

Software-Engineering-Simulation als Brücke zwischen Vorlesung und Praktikum

Tilman Hampp · Stefan Opferkuch

Abteilung Software Engineering, Institut für Softwaretechnologie, Universität Stuttgart
Universitätsstraße 38, 70569 Stuttgart
{hampp|opferkuch}@informatik.uni-stuttgart.de

Zusammenfassung

In der Ausbildung angehender Projektleiter müssen nicht nur theoretische Kenntnisse vermittelt werden, sondern auch deren praktische Anwendung. Die Kombination aus Vorlesung und Praktikum vermittelt die Theorie und verlangt ihre Umsetzung. Sie erlaubt es aber nicht, die Zusammenhänge zwischen den theoretischen Grundlagen und der Umsetzung nachzuvollziehen. Diese Schwäche lässt sich ausgleichen, indem während der Ausbildung eine Projektsimulation mit detaillierter Auswertung durchgeführt wird. In diesem Beitrag zeigen wir, wie die simulierten Projekte ausgewertet werden, stellen Beispielauswertungen vor und diskutieren den dadurch erzielten Lernerfolg.

1 Einleitung und Motivation

Der Erfolg eines Softwareentwicklungsprojekts wird maßgeblich durch die Projektleitung bestimmt. Empirische Untersuchungen [Lic98] [Aho03] zeigen typische Fehler von (angehenden) Projektleitern: Die Planung ist unzureichend, Qualitätssicherung und Fortschrittskontrolle werden vernachlässigt, Termine und Budget überschritten. Daraus ergibt sich der Bedarf für eine Verbesserung der Projektleiter-Ausbildung.

1.1 Projektleiter-Ausbildung durch Vorlesungen

Die für die Leitung eines Softwareprojekts erforderlichen theoretischen Kenntnisse lassen sich durch Vorlesungen vermitteln. Vorteilhaft dabei ist, dass in Vorlesungen einer großen Anzahl von Teilnehmern Inhalte vermittelt werden können und sich die Vorlesungen jedes Semester einfach wiederholen lassen. Für die Projektleiter-Ausbildung haben diese konventionellen Lehrveranstaltungen jedoch Nachteile: Die praktische Anwendung der Inhalte muss geübt werden. Die Studierenden haben eine unrealistische Vorstellung von der Projektarbeit, weil diese durch Programmieren im Kleinen geprägt ist und nicht reale Softwareprojekte widerspiegelt.

1.2 Projektleiter-Ausbildung durch praktische Lehrveranstaltungen

Praktische Projekte in der Ausbildung ermöglichen es dagegen, theoretische Kenntnisse anzuwenden. Dazu muss die Problemstellung realistisch sein: Die Projekte müssen so umfangreich sein, dass die Arbeit nur im Team geleistet werden kann. Planung, Qualitätssicherung, Projektleitung und Organisation dürfen nicht vorgegeben werden, damit die Teilnehmer die typischen Schwierigkeiten kennenlernen und dadurch eine deutlichere und realistischere Vorstellung von Softwareprojekten und ihren Problemen gewinnen. Dabei ist nachteilig, dass nur ein Teilnehmer die Rolle des Projektleiters übernehmen kann. Eine Rotation der Rolle des Projektleiters zwischen den Teilnehmern während des Projekts hat sich nicht bewährt.

Der Dozent steht aufgrund der komplexen Zusammenhänge in Projekten vor Problemen: Er kann den Erfolg nicht selbst herbeiführen oder erzwingen, sondern höchstens beratend eingreifen. Nachträglich kann das Vorgehen kaum analysiert werden, weil sich der Zusammenhang zum Projektergebnis nicht nachvollziehen lässt und durch andere Einflüsse überlagert wird. Für eine Analyse wäre eine Art gläsernes Projekt notwendig, so dass der Dozent die einzelnen Handlungen des Projektleiters und ihre Auswirkungen identifizieren kann. Selbst wenn Fehler im Vorgehen erkannt werden, kann das Projekt nicht wiederholt werden. Die Teilnehmer haben keine Möglichkeit, diese Fehler zu vermeiden und zu sehen, wie sich die Ergebnisse verbessern.

1.3 Projektleiter-Ausbildung durch Simulation

Die Nachteile der praktischen Lehrveranstaltungen versuchen wir an der Universität Stuttgart im Diplomstudiengang Softwaretechnik [Lud99] durch die Simulation von Softwareprojekten mit dem SESAM-System (Software Engineering durch Animierte Modelle) auszugleichen [Lud94]. Simuliert werden Softwareprojekte von der Analyse bis zur Übergabe an den Kunden mit allen für die Aus-

bildung relevanten Aspekten. Ausnahme ist der Projektleiter, der nicht simuliert wird, sondern das Projekt am Simulator steuert und kontrolliert. Modelliert werden kleine und mittelgroße Projekte, die ein Kunde als Auftrag für ein Informationssystem vergibt [Dra98].

Die Simulation bietet den wichtigsten Vorteil der praktischen Arbeit, indem sie es ermöglicht, die Theorie anzuwenden. Sie vermeidet die Nachteile praktischer Lehrveranstaltungen, weil jeder Teilnehmer sein eigenes Projekt leitet. Der Dozent kann die Stärken und Schwächen im Vorgehen des Projektleiters analysieren und die Auswirkungen auf das Projektergebnis zeigen. Diese Analyse basiert auf den im Simulationsmodell enthaltenen Zusammenhängen in Softwareprojekten.

Damit Simulationen mit SESAM auf die Realität übertragbar sind, müssen die enthaltenen Zusammenhänge valide sein. Sie sind durch Literatur belegt, die Quantifizierung erfolgte mit empirischen Daten, vorwiegend aus [Jon96]. Weil kaum andere zuverlässige empirische Daten veröffentlicht wurden, ist das Modell auch gegen das Kostenschätzmodell Cocomo II [Boe00] validiert worden. Die Plausibilität wurde durch Vergleich von Simulationsergebnissen mit unterschiedlichem Vorgehen geprüft.

2 Schulungen mit dem SESAM-System

[Dra98] untersuchte, ob sich Studierende durch die Leitung simulierter Projekte verbessern. Dabei zeigte sich, dass die Zusammenhänge zwischen dem Vorgehen des Projektleiters, den Effekten des Modells und den Projektergebnissen zu komplex sind, als dass man ohne Analyse ausschließlich aus der Projektdurchführung lernen könnte. Darum wird die Simulation in ein Schulungskonzept eingebettet [Man01]. Dazu gehört eine Einführungsveranstaltung, in der SESAM und seine Bedienung vorgestellt werden. Anschließend führen die Teilnehmer ein simuliertes Projekt durch, das der Dozent auswertet. In der Analyseveranstaltung werden Probleme im Vorgehen anhand dieser Auswertung mit den Teilnehmern durchgesprochen und diskutiert. Danach wiederholen die Teilnehmer das Projekt mit den gleichen Vorgaben und können ihre Fehler aus dem ersten Spiel vermeiden oder alternative Lösungsstrategien ausprobieren. Sie sehen als Abschluss ihre Projektergebnisse und können daran Verbesserungen erkennen.

Mit diesem Konzept werden seit dem Wintersemester 2000/2001 Schulungen durchgeführt. Sie erfolgen für alle Studierenden des Studiengangs Softwaretechnik im dritten Semester nach der Vorlesung »Einführung in die Softwaretechnik 1« und vor den praktischen Lehrveranstaltungen [Lud06].

3 Analyse simulierter Projekte

Für den Lernerfolg der Teilnehmer ist die Analyse der simulierten Projekte ausschlaggebend, weil sie dem Dozenten erlaubt, konkrete Hilfestellung für Verbesserungen im Projekt zu geben. Für die Analyse werden vom Simulator gespeicherte Protokolldateien mit zwei Werkzeugen ausgewertet. Das Werkzeug Sesamalyzer analysiert den Projektzustand über die simulierte Zeit, das Werkzeug Sesamscore ermöglicht die Auswertung der Interaktion zwischen Studierenden und Simulator.

Zu Beginn der Analyse prüft der Dozent, welche Zielvorgaben des Projekts jeder Studierende erreicht hat. Wenn die Zielvorgaben nicht eingehalten wurden, wird der Projektverlauf anhand eines Gantt-Diagramms betrachtet. Dies liefert einen ersten Anhaltspunkt, welche Probleme bei der Projektdurchführung aufgetreten sind. Ausgehend von diesen Erkenntnissen erarbeitet der Dozent Hypothesen über die Handlungen und Effekte, die zu dem vorliegenden Ergebnis geführt haben könnten. Durch weitere Analysen wird versucht, die Hypothesen zu widerlegen oder zu bestätigen. So kommt der Dozent zu einem Ergebnis, welche Handlungsweisen eines Studierenden sinnvoll oder problematisch waren. Die Ergebnisse aller Studierenden werden für die Analyseveranstaltung verdichtet, aufbereitet und in der Veranstaltung diskutiert. Die Studierenden sollen sich dabei aktiv an der Ursachenforschung beteiligen. Durch den Vergleich mit dem Vorgehen anderer Projektleiter lernen die Studierenden Alternativen zu den von ihnen getroffenen Maßnahmen und Entscheidungen kennen.

Wie eine Auswertung zu erarbeiten ist, wird in [Opf06] beschrieben. Im Folgenden werden drei Auswertungen realer Schulungen exemplarisch vorgestellt.

3.1 Erstes Auswertungsbeispiel

Im ersten Beispiel überschreitet der Studierende die Zielvorgaben bezüglich Dauer, Korrektheit und Vollständigkeit. Abbildung 1 zeigt das Gantt-Diagramm des Projekts.



Abb. 1 Gantt-Diagramm eines Projektverlaufs in SESAM

Neben den Tätigkeiten zur Erstellung der Spezifikation (*Spez* in *Abb.1*), des Entwurfs (*Entwurf*), der Modulspezifikation (*MSpez*), des Codes (*Code*) und des Handbuchs (*HB*) wurden mehrere Tests, nämlich Modultest (*MTest*), Integrationstest (*ITest*), Systemtest (*STest*) und Abnahmetest (*ATest*), durchgeführt. Keines der erstellten Artefakte wurde jedoch durch Reviews überprüft und korrigiert; diese wären sonst im Gantt-Diagramm als Aktivitäten zu sehen. Dadurch erklärt sich die Überschreitung der Zielvorgaben.

Dieser Fall deutet auf ein grundlegendes Missverständnis über die Bedeutung von Reviews hin. Offensichtlich ist es dem Studierenden nicht gelungen, sein Wissen über Reviews aus der Vorlesung in Handlungen im simulierten Projekt umzusetzen. Der Dozent muss in der Analyseveranstaltung nochmals die Bedeutung von Reviews herausstellen. Das nächste Beispiel zeigt dies anschaulich.

3.2 Zweites Auswertungsbeispiel

Im zweiten Beispiel werden die Spielverläufe zweier Studierender gegenübergestellt, um die Auswirkungen unterschiedlicher Handlungsweisen zu zeigen. In *Abbildung 2* sind die frühen Phasen in zwei Projekten über die Zeit dargestellt.

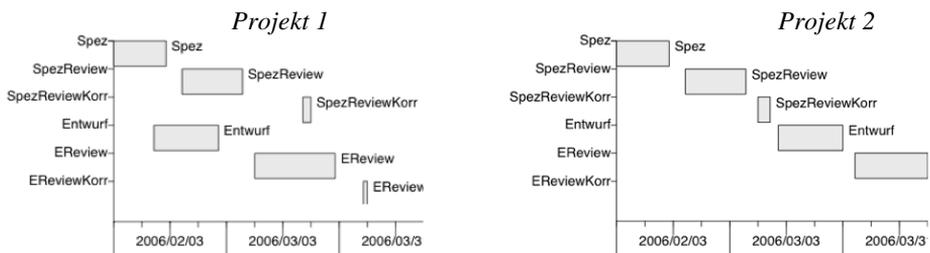


Abb. 2 Frühe Phasen in den simulierten Projekten zweier Studierender

In Projekt 1 wurde mit dem Entwurf direkt nach der Fertigstellung der Spezifikation begonnen. In Projekt 2 hingegen wurde die Spezifikation zunächst einem Review (*SpezReview*) unterzogen und dann korrigiert (*SpezReviewKorr*), bevor mit dem Entwurf begonnen wurde.

Die Auswirkungen dieser beiden unterschiedlichen Vorgehensweisen sind in *Abbildung 3* grafisch dargestellt. Da der Entwurf auf der Spezifikation basiert, werden unentdeckte Fehler von der Spezifikation in den Entwurf übertragen.

Abbildung 3 zeigt die im Entwurf enthaltenen Analysefehler für Projekt 1 und 2. Die Plateaus der beiden Kurven (1 und 2 in der *Abbildung*) entsprechen der maximalen Anzahl an Fehlern, die in den Entwürfen enthalten waren. Der Rückgang der Fehlerzahl im Laufe des Projekts (3 und 4 in der *Abbildung*) wird durch Entwurfsreviews erreicht.

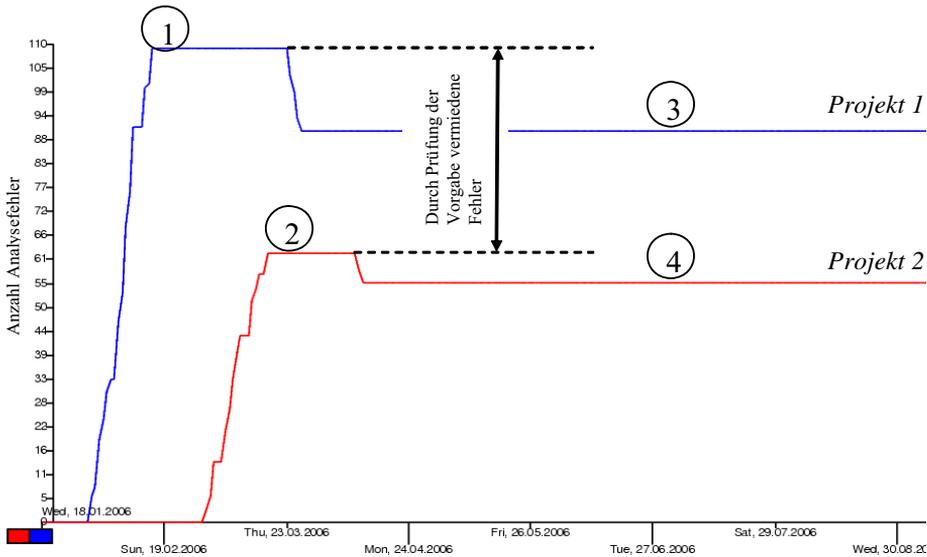


Abb. 3 Analysefehler im Systementwurf

Der Dozent sollte hier herausstellen, dass durch die Verwendung geprüfter Ergebnisse das Übertragen von Fehlern von einem Dokument zum nächsten deutlich reduziert werden kann. Zwar können Fehler auch durch spätere Reviews entdeckt und beseitigt werden – Abbildung 3 zeigt, dass dies in Projekt 1 sogar besser gelang als in Projekt 2, da der Rückgang der Fehleranzahl in Projekt 1 deutlich größer ist –, aber es gilt: Wenn Fehler erst gar nicht gemacht werden, entstehen auch keine Kosten für ihre Behebung.

3.3 Drittes Auswertungsbeispiel

Im dritten Beispiel hat der Studierende die Budgetvorgaben überschritten. In der Simulation sind wie in der Realität die Mitarbeiter der größte Kostenfaktor. In Abbildung 4 sind die im Projekt eingestellten Mitarbeiter über die Zeit dargestellt.

Die Einstellungsdauer ist durch den unteren durchgehenden Balken neben dem Namen des jeweiligen Mitarbeiters dargestellt. Der Balken beginnt beim Einstellungs- und endet beim Entlassungsdatum. Darüber sind durch mehrere Balken die Zeiten dargestellt, in denen dieser Mitarbeiter mit einer Aufgabe beschäftigt war. Die dunklen Balken zeigen die Zeiten, in denen der Mitarbeiter ohne Aufgabe war. Bei der Analyse muss beachtet werden, dass eine durchgängige Beschäftigung eines Mitarbeiters in SESAM nicht erreicht werden kann, weil es mit einer Simulationsschrittweite von einem Tag zu Leerlauf bei der Aufgabenverteilung an die Mitarbeiter kommen kann.

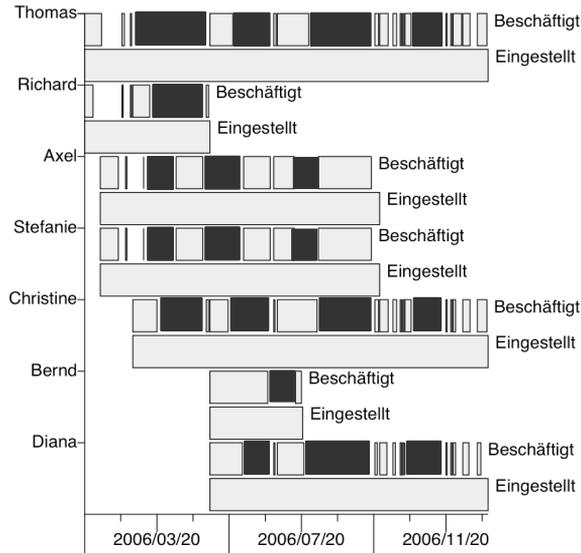


Abb. 4 Beschäftigte Mitarbeiter

Abbildung 4 zeigt jedoch Zeiträume von bis zu zwei Monaten, in denen ein Mitarbeiter zwar eingestellt war, er aber keine vom Projektleiter zugewiesene Aufgabe hatte. Dies lässt auf eine ungenügende Personalplanung schließen. Der Dozent muss betonen, dass eine detaillierte Personalplanung bereits vor Projektbeginn durchgeführt werden muss. Dabei muss der Projektleiter planen, welche Mitarbeiter in welchen Zeiträumen im Projekt für welche Aufgaben benötigt werden. Diese Planung muss natürlich im Projektverlauf überprüft und angepasst werden.

4 Lernerfolge in der Schulung mit SESAM

Der Lernerfolg zeigt sich deutlich in empirischen Untersuchungen [Opf02] und in den Ergebnissen, die die Schulungsteilnehmer in den simulierten Projekten erzielen.

4.1 Verbesserungen der Studierenden nach der Analyseveranstaltung

Die Verbesserungen der Projektergebnisse zeigen sich durch den Vergleich der erreichten Zielvorgaben zwischen erstem und zweitem Spiel. Abbildung 5 stellt für die Schulung im Wintersemester 2004/2005 dar, mit welcher Häufigkeit welche Anzahl der Zielvorgaben von den 60 Studierenden erreicht wurde. Im Spiel vor der Analyseveranstaltung haben 5 Studierende alle sechs Zielvorgaben erreicht. Das zweite Spiel haben 24 Studierende innerhalb aller Zielvorgaben beendet.

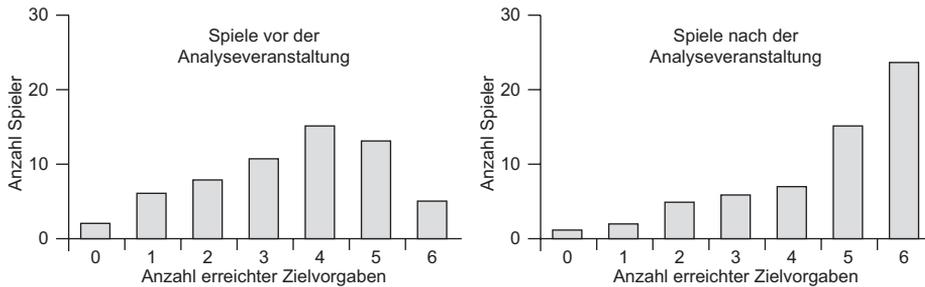


Abb. 5 Anzahl erreichter Zielvorgaben aller Studierenden im Wintersemester 04/05

In den Projekten vor der Analyseveranstaltung hat mehr als die Hälfte der Studierenden weder den Termin noch die Kosten eingehalten. Die Überschreitungen sind sehr deutlich: Ein Teilnehmer hat das vorhandene Budget um mehr als das Doppelte überzogen und den Termin um 5 Monate verfehlt. Nur ein Drittel konnte den Code mit für den Kunden zufriedenstellender Korrektheit ausliefern. Nach der Analyserunde konnten über zwei Drittel der Studierenden Budget und Termin einhalten. Auffällig ist, dass die Überschreitungen deutlich abnehmen: Der Termin wurde maximal um 40 Tage überschritten, das Budget maximal um rund 80.000 €. Immerhin 36 der 60 Teilnehmer konnten den Code mit zufriedenstellender Korrektheit abliefern.

Diese Verbesserungen sind nach unserer Beobachtung nicht durch die Gewöhnung an den Simulator und seine Bedienung geprägt: Die Studierenden haben für das erste Spiel ausreichend Zeit und stellen häufig Fragen an den Dozenten. Aus diesen Kontakten wissen wir, dass sich die Studierenden für das erste Spiel zuerst eine Sicherheit im Umgang mit dem Simulator antrainieren und dann das Projekt durchführen. Wir führen darum die Verbesserung im zweiten Spiel auf die Analyseveranstaltung zurück.

Die Schulungsteilnehmer nehmen die Projektsimulation ernst und identifizieren sich mit der Rolle des Projektleiters. Die Diskussionen in den Analyseveranstaltungen zeigen, dass sie die Simulation als Realität erleben. Teilnehmer von Schulungen in der Industrie bestätigen, dass die Simulation ihre Realität widerspiegelt. Sie übertragen ihre Wertungen auf das simulierte Projekt: Während Studierende eher bereit sind, für bessere Qualität den Termin oder die Kosten zu überschreiten, wird in Schulungen in der Industrie verstärkt auf Termin- und Kosteneinhaltung geachtet, auch zu Lasten der Qualität.

4.2 Auswirkungen der Schulung auf praktische Projektarbeit

Wir können kaum bewerten, ob sich die Schulung auf die praktische Projektarbeit auswirkt: Eine empirische Untersuchung von Projektleitern in der Industrie, unter denen auch ehemalige Teilnehmer der SESAM-Schulung sind, kann keine belastbaren Ergebnisse erbringen, denn alle neu erworbenen Fähigkeiten und die

Projektkultur ihrer Firma würden die Einflüsse überlagern. Ein kontrolliertes Experiment, das den Unterschied zwischen Studierenden mit und ohne Schulung zeigt, ist nicht möglich. Der Kontrollgruppe müsste die Schulung verweigert werden, dieses Vorgehen ist nicht vertretbar.

5 Fazit und Ausblick

Die SESAM-Schulung ist integraler Bestandteil der Softwaretechnik-Ausbildung in Stuttgart. Dabei haben wir den Eindruck gewonnen, dass die Studierenden besser auf Praktika vorbereitet werden als einzig durch Software-Engineering-Vorlesungen. Aufgrund des Interesses an SESAM im akademischen und industriellen Umfeld haben wir bereits Erfahrungen mit Schulungen außerhalb der Universität Stuttgart gesammelt, die unseren Eindruck bestätigen.

Der anspruchsvollste Teil der Schulung ist die Auswertung. Um diese zu erleichtern, erstellen wir eine Anleitung, die das Vorgehen bei einer Auswertung beschreibt und anhand verschiedener Beispiele illustriert. Die Auswertungswerkzeuge werden verbessert, um die Routineaufgaben automatisiert auszuführen.

Literatur

- [Aho03] J. J. Ahonen, T. Junttila: A Case Study on Quality-Affecting Problems in Software Engineering Projects. Proc. of the IEEE Intl. Conf. on Software-Science, Technology & Engineering. 2003.
- [Boe00] B. W. Boehm et al.: Software Cost Estimation with Cocomo II. Prentice Hall PTR. 2000.
- [Dra98] A. Drappa: Quantitative Modellierung von Softwareprojekten. Dissertation. Universität Stuttgart. Shaker Verlag. 1998.
- [Jon96] C. Jones: Applied Software Measurement. 2. Auflage. McGraw-Hill. 1996.
- [Lic98] H. Lichter, P. Mandl-Striegnitz: A Case Study on Software Project Management in Industry – Experiences and Conclusions. Proc. of the European Software Measurement Conference. Antwerpen. 1998.
- [Lud94] J. Ludewig (Hrsg.): SESAM – Software Engineering Simulation durch Animierte Modelle. Bericht der Fakultät Informatik 5/94. Universität Stuttgart. 1994.
- [Lud99] J. Ludewig: Softwaretechnik in Stuttgart – ein konstruktiver Informatik-Studiengang. Informatik-Spektrum 22(1). 1999.
- [Lud06] J. Ludewig (Hrsg.): Praktische Lehrveranstaltungen im Studiengang Softwaretechnik. Bericht der Fakultät Informatik. Universität Stuttgart. 4. Auflage. 2006.
- [Man01] P. Mandl-Striegnitz: How to Successfully Use Software Project Simulation for Educating Software Project Managers. Proc. of the 31st Frontiers in Education Conference. 2001.
- [Opf02] S. Opferkuch: Eine Untersuchung zum Einsatz komplexer Simulationsmodelle in der Projektmanagement-Ausbildung. Diplomarbeit. Universität Stuttgart. 2002.
- [Opf06] S. Opferkuch (Hrsg.): Auswertungen von SESAM-Spielen. Informatikbericht der Fakultät 5. Universität Stuttgart. 2006.