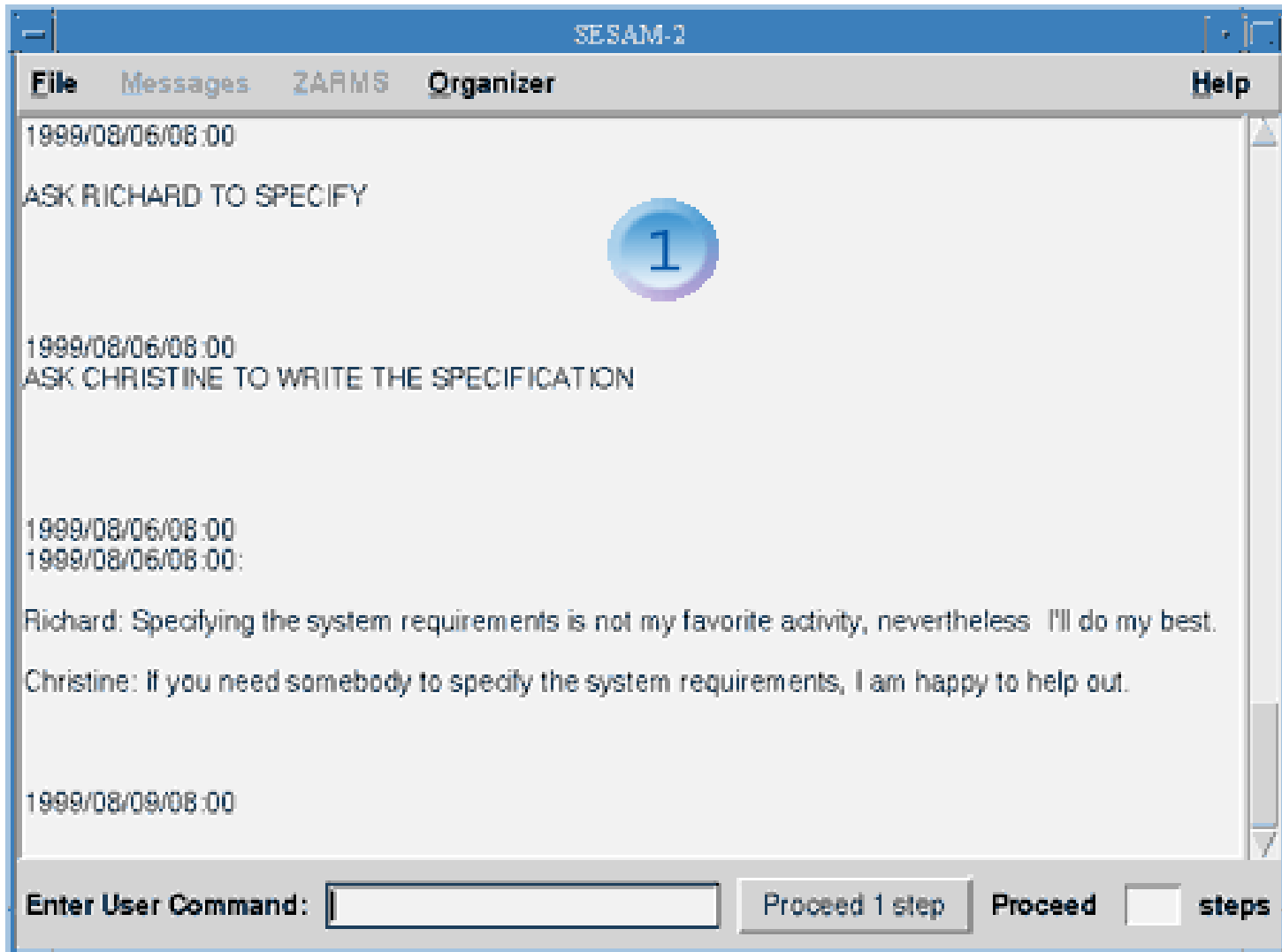
SESAM – Beispiel (2/8)

Ausbildung (2/13)



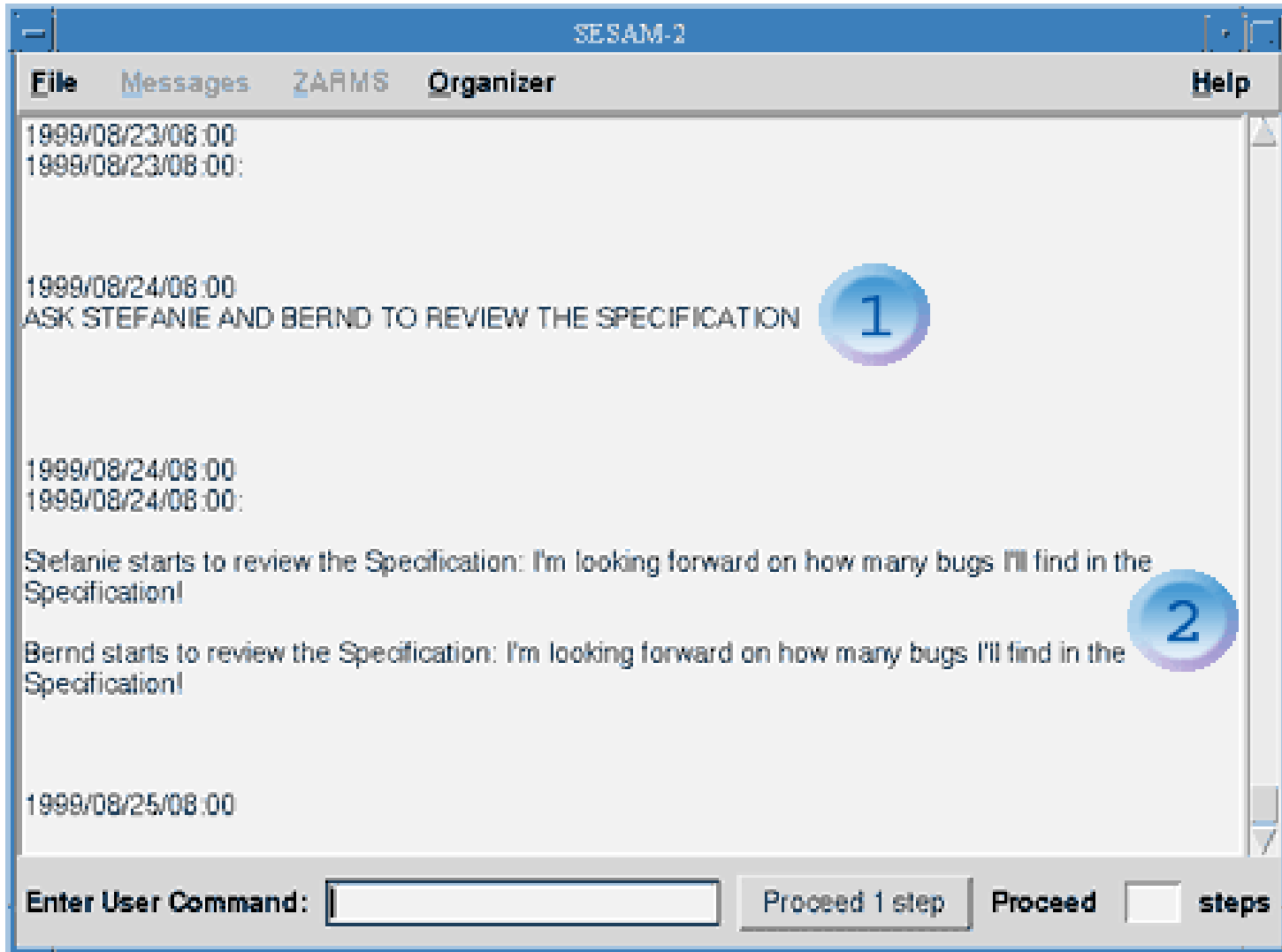
SESAM – Beispiel (3/8)

Ausbildung (2/13)



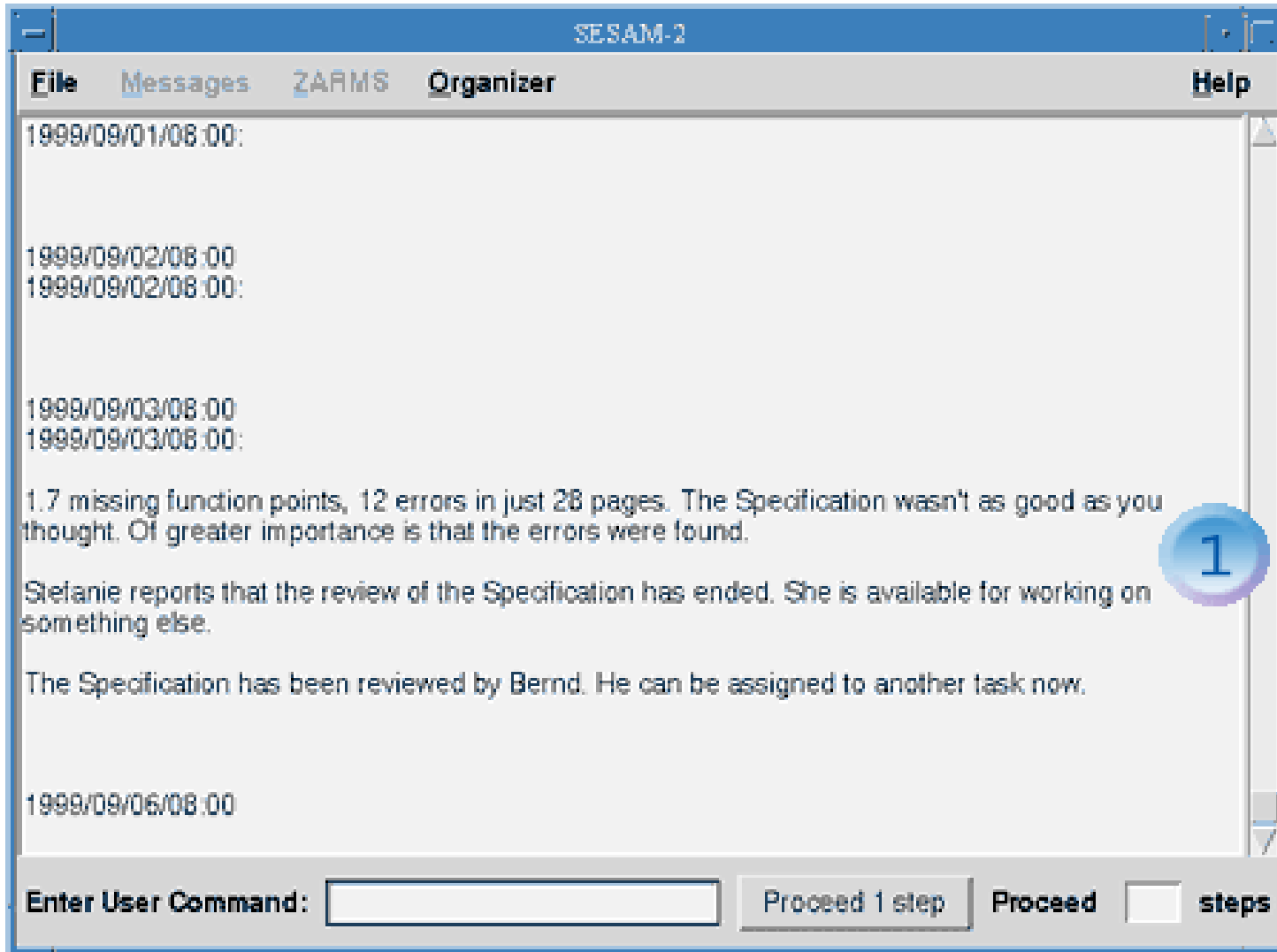
SESAM – Beispiel (4/8)

Ausbildung (2/13)



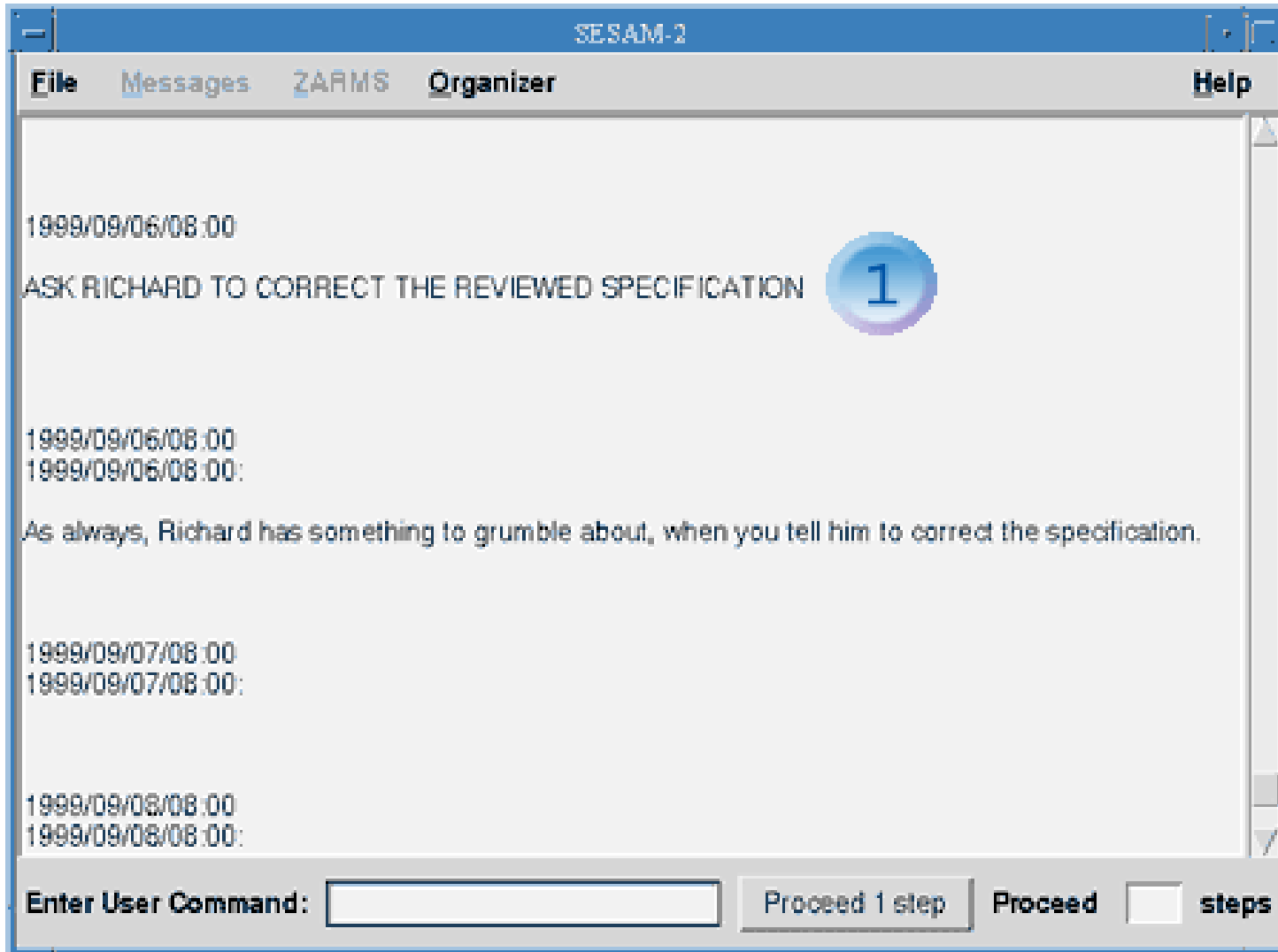
SESAM – Beispiel (5/8)

Ausbildung (2/13)



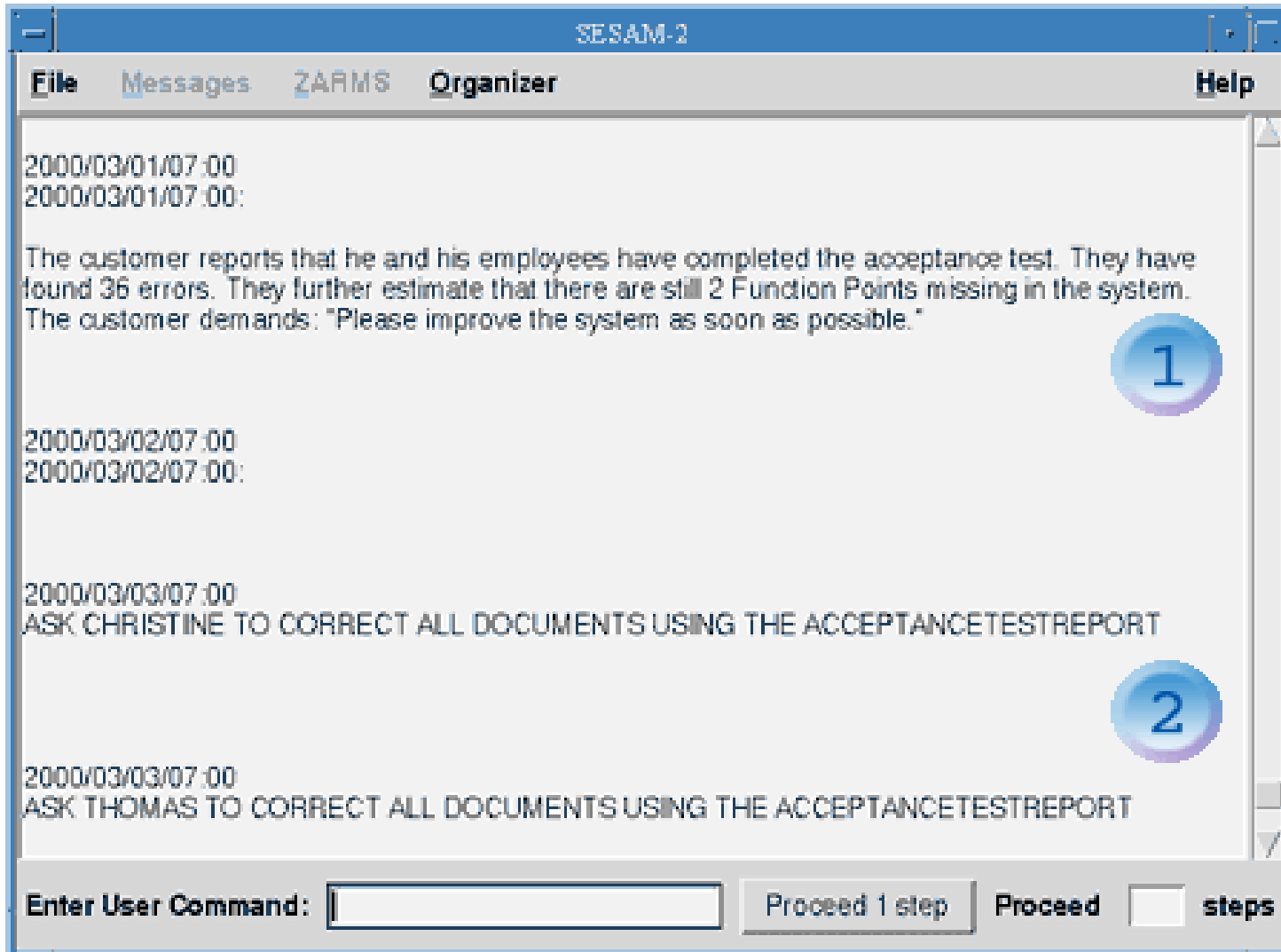
SESAM – Beispiel (6/8)

Ausbildung (2/13)



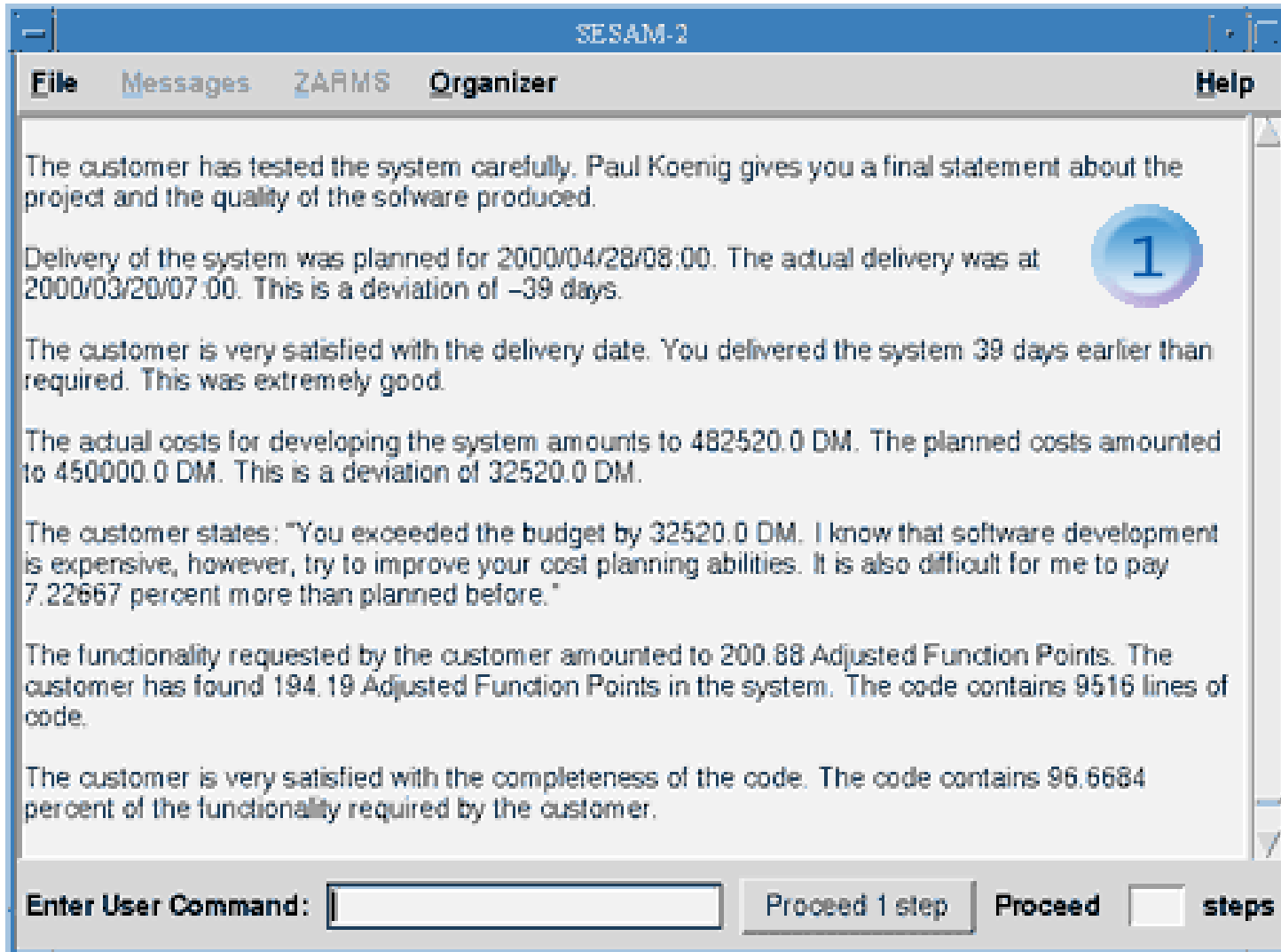
SESAM – Beispiel (7/8)

Ausbildung (2/13)



SESAM – Beispiel (8/8)

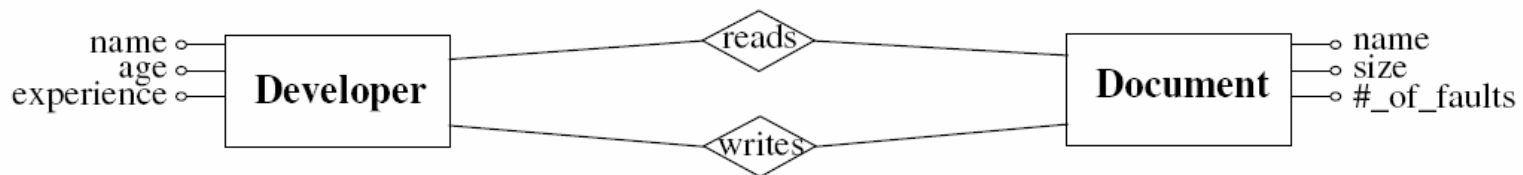
Ausbildung (2/13)



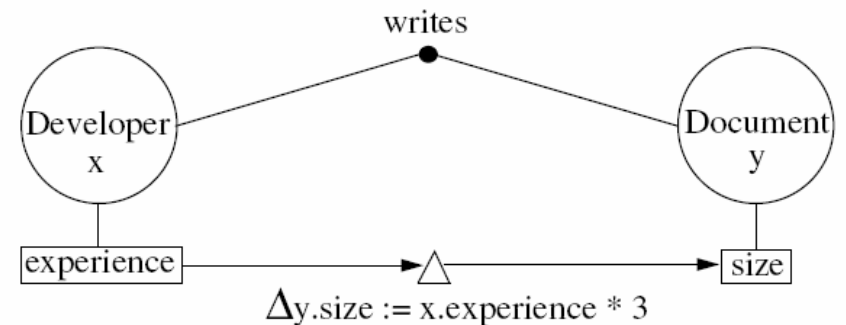
SESAM: Was bildet es ab?

Ausbildung (3/13)

- Hauptthemen sind qualitätssichernde Maßnahmen
 - Dokument X wird von Y erstellt und von Z geprüft
 - Auswirkungen oft erst am Projektende
 - Basierend auf Literatur und empirische Studien, 400 Regeln
- Auftrag vom Kunden ist Ausgangssituation
- Spieler (Projektleiter) ist Chef
 - Es wird genau das getan, was er sagt
 - Einstellungen und Entlassungen



- Kenntnisse und Erfahrungen der Entwickler
 - Methoden, Notationen und Aufgaben
 - Kosten der Entwickler
 - daraus entsteht u.a. Produktivität und Fehlerrate
- Teamleistung
 - Anzahl Mitarbeiter
 - Zeitdruck
- Qualitätssicherung
 - Fähigkeiten der Arbeiter
 - Zustand der Dokumente
- Fehlerbehebung
 - je später, desto teurer



- hire / fire {name}
- ask for information about {name}
- specify {name}
- inspect specification {name}
- correct specification {name}
- create/inspect/correct system design {name}
- integrate system {name}
- make integration test {name}
- ask customer to do the acceptance test
- quit current task of {name}
- inspect job of {name}

Ablauf

Activities	A	B	C	D	#Dev.
Analyze	x	x	x	x	1
Specify	x	x	x		2
Rev. of Specification & Corr.	x		x		2
Write User Manual	x	x	x		2
Rev. of User Manual & Corr.	x		x		2
Design System	x	x	x		3
Rev. of System Design & Corr.	x		x		3
Specify Modules	x	x	x		4
Rev. of Detailed Design & Corr.	x		x		4
Write Code	x	x	x	2x	4
Module Test & Corr.	x	x		2x	4
Integration	x	x	x	x	2
Integration Test & Corr.	x	x			2
System Test & Corr.	x	x		x	2
Acceptance Test & Corr.	x	x	x	x	2

Vier simulierte Prozesse A, B, C, D

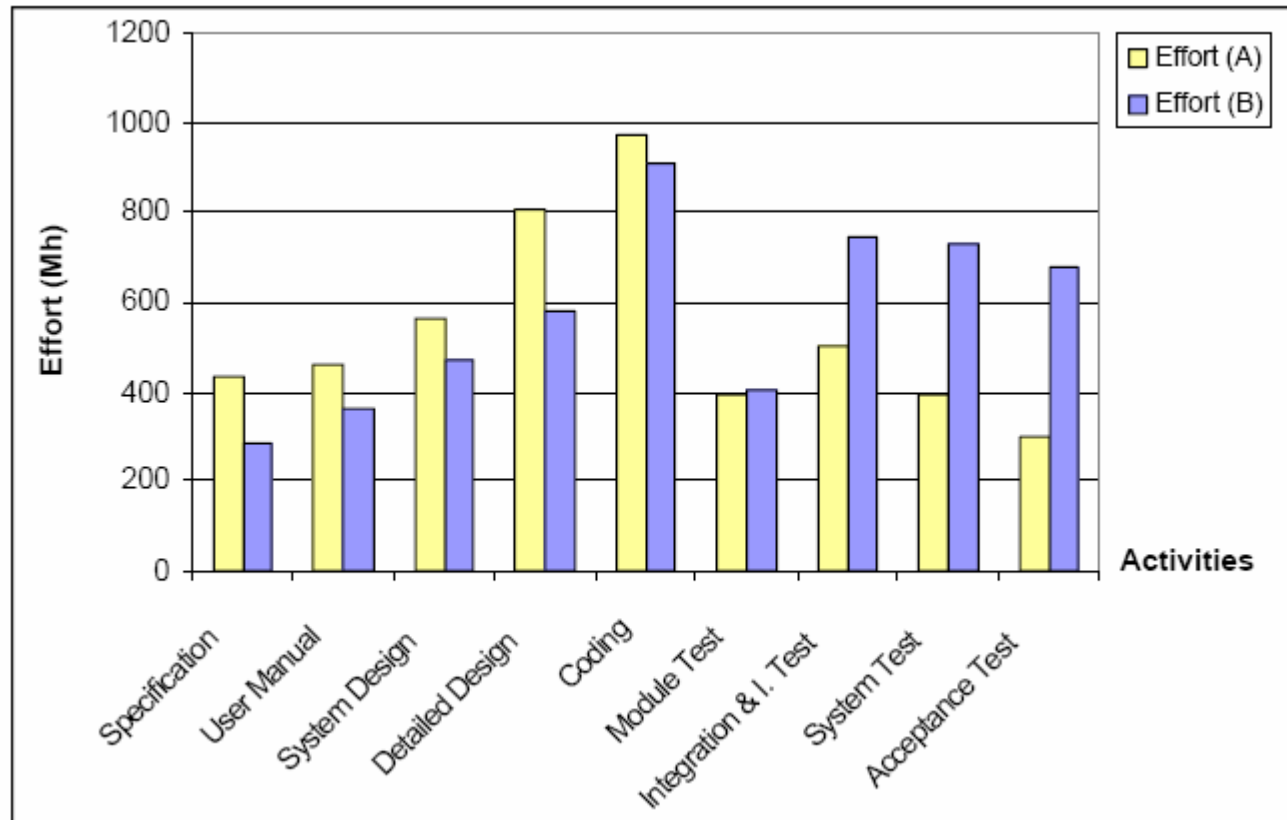
Ergebnis

Use of ...	Effort (Mh)	Completeness	Defect Density
Process A	4832.60	96.42 %	10.42
Process B	5165.22	93.38 %	19.78
Process C	4147.16	94.63 %	18.39
Process D	3165.41	79.39 %	53.85

SESAM: Beispiel Modellevaluation

Ausbildung (7/13)

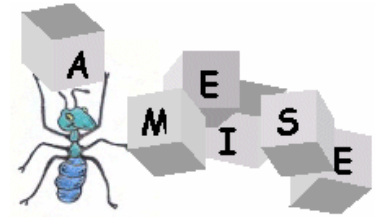
Aufwand von B (keine Reviews) ist höher als A (mit Reviews). Warum?





ask Richard to specify

- Entwickelt in Klagenfurt/Kärnten
- AMEISE baut auf SESAM auf
 - Neue Oberfläche
 - Erklärungs- und Hilfekomponente
- Neu: „Erklärungs“komponente
 - Kennzahlen für die aktuelle Situation
 - Anzahl gefundener Defekte in Relation zu Inspektionsaufwand
 - Anzahl gefundener Defekte pro Phase
- Neu: Hilfekomponente
 - Ratschläge für die aktuelle Situation, z.B.
 - „Achtung! Du hast zu wenig Funktionspunktdeckung!“
 - „Sehr gut! Du bist im Zeitrahmen.“
- Fremd-Server-basiert
 - Installation von SESAM eh schwer (Ada mit Java-Client)



AMEISE Client

Spiel Optionen Hilfe

SimulationPlus Simulation Treeview EK-Einheit Profil

Kommando wurde akzeptiert.
Told Christine to work on the module design.

Kommando wurde akzeptiert.
Told Richard to work on the system design.

1 Schritt(e) im Projekt nach vor.

Kommando wurde akzeptiert.
2004/01/22/08:00:
Richard: You finally recognized my talents. I am the best you could find but she won't object your decision and starts to specify the module design.

Befehlseingabe

code

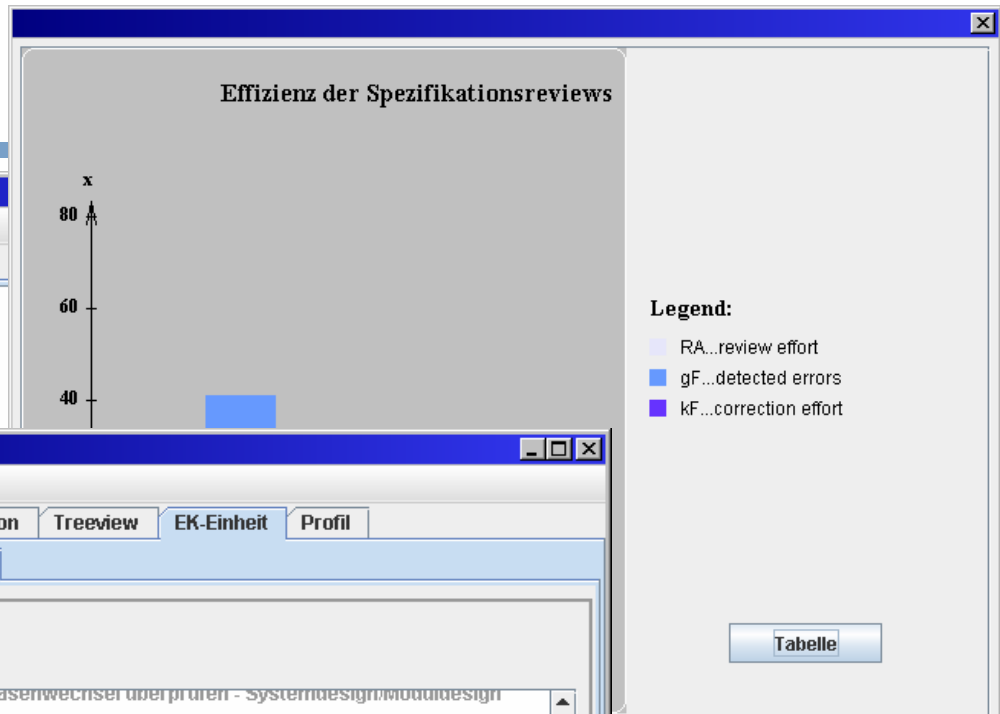
- correct all documents
- correct documentation
- correct module design
- correct reviewed code
- correct specification
- correct system design
- correct tested code
- customer do acceptance test
- customer participate in review documentation
- customer participate in review specification
- deliver system
- determine spent resources

Schritt: Simulationszeit

Aktueller Befehl

Serverstatus:

Warte auf Benutzereingabe



AMEISE Client

Spiel Optionen Hilfe

SimulationPlus Simulation Treeview EK-Einheit Profil

Auswertung Ratgeber

Fragenkatalog

Ratschläge holen

- Referenzdokument vor Phasenwechsel überprüfen - Systemdesign/Moduldesign
- Referenzdokument vor Phasenwechsel überprüfen - Moduldesign/Code
- Referenzdokument vor Phasenwechsel korrigieren - Spezifikation/Entwurf
- Referenzdokument vor Phasenwechsel korrigieren - Systemdesign/Moduldesign
- Referenzdokument vor Phasenwechsel korrigieren - Moduldesign/Code
- nach Tests nicht auf Korrektur vergessen - Modultest/Integrationstest
- nach Tests nicht auf Korrektur vergessen - Integrationstest/Systemtest
- nach Tests nicht auf Korrektur vergessen - Systemtest/Abnahmetest
- unvollständige Dokumente erneut überarbeiten - Spezifikation**
- unvollständige Dokumente erneut überarbeiten - Systemdesign
- unvollständige Dokumente erneut überarbeiten - Moduldesign
- unvollständige Dokumente erneut überarbeiten - Code
- unvollständige Dokumente erneut überarbeiten - Handbuch

Auswahl aufheben unvollständige Dokumente erneut überarbeiten Ratschlag anzeigen

Ratschlag

Ihre Spezifikation enthält noch zu wenige AFPs. Wenn sie dieses Dokument in der nächsten Phase einsetzen, werden viele Fehler übernommen. Je früher Sie die Fehler verbessern, desto geringer sind Aufwand und Kosten. Eine erneute Überarbeitung der Spezifikation wäre deshalb ratsam.

Warte auf Benutzereingabe

user_1 ist angemeldet

und für einen Review gerechtfertigt war, das Review besser verzichtet hätte.

Warum

Zurück

- Studenten planen ihre Aktionen nicht. Die meisten enden in einem Code & Fix – Verhalten
- Ohne Erklärung der Ursachen für schlechte Ergebnisse machen die Studenten immer wieder die gleichen Fehler
- Die Simulation ist noch zu abstrakt
 - Zusammenhänge nicht visualisiert
 - Problem taucht erst bei der „Abrechnung“ auf
 - Kein Flugsimulatorgefühl
- Kann ich bestätigen!
- Entspricht das Modell der Wirklichkeit?
 - Oder will es uns missionieren?
- Hang zum „Mal schauen, wie das System reagiert“

Andere Ansätze: GenSim (Generic Simulator)

Ausbildung (12/13)

- Fraunhofer IESE, Kaiserslautern (Dietmar Pfahl, Diss.)
- Thema ist Zeitplanung der Phasen Entwurf, Impl., Test
 - Basiert auf COCOMO
- „Ein Schritt“-Simulation
 - Stellschrauben Qualität, Größe, Mitarbeiter
 - Ergebnis: Projektablauf als Kurve

Simulation - Input - General

Manpower

Planned Manpower: 16

Manpower Adjustment: 0

Manpower Adjustment Time: 0

Max Manpower Limit: 0

Manpower Automatic Adjustment Time Flag: 0

Schedule

Planned Completion Time: 0

Quality

Planned Design Inspection Practice: 0.00

Planned Impl. Inspection Practice: 0.00

Goal Field Defect Density: 1.2

Size

Initial Job Size In Tasks: 1000

Job Size Adjustment: 0

Job Size Adjustment Time: 0

Complexity & Skill

Project Complexity: 3.0

Manpower Skill: 1.0

Input

General

Advanced

Output

Analysis

Simulate

Reset All

Breaks at:

Simulation - Output - General

Size:	1000	Design:	Implementation:	Test:
Manpower:	16.00	Size:	Size:	
Field Defect Density:	1.202	1000	1000	
Time:	506	Size inspected:	Size inspected:	Size tested:
Total Effort:	8080	0	0	999
		Effort:	Effort:	Effort:
		2432	1376	4272
		Time:	Time:	Time:
		152	238	505

Graph Data:

Y-axis: 0, 750, 1,500

X-axis: 0, 125, 250, 375, 500 (Time in Days)

Legend:

- requirements: Current (Red line)
- requirements: Default (Green line)
- tasks for impl: Current (Blue line)
- tasks for impl: Default (Black line)
- tasks for test: Current (Red line)
- tasks for test: Default (Black line)

Input

Output

General

Advanced

Internal

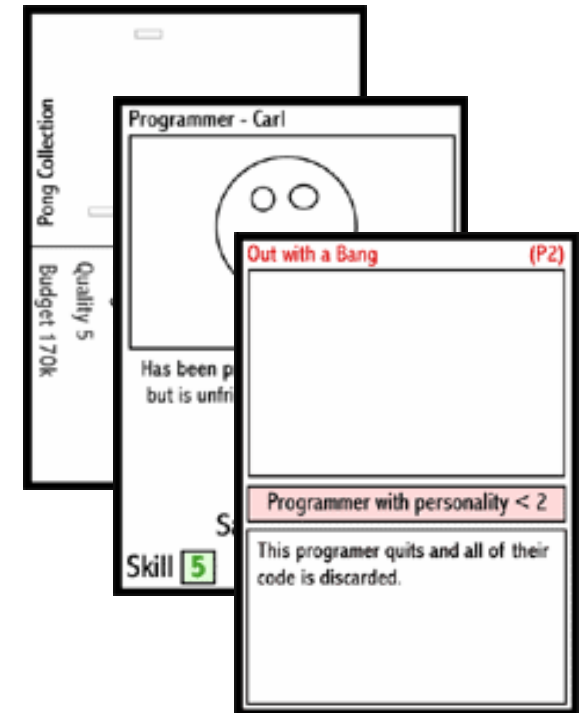
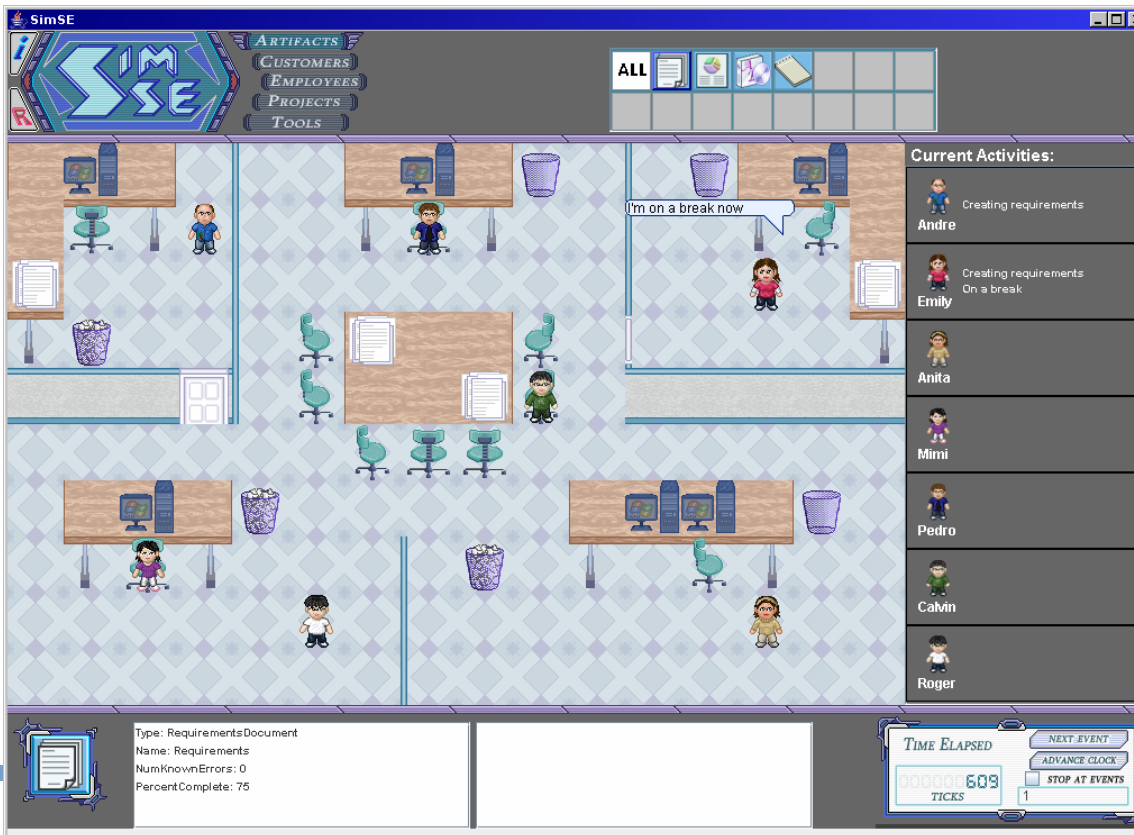
Analysis

Simulate

Reset All

Breaks at:

- University of California, Irvine
- SimCity – mäßig
 - Zuweisung von Aufgaben zu diversen Mitarbeitern
 - Zwischendurch neue Anforderungen, Krankheiten, etc.



1. *Modellierungsbeispiel: Mr. Tompkins' Personalproblem*
2. *Einführung: Was ist das? Was soll das?*
3. *Pionierarbeit: Abdel-Hamid und andere*
4. *Einsatzgebiete: Chancen durch Simulation*
5. *Ausbildung: SESAM und AMEISE*
6. **Systematik: Was für SWT-Simulationen gibt es?**
7. *Probleme: Warum macht es nicht jeder?*

- Viele Verfeinerungen der Pionierarbeit von Abdel-Hamid
 - fuzzy logic u.a. Modellierungs- und Simulationstechniken
 - Anleitung zur Anpassung an spezielle Projekte
 - Einbindung in CMM 2/3-Firmen
- Viele andere Modelle
 - wurden schon genannt
- Aber (meines Wissen) keine „Explosion“ von Arbeiten
 - steter Zufluss
- Es folgt ein Versuch der Systematik...

Organizational Breadth	Time Span		
	Short	Medium	Long
< 1 product / project	portion of life cycle	portion of life cycle	
= 1 product / project	portion of life cycle	development project	long-term product evolution
> 1 product / project		multiple, concurrent projects	long-term organization

- Aktivitäten und Arbeiten
- Objekte (Codemodule, Entwürfe, Fehlerberichte)
- Ressourcen (Mitarbeiter, Budget)
- Arbeits- und Dokumentenfluss
- Abhängigkeiten zwischen Arbeiten
- Entscheidungspunkte

- und anderes

- Aufwand und Kosten
- Defektanzahl
- Arbeitsaufwand über die Zeit
- Return on investment

- und anderes

- Softwaregröße
- Defekteinbauraten
- Defektfindungsraten
- Aufwand für bestimmte Arbeiten
- Mitarbeiterdaten (Fähigkeiten, Motivation)

- und anderes

Einsatzprobleme: Warum geht's nicht weiter?

- Liegt es daran, dass das Modellieren von Menschen schwer ist?
- Liegt es daran, dass die Menge der Variablen riesig ist?
- Liegt es daran, dass es keine „Physik“ der Softwareentwicklung gibt?
- Liegt es daran, dass z.B. 270 Gleichungen nicht in den Griff zu bekommen sind?
 - Änderungen werden schwer
 - Spezialisten nötig
- Stimmen die Modelle nicht? (Wie findet man das heraus?)
- Große Abhängigkeit von Projekt und Hersteller
- Schwierige Kalibrierung

Das war's (und hier ist die Literatur)

- Tom DeMarco. *Der Termin* (The deadline), 1997
- Dan Houston. *System Dynamics Modeling and Simulation of Software Development: A Tutorial*, Un-Published, Arizona State University, 1996
- A. Drappa, J. Ludewig. *Simulation in software engineering training*. ICSE 2000, Limerick, Ireland
- M. Kellner, R.J. Madachy, and D.M. Raffo. *Software Process Modeling and Simulation: Why? What? How?* Journal of Systems and Software, Vol. 46, No. 2/3, 1999

Fragen, Anmerkungen, Meinungen?