

# Vier Formen der Erfahrungsvermittlung im Studium

*Kurt Schneider*

FG Software Engineering, Universität Hannover

Welfengarten 1, 30167 Hannover

Kurt.Schneider@Inf.Uni-Hannover.de

## Zusammenfassung

*Software Engineering ist eines der Fächer, in denen Wissensvermittlung nicht ausreicht. Bei Software-Qualität oder im Software-Entwurf ist darüber hinaus Erfahrung erforderlich – und Erfahrung schließt eigenes Erleben und dadurch erzeugte Gefühle ein.*

*Wie kann aber Erfahrung in diesem Sinne vermittelt werden? Schon lange werden verschiedene Praktikumsformen diskutiert, die dies leisten sollen. Im vorliegenden Beitrag stelle ich vier Ansätze zur gezielten Vermittlung von Erfahrungen kurz vor. Sie werden danach verglichen und auf ihre Tauglichkeit für die Software-Engineering-Lehre hin diskutiert.*

## 1 Einführung

Im Software Engineering gibt es zahlreiche Techniken und Methoden, die Studenten einfach kennen müssen. Dieses Wissen zu vermitteln, ist eine Aufgabe der Software-Engineering-Lehre. Aber solches Wissen reicht bekanntermaßen nicht aus, um praktische Projekte zu bearbeiten. Vielmehr sind Fertigkeiten und Erfahrungen erforderlich, die das Wissen ergänzen müssen.

Software Engineering insgesamt und mehrere seiner Teilgebiete haben einige Eigenschaften, die für besonders „erfahrungslastige“ Disziplinen typisch sind [Sch 02]:

- Technisches Wissen und wirtschaftlich-pragmatisches Urteilsvermögen müssen zusammenkommen.
- Die Rahmenbedingungen und Traditionen der Anwender sind zu beachten. Das scheinbar Optimale ist nicht immer vermittelbar.
- Es gibt keine klaren Kriterien für eine beste Lösung. Gangbare Lösungen müssen in der Diskussion erarbeitet werden.

Zum Beispiel braucht ein Software-Architekt diese Fähigkeiten, um zu einer befriedigenden Lösung zu gelangen. Aber auch Qualitätsbeauftragte müssen beurteilen können, welche Maßnahmen durchsetzbar sind. Nahe liegend mag scheinen, Studierende in „echte Projekte“ in Wirtschaftsunternehmen zu stecken, damit sie dort ihre Erfahrungen machen. Diese gute Idee hat aber leider auch einige Nachteile:

- Nicht an jedem Hochschulstandort sind auf Dauer ausreichend viele Praktikumsplätze vorhanden. Firmen scheuen oft davor zurück, ständig für wenige Wochen neue Praktikanten einzulernen.
- Die Pflicht, einen Praktikumsbericht zu schreiben, ersetzt nicht Betreuung und bietet kaum Unterstützung zur eigenen Reflexion. Es kann also durchaus gelingen, dass Studierende zwar „in der Praxis“ arbeiten und dort auch an Vorgängen teilhaben, die durchaus typisch sind – dies aber gar nicht richtig wahrnehmen. Wie sollen sie auch erkennen, was nützliche Erlebnisse sind, und diese in nutzbare Erfahrung übersetzen?
- Schließlich führen Praktika in den neuen Bachelor-Studiengängen zu einem Engpass im ohnehin sehr kurzen Studienplan. Wenn in sechs Semestern die Berufsbefähigung erworben werden soll, kann man sich kaum leisten, zwei davon für Bachelorarbeit und Praktikum aufzuwenden. Irgendwann muss ja auch das nötige Grundwissen vermittelt werden.

An der Universität Hannover wurde daher das Praktikum aus dem Studienplan genommen – wenn auch mit ausdrücklichem Bedauern. In der Industrie ist man sich der Bedeutung von *Erfahrung* sehr bewusst. Es gibt in vielen großen Firmen Initiativen, die den Wissens-, aber auch Erfahrungsaustausch explizit anstreben und durch Projekte unterstützen [Joh 99; Con 00; Din 00]. Für DaimlerChrysler war ich am SEC-Projekt beteiligt, in dem auf mehreren Ebenen der Austausch und die Nutzung von Erfahrungen betrieben wurde [Lan 99].

Unsere Definition für Erfahrung war:

***Erfahrung* =<sub>Def</sub> *Beobachtung* + *Gefühl* + *Schlussfolgerung***

Eine Beobachtung kann nur machen, wer an einem Vorgang beteiligt ist oder ihm zumindest beiwohnt. Erfahrung kann nicht aus der Theorie abgeleitet werden. Wichtig ist dabei das Gefühlsmoment: Wird die Beobachtung durch Euphorie – oder aber durch ein Gefühl der Enttäuschung, der Frustration oder des Ärgers – begleitet, so prägt sich die Beobachtung tiefer ein und wird mit Assoziationen angereichert. Die Schlussfolgerung (oft mit sehr hypothetischem Charakter) macht aus dem Eindruck des Erlebten eine Lehre, die sich in der Zukunft wiederverwenden lässt. Wird Erfahrung weitergegeben, so wird sie häufig auf diese Schlussfolgerung (als Verhaltensregel oder Mahnung) verkürzt; dann fehlen ihr aber die Einprägsamkeit des Gefühls und die Authentizität einer konkreten Beobachtung.

Wie gibt man aber (vollwertige) Erfahrungen im Studium weiter? An der Hochschule wird die *Wissensweitergabe* ja über Vorlesungen, Übungen und Seminare bewältigt, Praktika richten sich schon stärker an das praktische Können und die Erfahrung. Ich möchte im vorliegenden Beitrag vier verschiedene Ansätze vorstellen, die im Studium explizit (auch) die Weitergabe von *Erfahrungen* bewirken sollen. Dann vergleiche ich die Kernideen und arbeite heraus, wofür sich welche Form gut eignet und wann weniger. Daraus ergeben sich einige Empfehlungen, wie man Erfahrungen (nicht Wissen) im Studium handhaben kann. Der Beitrag ist als Diskussionsimpuls zu verstehen.

## 2 Simulation eines ganzen Software-Projekts: SESAM

Das Projekt SESAM wurde vor über zehn Jahren am Lehrstuhl von Prof. Jochen Ludwig an der Universität Stuttgart gestartet [Dra 95]. Ihm liegt die Idee zugrunde, ein interaktives Simulations- und Lehrspiel für Software-Projektleiter zu entwickeln. Dabei sollte ein gesamtes Software-Projekt simuliert werden, nur der Projektleiter selbst war real und interagiert mit dem simulierten „Rest“ seines Projekts. SESAM wurde schon in den frühen 90er-Jahren in der Lehre eingesetzt [Sch 94]. Es wird auch heute noch weiterentwickelt [ASE 04].

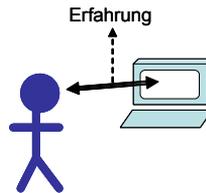


Abb. 1.: SESAM: Erfahrung erwächst aus Interaktion mit dem simulierten Projekt

Im Kern braucht SESAM ein umfassendes dynamisches Modell des simulierten Entwicklungsprojekts [Dei 94]. Es muss auf „korrektes“ (das heißt: der Vorlesung entsprechendes) Verhalten des Projektleiters reagieren, aber auch auf Verstöße und Fehler ein plausibles Verhalten zeigen. So ein Modell ist schwer zu validieren, und die Modellentwicklung ist selbst eine spannende Forschungsaufgabe. Rodriguez et al. beziehen sich auf einen anderen Simulationsansatz, aber mit ähnlicher Absicht [Rod 04].

Im Hinblick auf die Erfahrungsvermittlung versucht SESAM, in einem simulierten und absolut nicht realen Umfeld authentische Erfahrungen zu vermitteln. In erster Näherung soll es ja keine Rolle spielen, dass die Reize aus dem Projekt nur durch den Computer vermittelt werden. Der Projektleiter soll *beobachten*, wie sich seine Handlungen und die Eigendynamik des simulierten Projekts auswirken. Versäumt er zum Beispiel, ein Review anzuberaumen, so hat er später kaum eine Möglichkeit, die Qualität des Entwurfs zu beurteilen. Dies ist so unerfreulich wie in einem entsprechenden realen Projekt. Auch die *Gefühlsantwort* kann daher ähnlich und vergleichbar intensiv

sein. Wird nun noch die gleiche *Schlussfolgerung* gezogen, dann ist Erfahrung in allen drei Aspekten vom Modellentwickler an den spielenden Projektleiter vermittelt worden.

Wir haben bei den ersten Einsätzen von SESAM am Rande bemerkt, dass die computerbasierte Simulation nicht unbedingt die authentischsten Gefühle hervorrief. Solange der Simulator noch nicht zur Verfügung stand, hatten wir das Modell schon (mit Excel-Unterstützung) „manuell“ benutzt [Dei 94]. Die von Hand errechneten Reaktionen wurden von Assistenten den Spielern in definiertem Wochenraster überbracht. Menschliche Betreuer als Informationsüberbringer führten offenbar zu einer stärkeren gefühlsmäßigen Identifikation mit dem simulierten Projekt und erhöhten offenbar die Glaubwürdigkeit der simulierten Informationen aus Spielersicht. Dagegen wirkt (wieder nach Aussage der damaligen Studenten) die reine Simulation am Rechner eher steril und unecht – keine gute Voraussetzung für authentische Gefühle oder Erfahrungen.

In den letzten Jahren hat SESAM hier sicher Fortschritte gemacht, doch mir geht es hier um eine allgemeinere Beobachtung zu Simulationen: Eine „zu glatt und zu schnell“ laufende Simulation ist wohl generell eher kontraproduktiv für die Erfahrungsvermittlung. Wenn das simulierte Projekt in wenigen Minuten abgewickelt werden kann, wird allein dadurch Planung fast unmöglich, und Situationen wie banges Warten werden verhindert. Auch kann sich nicht in Minuten dieselbe Spannung aufbauen, wie wenn man auf die Antwort drei Tage warten muss. Spannung, banges Warten und dergleichen bräuchte man für starke, nach der obigen Definition vollwertige Erfahrungen. Einen anderen Anspruch kann aber auch die schnell laufende Simulation durchaus einlösen: Nämlich festzustellen, ob man das gelernte *Wissen* des Projektleiters schnell reproduzieren kann. Damit kann man richtiges Verhalten üben und falsches erkennen.

### 3 Erfahrung pur, obwohl man etwas anderes tut: Agile Hour

Agile Methoden sind in den letzten Jahren viel diskutiert worden. Besonders eXtreme Programming von Kent Beck hat großen Zuspruch, aber auch extreme Ablehnung ausgelöst [Bec 00]. Inzwischen hat die Kontroverse an Schärfe verloren; man versucht nun, die Stärken von agilen und anderen Ansätzen zu verbinden. Dazu gibt es interessante Bücher, zum Beispiel [Boe 03]. Für viele Hochschulen gehört Grundwissen über agile Methoden heute zum Handwerkszeug des Software-Ingenieurs. Aber wieder reicht es nicht, etwa die XP-Praktiken aufzulisten und zu definieren, was dabei getan wird. Wieder fehlen *Erfahrungen* mit ihren drei Aspekten: eigene Beobachtung, Gefühl und abgeleitete Schlussfolgerung. Gerade Verfahren wie agile Methoden, die so stark auf die gute Zusammenarbeit von Entwicklern und Kunden im Team bauen, brauchen dort eine breite gemeinsame Erfahrungsbasis, auf die sie sich abstützen können.

Offenbar wurde dieses Defizit auch von anderen empfunden, denn schon seit einiger Zeit gibt es einen interessanten Ansatz, der gezielt die Vermittlung von Erfahrungen zu einem Ausschnitt von agilen Methoden zum Ziel hat: die Extreme Hour [Mer 04] von Peter Merel (wir nennen sie Agile Hour, weil es uns nicht nur um XP geht). Das Prinzip lautet: Es werden zwei Iterationen eines Konstruktionsprojekts durchgespielt. Insgesamt

werden sieben Schritte unterschieden: (1) Erkundung durch Prototypen („Spikes“ genannt), (2) Schreiben von Story Cards, (3) Aufwandsschätzung, Priorisierung, (4) Auswahl und Implementierung. Schritte 2-4 (eine Iteration) werden dann noch einmal ausgeführt. In den meisten Schritten passieren noch einige weitere Aktivitäten (z.B. Refactoring), aber für diese Diskussion ist das ohne Belang. Wichtig ist nun:

- Jeder der sieben Schritte dauert genau 10 Minuten (die Agile „Hour“ also 70 Minuten).
- Das konstruierte Produkt wird nicht programmiert, sondern gezeichnet.
- Die Entwickler arbeiten paarweise („pair painting“) und haben nur einen Stift pro Paar.

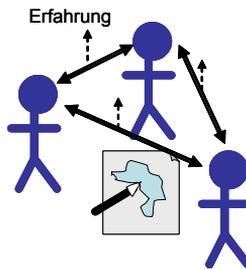


Abb. 2.: Agile Hour: Erfahrung erwächst aus der Interaktion, obwohl nur gemalt wird

Es gibt noch einige weitere Regelungen, die die Agile Hour einem agilen Software-Projekt ähnlicher machen sollen. Was bewirkt eine Agile Hour nun aber in Bezug auf *Erfahrungen*? Wir haben Gruppenübungen zum Thema agile Methoden in Form von Agile Hours durchgeführt. Jede Gruppe umfasste etwa 10 Studierende. Die Übungsbetreuer haben die Agile Hours als Coaches betreut, waren also dabei. Anschließend wurde reflektiert und wir haben mit einem anonymen Fragebogen die Stimmung abgefragt. In den meisten Agile Hours kam es zu Zeitnot und Hektik, mitunter bereitete das Malen in Paaren Probleme, fast immer auch die Integration der Teilergebnisse aller Paare. Die Rückmeldungen sprechen eine klare Sprache: Die Agile Hour vermittelt den Teilnehmern praktische Kenntnisse (so haben viele erst hier das Wechselspiel zwischen Kunden und Entwicklern im *Planungsspiel* richtig durchschaut). Alle fühlen sich *gefühlsmäßig* stark angesprochen, durchaus nicht immer nur mit positiven Gefühlen. Und in der Reflexion haben die Teilnehmer leicht Zusammenhänge zwischen ihrem Verhalten und den Konsequenzen erkannt. Alle Aspekte von Erfahrungen sind also intensiv vermittelt worden. Und das, obwohl gar keine Software entwickelt wurde! Hier ähnelt die Agile Hour dem SESAM-Ansatz: In beiden Fällen ist nur ein Kern authentisch, der Rest hat nicht viel mit einem realen Software-Projekt zu tun. Wir hatten 48 Teilnehmer in den Agile Hours; 31 haben per Fragebogen geantwortet. Alle 31 haben angegeben, die Agile Hour habe ihnen Spaß gemacht. Das reichhaltige mündliche Feedback direkt

nach den Agile Hours half uns, die Intensität der Erfahrungen in so kurzer Zeit als einen Grund dafür zu identifizieren.

#### 4 Gezielte Eingriffe: SWP04

Der dritte vorzustellende Ansatz klingt sicher besonders vertraut. Ähnliche Versuche hat es schon viele gegeben. Im letzten Wintersemester haben 46 Studierende Software-Projekte („SWP04“) in neun Gruppen zu je 5–6 Teilnehmern bei uns durchgeführt. Verschiedene Aufgabenstellungen wurden meist an mehrere Gruppen parallel vergeben. Bei etwas genauerem Hinsehen haben solche universitären Projekte oft nicht viel mit Industrieprojekten zu tun. Zu groß sind die Unterschiede in den Rahmenbedingungen, in der Motivation und in vielen weiteren Aspekten. Wir hielten es für unrealistisch, alle diese Unterschiede überbrücken zu wollen. Wir haben daher entschieden, die meisten Unterschiede zu akzeptieren – aber an einigen wenigen Stellen gezielt zu intervenieren. Die interessantesten waren:

- Strikte **Rollentrennung** zwischen Kunde, Entwicklern und der „Firma der Entwickler“. Wir haben am Lehrstuhl die Rollen verteilt und darauf geachtet, sie nicht zu vermischen. Die (fiktive) Entwicklungsfirma war mit Qualitätssicherern bestrebt, die Qualität ausgelieferter (Teil-)Produkte hoch zu halten. Diese Rollenteilung ist an Hochschulen oft unüblich. Unserer Ansicht nach ist sie aber für viele typische Projekteffekte unabdingbar. Und erfreulicherweise bereitet die Rollenteilung auch für Betreuer kaum zusätzlichen Aufwand.
- **Quality Gates** zur groben Qualitätssicherung. Das ist eine Variante von eher formalistischen Prüfungen an festgelegten Stellen im zuvor definierten Prozess – eine Art verallgemeinerte Meilensteine für das Multiprojektmanagement. Sie laufen ähnlich wie Review-Sitzungen ab und dienen vor allem dazu, festgelegte Mindeststandards an festgelegten Prozessstellen quer über alle Projekte sicherzustellen. Wir haben durch drei Quality Gates einen Qualitätsmindeststandard in allen abgelieferten Produkten erreicht. Quality Gates sind in der Industrie weit verbreitet. Sie sind eine pragmatische Art und Weise, mit vielen parallel laufenden Projekten umzugehen, von denen man sich nicht alle Details ansehen kann.
- **Zeitgutscheine**. Die Zeit der Kunden ist knapp und wertvoll. Dazu gibt es an der Hochschule häufig keine Entsprechung. Indem wir jedem Projekt nur sechs mal 15 Minuten für die Kommunikation mit dem Kunden zusprachen, fühlten sich alle Gruppen gefordert, diese Zeit optimal zu nutzen. Letztlich hat nur eine Gruppe überhaupt alle sechs Gutscheine verbraucht. Fast alle anderen haben sogar noch Prototypen mit dem Kunden diskutiert – in weniger als insgesamt 90 Minuten pro Gruppe, während des ganzen Semesters.

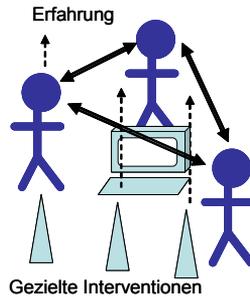


Abb. 3.: SWP04: Erfahrung wird durch gezielte Interventionen provoziert

Wieder haben wir durch Befragung, gezielte Erfahrungserfassung [Sch 00] und durch Fragebögen die Sicht der Studierenden abgefragt. Besonders bei den drei genannten Maßnahmen haben die Studierenden deutlich „Gefühle gezeigt“ und auch geäußert, hier hätten sie unerwartete Erfahrungen gemacht. Aber auch andere Entwicklungstätigkeiten, in die wir nicht eingegriffen haben, wurden als besonders realitätsnah eingestuft. Die Durchführung der Projekte und die dabei verfolgten Zielsetzungen sind in [Lüb 04] detaillierter beschrieben.

## 5 Erfahrungen per Internet vermitteln: Werkzeuge

Der vierte Ansatz unterscheidet sich stark von den oben beschriebenen. Hier geht es nicht darum, neue Erfahrungen zu machen, sondern bestehende Erfahrungen an andere (Studierende) weiterzugeben. Wir haben die Technik LIDs [Sch 00] verwendet, um mit wenig Aufwand den Studentenprojekten Erfahrungen zu entlocken. Die Erfahrungen sollen nun künftigen Projektgruppen zugute kommen. Dazu werden sie in ein Internet-Werkzeug gesteckt, das in einer Bachelor-Arbeit entwickelt wurde [Lir 04]. Besonders interessant ist, dass Leser die dort stehenden „Erfahrungsberichte“ im Stil von Amazon oder eBay bewerten können. So erhalten Autoren und künftige Leser eine schnelle Einschätzung über den Nutzen eines solchen Berichts, der oft „Erfahrungspaket“ heißt.

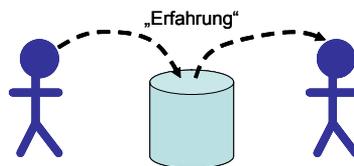


Abb. 4.: Erfahrungswertung mit Werkzeug ist schwieriger, als man denkt

Derlei Werkzeuge werden gerne entwickelt, wo Erfahrungen vermittelt werden sollen. Meiner Überzeugung nach sind zwar Werkzeuge für die Verteilung nötig – aber sie stellen das kleinste Problem dar. Es gelingt nur mit großer Mühe, den Erfahrungsschaarakter (eigene Beobachtung? Gefühl?) über ein schriftliches Erfahrungspaket zu transportieren – egal wie elegant das Werkzeug dafür ist. Diese Ansicht wird von vielen anderen geteilt, die Erfahrungsvermittlung nicht hauptsächlich als Werkzeugproblem sehen, z.B. [Bar 99]. In [Sch 02] habe ich eine ganze Reihe von Missverständnissen und Hürden beschrieben, die der Wieder-Nutzung von Erfahrungen im Wege stehen. Werkzeuge helfen dabei kaum. Dies deckt sich mit Erkenntnissen bei DaimlerChrysler und hat wiederum Konsequenzen auf die Gestaltung *dennoch* nützlicher Werkzeuge [Sch 01]. Wir sehen Werkzeuge nur als einen eher kleinen Mosaikstein im größeren Zusammenhang der systematischen Erfahrungsnutzung. Sie werden hier hauptsächlich erwähnt, um auf diese Tatsache hinzuweisen.

## 6 Diskussion und Schlussfolgerungen

In diesem Beitrag habe ich vier unterschiedliche Verfahren vorgestellt, um im Rahmen des Informatik-Studiums gezielt und ausdrücklich Erfahrungen (nicht Wissen) zu vermitteln. Während SESAM, die Agile Hour und das Software-Projekt mit gezielten Interventionen auch dazu beitragen, Erfahrungen aufzubauen, erheben wir mit dem Werkzeug diesen Anspruch nicht: Hier geht es nur um die Verteilung von bestehenden Erfahrungspaketen. Da Erfahrung nach der obigen Definition aber eigene Beobachtung, Gefühle und Hypothesen einschließt, kann das Werkzeug kaum Mehrwert stiften, alles hängt von den Erfahrungspaketen selbst ab. In Tabelle 1 ist gegenübergestellt, ob und wie die vier Ansätze die drei Definitionsbestandteile von Erfahrungen an die Teilnehmer vermitteln.

Tab. 1: Vergleich der vier Verfahren: Wodurch werden die Erfahrungsaspekte umgesetzt?

Erfahrungsaspekt	SESAM	Agile Hour	SWP04	Werkzeug
Eigene Beobachtung	Reaktion des simulierten Projekts	Verhalten der anderen Teilnehmer, Zeitknappheit	Besonders bewusst bei gezielten Interventionen	(entfällt)
Gefühle	Je schneller und glatter die Simulation läuft, umso weniger	Wenn Hektik entsteht, schlagen auch Gefühle hoch	Zum Teil sehr stark, da echte Konsequenzen drohen (z.B. kein Schein)	(aus zweiter Hand) Bewertung ist auch gefühlsbeladen
Schlussfolgerungen	Kaum d. SESAM selbst, erfordert Nachbesprechung	Nachbesprechung wichtig; sprudeln dann	Entstehen besonders um Interventionen	Bildet sich aus Inhalt und Bewertungen

Freilich reichen unsere Rückmeldungen und Fragebögen nicht aus, um die hier präsentierten Schlussfolgerungen statistisch zu belegen; aber dieses Material hilft uns doch immerhin, deutlich über das Niveau persönlicher Vermutungen hinauszukommen.

Der direkte Vergleich zeigt, dass mit dem Werkzeug eine andere Art von Erfahrungsvermittlung gemeint ist, nämlich die technische Speicherung und der Vertrieb. Es zeigt sich auch, dass Nachbesprechungen für die drei anderen Varianten sehr wichtig sind, was zwar nicht überrascht, aber auch nicht selbstverständlich ist [Lew 01]. Besonders eine reine Simulation wie SESAM erfordert unbedingt eine Nachbesprechung, damit sich nicht falsche Schlüsse festsetzen; schon vor zehn Jahren waren viele Teilnehmer schnell bereit, unerwünschtes Projektverhalten der angeblich „fehlerhaften“ Simulation zuzuschreiben.

SWP04 als der „üblichste“ Ansatz hat uns gezeigt, dass eine kleine Zahl sehr deutlicher Interventionen zu bewusst wahrgenommenen Erfahrungen im vollen Sinn der Definition führen kann. Damit hat man zwar nur wenige Aspekte abgedeckt, diese aber wirksam vermittelt. Wir finden diesen Ansatz sehr vielversprechend, zumal er auch noch den Aufwand in Grenzen hält. Wir werden nun mit anderen Interventionen experimentieren als beim ersten Durchgang.

Das Prinzip der Agile Hour zur Erfahrungsvermittlung scheint zu funktionieren. Gewiss ist auch der Neuigkeitswert für die positive Reaktion mitverantwortlich. Im nächsten Schritt denken wir über ähnliche Anwendungen (nicht nur agile Prozesse) nach, bei denen Erfahrungen konzentriert erzeugt und kompakt vermittelt werden sollen. Die Agile Hour ist geradezu dafür optimiert, in kürzester Zeit Gefühle wachzurufen und echte Erfahrung möglich zu machen. So einen Ansatz kann man freilich nicht ständig einsetzen, weil das Abrufen von Gefühlen auch anstrengend ist und sich nicht beliebig häufig wiederholen lässt.

In Hannover werden wir weiter gezielt nach Techniken forschen, die Erfahrungen erzeugen, sie durch Speicherung und Verteilung „frisch halten“ und für die Nutzung bereit machen. Für Anregungen und Gedankenaustausch auf diesem Weg sind wir dankbar.

## Literatur

[ASE 04] Abteilung Software Engineering: SESAM-Projektseite: <http://www.iste.uni-stuttgart.de/se/research/sesam/index.html>.

[Bar 99] Bartsch-Spörl, B.: Transfer of Experience – Issues Beyond Tool Building. Workshop on Learning Software Organizations, Kaiserslautern, Germany.

[Bec 00] Beck, K.: Extreme Programming Explained, Addison-Wesley.

[Boe 03] Boehm, B. and R. Turner: Balancing Agility and Discipline – A Guide for the Perplexed, Addison-Wesley.

[Con 00] Conradi, R. and T. Dingsoyr: Software Experience Bases: A Consolidated Evaluation and Status Report. Profes, Oulu, Finland.

- [Dei 94] Deininger, M. and K. Schneider: Teaching Software Project Management by Simulation – Experiences with a Comprehensive Model. Conference on Software Engineering Education (CSEE), Austin, Texas.
- [Din 00] Dingsoyr, T.: An Evaluation of Research on Experience Factory. Workshop on Learning Software Organizations, Oulu, Finland.
- [Dra 95] Drappa, A., M. Deininger, et al.: "Forschungsprojekt SESAM." Informatik Forschung und Entwicklung. 10 (1995)(1): 49-50.
- [Joh 99] Johansson, C., P. Hall, et al.: Talk to Paula and Peter – They are Experienced. International Conference on Software Engineering and Knowledge Engineering (SEKE '99). Workshop on Learning Software Organizations, Kaiserslautern, Germany, Springer.
- [Lan 99] Landes, D., K. Schneider, et al.: "Organizational Learning and Experience Documentation in Industrial Software Projects." International Journal on Human-Computer Studies (IJHCS) 51(Organizational Memories).
- [Lew 01] Lewerentz, C. and H. Rust: Die Rolle der Reflexion in Softwarepraktika. Software Engineering im Unterricht der Hochschulen (SEUH 2001), Zürich, dpunkt.verlag.
- [Lir 04] Liro, T.: Konzept und Realisierung eines Werkzeugs zur Erhebung, Verwaltung und Bewertung von Erfahrungen. Fachgebiet Software Engineering. Hannover, Universität Hannover: 40.
- [Lüb 04] Lübke, D., T. Flohr, et al.: Serious Insights through Fun Software-Projects. EuroSPI 2004: European Software Process Improvement Conference, Trondheim, Norway, Springer.
- [Mer 04] Merel, P.: Extreme Hour: <http://c2.com/cgi/wiki?ExtremeHour>.
- [Rod 04] Rodríguez, D., M. Satpathy, et al.: Effective software project management education through simulation models. An externally replicated experiment. International Conference on Product Focused Software Process Improvement (PROFES), Kansai Science City, Japan, Bomarius, F.
- [Sch 94] Schneider, K.: Ausführbare Modelle der Software-Entwicklung. Struktur und Realisierung eines Simulationssystems. Zürich, vdf.
- [Sch 00] Schneider, K.: LIDs: A Light-Weight Approach to Experience Elicitation and Reuse. Product Focused Software Process Improvement (PROFES 2000), Oulo, Finland, Springer.
- [Sch 01] Schneider, K. and T. Schwinn: "Maturing Experience Base Concepts at DaimlerChrysler." Software Process Improvement and Practice 6: 85-96.
- [Sch 02] Schneider, K.: "What to Expect from Software Experience Exploitation." Journal of Universal Computer Science (J.UCS). [www.jucs.org](http://www.jucs.org), 8(6): 44-54.