

Game Engineering – Computerspiele in der Software-Engineering-Ausbildung

Daniel Volk

Institut für Softwaretechnologie
Fakultät für Informatik
Universität der Bundeswehr München
daniel.volk@unibw.de

Zusammenfassung

Softwarepraktika in Projektform leisten einen wertvollen Beitrag zur Software-Engineering-Ausbildung im Unterricht der Hochschulen. Insbesondere hier können Studenten in einem der Realität nahen aber dennoch sicheren Umfeld das im späteren Beruf unbedingt notwendige Handwerkzeug zur erfolgreichen Softwareentwicklung erwerben. Eine klassische Herausforderung in der Konzeption solcher Praktika stellt die Auswahl interessanter Themengebiete dar. Computerspiele sind genau eine solch interessante Domäne mit mittlerweile nicht unerheblicher Relevanz und haben zudem den ganz besonderen Ruf, als Themenfeld mit einem deutlichen Motivationsschub einherzugehen. Im Folgenden sind nun der Ansatz und erste Erfahrungswerte eines Game-Engineering-Praktikums beschrieben, das im Rahmen des Hauptstudiums an der Fakultät für Informatik der Universität der Bundeswehr München durchgeführt wurde.

1 Einleitung

Den Ausgangspunkt für erste Überlegungen in Hinblick auf das beschriebene Praktikum im Bereich *Game Engineering* (GE) stellte das an der Universität der Bundeswehr München und an der TU Dresden etablierte Konzept eines frameworkbasierten Softwarepraktikums [Dem99] [Dem05] dar, das an beiden Universitäten ein fester Bestandteil des Grundstudiums ist und bereits von zahlreichen

Studententeams erfolgreich absolviert wurde. In diesem Umfeld entstand die Idee eines fortgeschrittenen Praktikums im Hauptstudium, das die im vorangegangenen Programmierpraktikum erworbenen Fähigkeiten im Bereich Software Engineering (SE) vertiefen sollte. Insbesondere erschien es hierbei als wichtig, sowohl die zusätzliche Komplexität als auch die Motivation der Studenten auf natürliche Weise aus dem anvisierten Themengebiet abzuleiten. Die Umsetzung selbiger Idee als *GE-Praktikum*, ergo einer domänenspezifischen SE-Ausbildung [Jtf04, S.8] mit Fokus auf der Entwicklung von Computerspielen, erwies sich in Bezug auf obige Überlegungen aus den folgenden Gründen von Anfang an als idealer Kandidat: Obwohl die üblichen Projektparameter industrietypischer Spieleprojekte [Bad00] für ein Praktikum natürlich illusorisch sind und erheblich eingeschränkt werden müssen, weist die Entwicklung von Computerspielen zusätzlich auch eine qualitative Komplexität auf, die ein solches Projekt trotz Skalierung auf Praktikumsgröße interessant werden lässt [Swe05]. So ist der unerlässliche Anteil des *Game Design* (des theoretischen Spielentwurfs) aufgrund seiner kreativen Natur ein klassischer Reibungspunkt, welcher u.a. durch Betonung nicht funktionaler Anforderungen Friktionen im Bereich der Analyse zu verursachen vermag [Cal05]. Zudem erfordert die schwere Greifbarkeit des *Game Play* (des eigentlichen Spielerlebnisses) typischerweise auch ein prototypenorientiertes und evolutionäres Vorgehen, das nicht unerheblich auf das Projektrisiko ausstrahlt. Abgesehen von einem interessanten und komplexen Profil der Computerspiele-Entwicklung ist aber natürlich auch der zugehörige Motivationsschub nicht zu übersehen, der faktisch regelmäßig zu besseren Ergebnissen im Rahmen entsprechender Lehrveranstaltungen führt [Bur08] [Rez06]. Um allerdings den Motivationsbonus nicht verpuffen zu lassen, erschien es als eine Notwendigkeit, nicht nur einzelne Aspekte der Spieleentwicklung in ein klassisches SE-Praktikum zu übertragen, sondern zudem ein authentisches Umfeld zu schaffen. In Hinblick auf die konkrete Umsetzung des angedachten *GE-Praktikums* ergaben sich also die folgenden Zielsetzungen:

■ **Das primäre Ziel – der Anteil Software Engineering**

Das Praktikumskonzept sollte also zum einen eine Fortführung der bisherigen Lehranteile im Bereich SE darstellen und sich in diese integrieren. Die anvisierten Lerneffekte sollten sich in natürlicher Form aus der Domäne ergeben.

■ **Das sekundäre Ziel – der Anteil Game Development**

Um einen glaubwürdigen Kontext für die entsprechenden Studentenprojekte zu schaffen, sollte zum anderen der originäre Modus Operandi der Spieleentwicklung (bis zu einem vertretbarem Grad) simuliert werden.

2 Die Bestandsaufnahme

In Vorbereitung des Praktikums wurden entsprechend der oben erwähnten Zielsetzungen also zunächst die beiden Ausgangspunkte näher betrachtet. Einerseits waren dies der SE-Anteil des bestehenden Informatik-Curriculums, andererseits wurde hier das Curriculum der IGDA (*International Game Developer Association*) [Igd03] zu Rate gezogen, das die für die Spieleentwicklung relevanten Themen umreißt.

2.1 Das bestehende Software-Engineering-Curriculum

Der für diese Betrachtung relevante Anteil des Curriculums im Diplomstudengang Informatik besteht zum einen aus den Vorlesungen Objekt-Orientierte Programmierung und SE sowie dem bereits erwähnten Programmierpraktikum. Um nun die Einbettung des Praktikums in das bestehende Curriculum zu erläutern, werden im Folgenden kurz die Inhalte der Veranstaltungen skizziert:

■ Objekt-Orientierte Programmierung (3. Trimester)

Aufbauend auf zwei initialen Vorlesungen zur Vermittlung der Grundfertigkeiten im Bereich der Softwareentwicklung, behandelt selbige Vorlesung zum einen objekt-orientierte Konzepte, das zugehörige Paradigma am Beispiel der Sprache Java sowie weiterführende Konzepte wie Wiederverwendung auf Basis von Klassenbibliotheken, Entwurfsmustern und Rahmenwerken. Zum anderen wird aber auch die objekt-orientierte Methode besprochen, am Beispiel ausgewählter Entwicklungsartefakte vertieft und in einfache Vorgehensmodelle eingebettet.

■ Das Programmierpraktikum (4. Trimester)

Aufbauend auf den Kenntnissen der Vorlesung Objekt-Orientierte Programmierung vermittelt nun das Programmierpraktikum einen ersten Einblick in die Organisation von Gruppenarbeit und erlaubt den Studenten, einen einfachen Entwicklungsprozess in der Realität zu erleben. Um dies zu erreichen wird die Situation eines Softwarehauses simuliert, das auf Basis kleiner Entwicklerteams einfache Projekte realisiert. In diesem Rahmen werden nun auch die wichtigsten Typen von Entwicklungsartefakten von den Studenten erstmalig in einem größeren Sinnzusammenhang genutzt. Um den Studenten hierbei Halt zu bieten, sind sowohl der Projektrahmen als auch das zu nutzende Framework (das SalesPoint-Framework, vgl. [Dem99]) vorgegeben. Weiterhin wird auch der konsequente Gebrauch von Softwarewerkzeugen im Gruppenrahmen eingeübt.

■ Software Engineering (6. Trimester)

Nach dem praktischen Einüben der Basisfertigkeiten des SE und dem Lösen typischer Problemstellungen im Rahmen des (ersten) eigenen Projekts, erfolgt

nun die theoretische Festigung und Erweiterung selbigen Wissensstands. So werden in der SE-Vorlesung nun u.a. komplexere Vorgehensmodelle (z.B. auch evolutionäre Modelle) aber auch weitere Modellierungskonzepte behandelt und bekannte Anteile detailliert. Zudem werden bis dato kaum behandelte Bereiche wie z.B. die der Projektplanung oder der Qualitätssicherung näher ausgeführt.

Das GE-Praktikum ist im Anschluss an die genannten Veranstaltungen einzuordnen und stellt damit eine Fortführung des Programmierpraktikums dar. Hierbei sollte der bekannte Modus Operandi zumindest im groben Rahmen beibehalten, die thematische Ausrichtung jedoch an den Wissenstand der SE-Vorlesung angepasst werden.

2.2 Das IGDA-Curriculum

In Hinblick auf die inhaltliche Ausrichtung fanden sich für das anvisierte Spektrum an SE-Fertigkeiten im vorliegenden IGDA-Curriculum die Anteile **Game Programming** und **Game Production**. Unter ersterem Bereich sind hierbei klassische Informatikthemen wie z.B. Datenstrukturen und Algorithmen zu verstehen, die weitestgehend durch Einführungsvorlesungen der ersten beiden Semester abgedeckt werden. Weiterhin waren hier jedoch auch spezielle Einzelthemen der Informatik, wie z.B. selektive Anteile der Computergrafik, aufgeführt, die dem vorgesehenen Teilnehmerkreis (noch) nicht zur Verfügung standen. Der zweite Bereich umriss dagegen ein Mischspektrum aus klassischen SE-Themen, durchsetzt von Aspekten des Projektmanagements – beides Anteile, die zum Großteil durch die vorliegenden Vorlesungen des dritten und sechsten Trimesters behandelt wurden. Zusammengefasst erschienen also die bis hierhin beschriebenen Anteile eines GE-Praktikums als verhältnismäßig konventionell, wenn auch etwas »eingefärbt« durch die spezielle Domäne. Selbige Erkenntnis wurde auch durch eine erste Evaluation der zugehörigen Literatur im Kernbereich SE [Fly04] sowie im umschließenden Bereich des Projektmanagements [Cha06] bestätigt. Nachdem das angestrebte Praktikumsformat nun aber auch einen gewissen Grad an Authentizität aufweisen sollte, waren neben den reinen Informatikanteilen auch Bereiche zu inkludieren, die den Studentenprojekten zudem den Charakter wirklicher Spieleentwicklungen verleihen würden. Eine umfangreiche inhaltliche Produktion erschien diesbezüglich als unangebracht, zumal in diesem Fall eine Ablenkung von den eigentlichen Kerninhalten des Praktikums zu erwarten gewesen wäre. In gewisser Weise erschien selbige Fragestellung der Initialisierung eines neuen Game-Development-Curriculums ähnlich zu sein, bezüglich dessen allgemein ein »**Game-Design-first**«-Ansatz empfohlen wird [Raj05]. In diesem Sinne ist das Game Design als Kernanteil der Spieleentwicklung zu verstehen, von dem alle anderen Teile der Entwicklung abzuleiten sind und der diese zusammenhält [Rol04].

3 Das Praktikumskonzept

Ausgehend von der Evaluation der beiden Curricula ergab sich also für die Konzeption des *GE*-Praktikums folgende Ausprägung: Den Ausgangspunkt der Entwicklung sollte ein *Game-Design*-Anteil bilden, der als Aufgabenstellung für das jeweilige Projekt zu verstehen war. Ableitend hiervon sollte unter Verwendung domänenspezifischer SE-Techniken die Entwicklung der technischen Anteile auf Basis bekannter Implementierungstechniken erfolgen, was einen engen Fokus des Praktikums sicherstellen sollte (vgl. auch [Mcc04]). Die Verwendung allzu sehr spezialisierter Technologie sollte hierbei weitestgehend vermieden werden, ggfs. durch Einschränkungen des *Game Design*, am besten aber durch Nutzung eines geeigneten Frameworks. Den Rahmen des Praktikums sollte ein (bis zu einem gewissen Grad) authentisches Projektmanagement gemäß den üblichen Charakteristika der Spieleindustrie bilden.

3.1 Die Inhalte

Nach der abstrakten Formulierung des Praktikumskonzepts war es nun möglich, hieraus die konkreten Inhalte für das *GE*-Praktikum zu extrahieren, die sich aus den Anteilen des SE und des *Game Development* folgendermaßen zusammensetzen:

■ Die primären Inhalte – der Anteil Software Engineering

Ableitend aus dem Szenario der Spieleentwicklung sollten zum einen Aspekte vertieft werden, die bereits aus dem Programmierpraktikum bekannt waren, zum anderen sollten entsprechend der Vorlesung SE bestimmte Aspekte verändert werden. Hinsichtlich ersterer Forderung wurde so z.B. der bereits bekannte objekt-orientierte Analyse-Design-Implementierungszyklus aufgrund schwammiger Anforderungen des *Game Design* erschwert oder auch das Konfigurationsmanagement der Teams aufgrund des hinzugekommenen *Asset Management* (der Verwaltung der Spielinhalte) vor neue Herausforderungen gestellt. Zweiterer Forderung wurde z.B. durch Erweiterung des zuvor weitestgehend sequenziellen Prozessmodells auf Basis modellgetriebener Entwicklung durch evolutionäres Prototyping Rechnung getragen. Um die hiermit einhergehenden Risiken auszugleichen, wurden zudem Refactoring-Techniken und eine aktive Verwendung von Entwurfsmustern forciert. Die bekannten Phasen der Softwareentwicklung wurden zudem um eine Planungsphase bereichert.

■ Die sekundären Inhalte – der Anteil Game Development

Der charakteristische Kontext sollte sich zum einen aus der Strukturierung des Praktikums, gemäß des in der Spielebranche üblichen Projektmanagements, der Übernahme der entsprechenden Rollen bzw. Verfahrensweisen und der Integration eines *Game-Design*-Anteils ergeben.

3.2 Die Bewertungskriterien

Als Bewertungsrahmen für die erbrachten Leistungen im unbenoteten Hauptstudiums-Praktikum diente ein anvisierter Arbeitsaufwand von 180 Stunden pro Teilnehmer. Hinsichtlich der Wertigkeit der generierten Artefakte wurden die SE-Anteile den *Game-Design*-Anteilen gegenüber prinzipbedingt bevorzugt. Äquivalent wurde auch der Qualität des Entwicklungsprozesses und der technischen Umsetzung der Prototypen im Verhältnis mehr Beachtung beigemessen als dem Endergebnis im Sinne der technischen und inhaltlichen Quantität (es lagen keine Erfahrungswerte in Form konkreter Metriken aus früheren Praktika vor). So wurde z.B. keine möglichst facettenreiche Spielwelt eingefordert, die korrekte Umsetzung der Kernmechanik, also des zentralen Elements des Spielkonzepts, war dagegen obligatorisch. Die Bewertung selbst erfolgte hierbei auf Basis der im Projektverlauf erstellten Artefakte:

- Game-Development-Artefakte
 - 3× *High-Concept*-Dokument (2-3 Seiten)
 - 1× *Game-Design*-Dokument (abhängig vom Spieltyp 20-30 Seiten)
- Software-Engineering-Artefakte
 - Planungsanteile (1× Grobplanung sowie 2× Feinplanung)
 - Priorisierte Anforderungsliste (technisch/inhaltlich)
 - Risiko-/Aufwandsanalyse/Meilensteindefinitionen
 - Modellierungsanteile (konkrete Ausprägung abhängig vom Spieltyp)
 - Verhaltensdiagramme (primär Use-Case-, Aktivitäts-, Zustands- und Sequenzdiagramme)
 - Strukturdiagramme (primär Paket- und Klassendiagramme)
 - Implementierungsanteile
 - Wegwerf-Prototypen (abhängig vom konkreten Bedarf)
 - Vertikaler Prototyp (inklusive JavaDoc und JUnit-Tests)
 - Evaluationsanteile (in Form der Ergebnisse dokumentiert)
 - Überarbeitungen des *Game-Design*-Dokuments
 - Überarbeitungen der Planungsanteile
 - Re-Design der Modellierungs-/Refactoring der Code-Anteile

4 Die Umsetzung

Da es sich bei der Umsetzung um das Pilotprojekt eines neuen Praktikumkonzepts handelte, war nicht zu erwarten, dass die Planung in allen Aspekten der Realität entsprechen würde. Deswegen wurde von Anfang an ein hoher Grad an Flexibilität und Selbstständigkeit von Seiten der Teilnehmer mit eingeplant, was auch der Auslegung als ein fortgeschrittenes Praktikum im Bereich SE entsprach.

Dies erhöhte zwar deutlich die an die Studenten gestellten Anforderungen, erzeugte im Endeffekt aber genau das notwendige Level an Autonomie, um eine **wirkliche Identifikation** der Studenten mit dem jeweiligen Projekt zu erlauben [Fic05]. Um diesen Effekt noch zu verstärken, wurden (mit Auflagen) auch die Gruppeneinteilung und die Kontrolle über das *Game Design* an die Praktikums Teilnehmer übertragen. Weiterhin wurde den Projektteams bei den bereits aus dem Programmierpraktikum bekannten Anteilen Freiheiten in der Anwendung eingeräumt, die stichpunktartig überprüft wurden. Die Kontrolle der Arbeitsergebnisse erfolgte zudem weniger im Stil von Praktikumspräsentationen gegenüber der Praktikumsleitung, sondern eher im Verständnis von Zwischenstandsberichten gegenüber dem verantwortlichen Projektmanager, was zusätzlich die Illusion realer Spieleprojekte vertiefte. Nur die Beschränkung auf 2-D-Grafik verstand sich in Hinblick auf die Vorkenntnisse von selbst.

4.1 Der personelle Rahmen

Von den sich ursprünglich bewerbenden 30 (spieleaffinen) Studenten (ein typischer Informatik-Jahrgang umfasst an der Universität der Bundeswehr lediglich ca. 75 Studenten), wurden die 20 vielversprechendsten Teilnehmer ausgewählt, die sich in vier Teams aufteilten.

Innerhalb der Entwicklerteams waren zudem die speziellen Rollen (vgl. Abb. 1) des Teamchefs (*Developer Producer*), des *Game-Design*-Verantwortlichen (*Lead Designer*) und des Technikverantwortlichen (*Lead Engineer*) zu vergeben. Die Rolle der Praktikumsleitung war mit der eines Projektmanagers (*Publisher Producer*) gleichzusetzen. Zur Kompensation evtl. technischer Probleme existierte zudem die Rolle eines *Game Consultant*, die von einem studentischen Tutor übernommen wurde

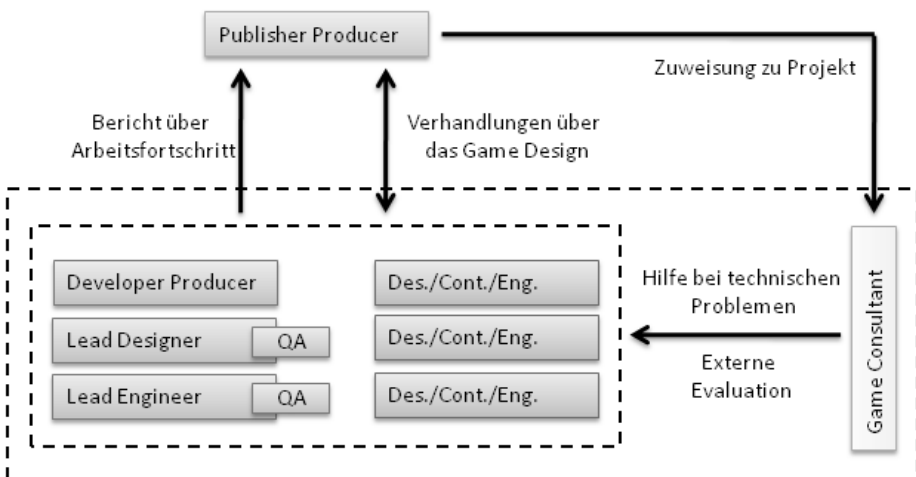


Abb. 1 Die Entwicklungsrollen

4.2 Der technische Rahmen

Gemäß dem vorliegenden Curriculum war Java als Sprache der Wahl gesetzt, was allerdings die Auswahl der verfügbaren Plattformen zur Spieleentwicklung erheblich einschränkte [Wal06]. Nach der Evaluation einer kleinen Anzahl möglicher Kandidaten fiel die Wahl auf das **Slick Game Framework**, eine relativ flexibel verwendbare Ansammlung einzelner Bibliotheken, die von einer Kernkomponente mit integrierter Spielschleife zusammengehalten werden. Zum abgebildeten Funktionsspektrum des Frameworks zählen z.B. eine einfache Ressourcen-Verwaltung und grafische Ausgabemöglichkeiten auf Sprite-Basis (vgl. hierzu auch [Tkd09]). Da es sich bei *Slick* allerdings nicht um ein Rahmenwerk mit Ausbildungshintergrund handelt, stand auch keine extensive Dokumentation zur Verfügung. Um dieses Problem etwas zu lindern, wurde die bereits erwähnte Rolle eines *Game Consultant* in das Praktikumskonzept aufgenommen, dessen Aufgabenfeld zum einen die Erstellung eines Tutorials zur Demonstration der Framework-Grundfunktionalität und zum anderen die technische Betreuung der Teams (z.B. durch einen Vortrag zum Framework) beinhaltete. Hinsichtlich der Entwicklungswerkzeuge waren die Versionsverwaltung SVN und die Entwicklungsumgebung *IBM Rational Systems Developer* vorgeschrieben.

4.3 Der Prozessrahmen

Organisatorisches

Entsprechend den üblichen Charakteristika eines Spieleentwicklungsprozesses zergliederte sich der Ablauf des Praktikums in drei grobe Abschnitte (vgl. Abb. 2): einer *Pre-Production*-Phase von vier Wochen Dauer, einer *Production*-Phase von acht Wochen Dauer und einer abschließenden *Post-Production*-Phase von zwei bis drei Tagen. Die erste Phase enthielt hierbei eine Konzeptions- (drei Iterationen à eine Woche) und eine Planungsphase (eine Woche). Die Produktionsphase beschränkte sich auf drei Iterationen des *Prototyping* (d.h. der technischen Umsetzung), eine *Full-Production*-Subphase war dagegen aufgrund der Vernachlässigung der Inhaltsanteile zunächst nicht vorgesehen. Zur Intensivierung des Spieleentwicklungskontexts wurden zu Beginn der Vorproduktion Vortragsanteile mit Informationen zum Spielmarkt und der zugehörigen Industrie integriert. Den Start- bzw. Endpunkt des Praktikums bildeten die Industrie-übliche *High-Concept*-Präsentation (die Vorstellung der Spielidee) respektive eine Marketing-orientierte Endpräsentation. Weiterhin wurden vor der jeweiligen Phase Informationsveranstaltungen zu ausgewählten *Game-Design*-, Planungs- und SE-Anteilen sowie zum *Slick*-Framework angeboten. Die Praktikumskommunikation erfolgte auf der Projektmanagementebene mit Hilfe einer Praktikumsseite inklusive »Hot News«-Anteil, Anleitungen/Hilfestellungen für die Entwicklung und einem Google-Kalender zur Planung. Die Projektseiten der Studententeams spiegelten

die SE-Ebene wider und beinhalteten sowohl den ständig aktuellen Stand des Projektfortschritts als auch alle externen Dokumentationsartefakte.

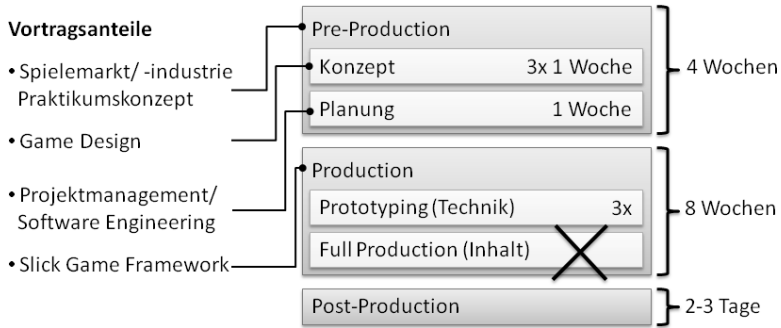


Abb. 2 Die Entwicklungsphasen

Alle übrigen Artefakte standen der Praktikumsleitung direkt über das Versionierungssystem zur Verfügung, was eine schnelle und effiziente Möglichkeit des Zugriffs erlaubte. Neben der eigenen Projektseite, die um übliche »Marketing-Anteile« bereichert werden durfte, wurden wöchentliche Präsentationen (60-90 Min.) vorgesehen. Diese dienten einerseits dazu, die erarbeiteten Anteile vorzustellen und zu besprechen, andererseits stand auf diese Weise aber auch jedem Team genügend Zeit zur Verfügung, Fragen zu klären bzw. Hilfestellungen bei Problemen zu erhalten. Zudem erhielten alle Teilnehmer aufgrund der noch spärlichen Dokumentation einen wöchentlichen »Praktikums-Newsletter«, der die Ergebnisse der vergangenen Woche besprach und Tipps und Anleitungen für die folgende Woche enthielt.

Pre-Production-Phase

Die Umsetzung der Konzeptphase, d.h. also der Erstellung eines *Game Design*, erschien zunächst als der schwierigste Anteil in Hinblick auf die Umsetzung des Praktikums, da die für eine zugehörige Einweisung zur Verfügung stehende Zeit als zu knapp bemessen erschien. Dies wurde noch dadurch verstärkt, dass eine Eingrenzung der Thematik durch die Wahl von vier komplett unterschiedlichen Spielekonzepten erheblich erschwert wurde. Aus diesem Grund wurde an dieser Stelle ein iterativer Ansatz gewählt. So umrissen die einleitenden Vortragsanteile nur die Grundlagen des *Game Design* und gaben eine Starthilfe für die ersten Schritte. Nun war es an den Studententeams, auf Basis der zur Verfügung stehenden Literatur ([Rol03] erwies sich hier als geeignete Wahl) eine erste Iteration der eigenen Spielidee zu erstellen. Diese Version wurde nun in der darauf folgenden Wochenpräsentation detailliert besprochen und der Weg zur nächsten Iteration anskizziert. Selbige Diskussionen erfolgten hierbei primär auf der Ebene von Genre-typischen Aspekten bzw. *Game-Design-Pattern* [Bjö05], was aufgrund der nicht zu

unterschätzenden Vorkenntnisse der Teilnehmer im Bereich der Computerspiele ohne Weiteres möglich war. Hierbei wurden von jeder Gruppe zunächst drei Initialkonzepte erstellt, von denen dann eines in Richtung eines *Game-Design*-Dokuments verfeinert wurde (selbiges enthielt sowohl die Kernidee des Spiels als auch alle narrativen Anteile und die zugehörige Spielmechanik). Das Dokument wurde wiederum als »lebendige« Form der Dokumentation verstanden, die bis zum Ende des Praktikums weitergepflegt und bei Bedarf angepasst wurde. In der sich anschließenden Planungsphase erfolgte nun zunächst die Extraktion inhaltlicher und technischer Anforderungen aus den *Features* des *Game-Design*-Dokuments sowie deren Priorisierung in Hinblick auf das *Game Play*.

Auf Basis der sich ergebenden Liste wurde eine erste Aufwands-/Risikoanalyse vorgenommen, die wiederum die Grundlage für die Formulierung der Meilensteine (der Ausbaustufen des vertikalen Prototypen) bildeten. Auch Aufwands-/Risikoanalyse wurden hier nur als »erster Wurf« verstanden und im weiteren Verlauf des Projekts mehrfach aktualisiert. Der fehlenden Erfahrung der Studenten wurde hier u.a. mit Hilfe einer relativen Schätzmethode – eines Planungspokers – begegnet.

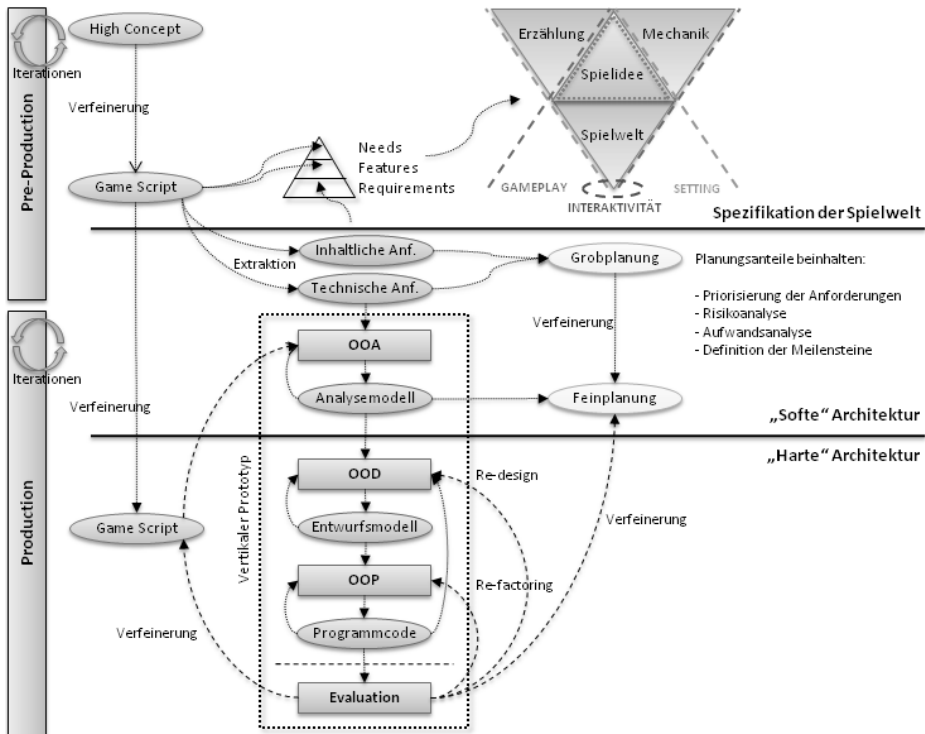


Abb. 3 Der Entwicklungsprozess

Production-Phase

Aufgrund der bestehenden Vorkenntnisse wurden den Projektteams nur geringfügige Vorgaben bzgl. des klassischen OOA-OOD-OOP-Zyklus gemacht. Allerdings ergaben sich einige domänenspezifische Eigenheiten, die im Folgenden besprochen werden: Während ursprünglich drei, rein technisch orientierte Prototyping-Iterationen vorgesehen waren, tendierte die Entwicklung (in der Realität dann zwei Iterationen) jedoch zum Ende hin teilweise in Richtung *Full Production*. Nur dadurch wurden die technischen Prototypen auch wirklich zu Spielprototypen. Nichtsdestotrotz verblieb der Fokus auf der technischen Entwicklung, die auf zweierlei Wege ihren Anfang fand: Zum einen erfolgte die Erstellung kleiner Wegwerf-Prototypen, deren Zweck das schlichte Ausloten des Fähigkeitsprofils des gewählten Frameworks war, wobei hier explizit darauf geachtet wurde, dass diese niederqualitativen Fragmente keinen direkten Eingang in die eigentliche Codebasis fanden (die Codequalität der Projekte wurde regelmäßig per SVN kontrolliert). Zum anderen fand nun eine genauere Analyse der zuvor extrahierten Anforderungen (des ersten Meilensteins) statt, die mit Hilfe unterschiedlicher Analysetechniken zu einem Domänenmodell zusammengeführt wurden. Neben einem modellgetriebenen Ansatz, primär in Form von Kontext- und »60 Sekunden *Game Play*« Use Cases [Fly04] sowie Klassen-/Zustandsdiagrammen, erwiesen sich auch ein weniger formales Storyboarding und (in Teilen) CRC-Kartensitzungen als hilfreich. Allerdings wurden in dieser Phase auch die qualitative Komplexität und Andersartigkeit der Domäne offensichtlich, was sich in klaren Lücken der Analysemodelle äußerte. Hierbei waren der für Computerspiele typische Kontrollfluss in Form einer Ereignisschleife und die zustandsbasierte Spielweltsimulation die primären Problemfelder.

Nach einer Verfeinerung der zugehörigen Planungsanteile wurde nun versucht, die beiden Anteile in Form eines objekt-orientierten Entwurfsmodells zusammenzubringen. Die modellbasierte Verschmelzung zwischen dem durch die Prototypen erarbeiteten Framework-Know-how und des in eine technische Sicht transferierten Domänenmodells beschränkte sich aber auch im Entwurf auf einige wenige Diagrammanteile. Viele der relevanten Aspekte konnten von den Projektteams erst durch Implementierung des vertikalen Prototypen umrissen werden. Da die Fokussierung auf solch evolutionäre Prototypen und die damit einhergehende Bottom-up-Entwicklung aber bekanntermaßen mit einem nicht unerheblichen Risiko behaftet sind, wurde im Rahmen des Praktikums auch ganz besonderer Wert auf die regelmäßige Überarbeitung der Codebasis und Anwendung von Entwurfsmustern [Gam01] gelegt. Hierbei wurde den Teilnehmern laut eigener Aussage die Sinnhaftigkeit selbiger Maßnahmen auch deutlich klarer als bei deren Verwendung im Rahmen des deutlich sequenzielleren Programmierpraktikums. Die gleiche Aussage gilt hierbei auch für den Bereich der Regressionstests, die konkrete Umsetzung auf Basis von JUnit fiel den Teilnehmern aufgrund der ungewohnten Domäne allerdings erkennbar schwer.

Post-Production-Phase

Nach zwei Durchläufen des zuvor beschriebenen *Prototyping* bestand die letzte Phase des Praktikums dann zum einen aus der Reflektion des umgesetzten Projekts aus Sicht des jeweiligen Projektteams, die – wie in der Spieleindustrie üblich – in Form eines *Post Mortem* umgesetzt wurde. Zum anderen erfolgten noch die Endpräsentation der Projekte und eine Abschlussbewertung im Gesamtrahmen.

5 Die Evaluation

Endergebnis des GE-Praktikums waren vier lauffähige Prototypen aus komplett unterschiedlichen Spiele-Genres (im Bereich von ~8.000 – 16.000 SLOCs): So wurde mit dem Projekt *AntZeit* (vgl. [Tkd09]) ein isometrisches Echtzeitstrategiespiel im Mikrokosmos der Ameisen umgesetzt. *Operation: DIPLOMA* dagegen vertrat das Genre der *Point&Click Adventures* auf Basis statischer Szenerien. Der Sidescroller *Auftrag Bachelor* realisierte ein klassisches *Jump&Run* mit Studienhintergrund. Das Rollenspiel *Tales of Lavankor* letztlich setzte in Top-down-Sicht eine Fantasy-Welt um. Der herausragendste Aspekt bei der Entwicklung der Spiele war hierbei der hohe Grad an Selbstständigkeit und Motivation der Teilnehmer, der bei zweien der Projektteams so weit ging, dass diese auch nach Abschluss des Praktikums die Entwicklung fortführten. So stellten der neue Anteil des *Game Design* und der erhöhte Schwierigkeitsgrad im Prozess und der technischen Umsetzung keine allzu hohen Hürden dar, wären allerdings ohne die bestehenden Vorkenntnisse im Bereich SE kaum zu realisieren gewesen. Aber auch die üblicherweise sehr unbeliebten Planungsanteile wurden sorgsam angegangen – man wollte ja das **eigene Projekt** im Griff behalten. In diesem Sinne kam in der Rolle des *Publisher Producer* auch eher das Gefühl einer wirklichen Projekt- denn Praktikumsbetreuung auf, was aber auch durch die entsprechende Zeitinvestition und Flexibilität erkaufte werden musste. Zudem war auch die Kreativität der Teilnehmer hinsichtlich der inhaltlichen Produktion etwas unterschätzt worden: Diese reichte letztlich von eigenen (einfachen) 3-D-Modellen bis hin zur Hintergrundmusik einer semiprofessionellen Rollenspielerband. In Hinblick auf die initiale Konzeption zeigte sich die geplante Dauer (2,5 Wochen) der drei angedachten Prototyping-Iterationen als zu kurz bemessen (dies wurde ad hoc auf zwei Iteration à vier Wochen angepasst). Die nicht eingeplante *Full-Production*-Phase fand in Teilen, parallel zur technischen Entwicklung und auf Basis von Eigeninitiative der Teams, doch statt. Bezüglich der reinen SE-Themen erwies sich der Analyseanteil als Herausforderung, was sich vor allem auf das noch spärliche Hilfsmaterial sowie fehlende Beispiele in Form von Vorgängerprojekten zurückführen ließ. Die mit fortschreitender Bearbeitung etwas vernachlässigten Entwurfsanteile sollten zukünftig stärker in Form einer speziellen Refactoring-/ Re-Design-Sitzung eingefordert werden. Letztlich könnte auch die Authentizität

des Ansatzes noch (relativ einfach) durch »Bewerbungen« der Projektteams mit dem angedachten *High Concept* und (deutlich aufwendiger) durch interdisziplinäre Teams gesteigert werden. Für letzteren Part würde sich die Kooperation mit externen Partnern aus Bereichen der inhaltlichen Produktion anbieten.

Zusammenfassend formuliert, wurden im Projektverlauf die in Abschnitt 3.2 aufgeführten Bewertungskriterien von allen Teams erfüllt, allerdings war eine vom Spieltyp abhängige, unterschiedliche Betonung von genretypischen Eigenheiten unerlässlich. Im Ergebnis erwies sich der Pilot des GE-Praktikums für beide Seiten als ein sehr zeitintensives und forderndes, aber auch äußerst lehrreiches und damit erfolgreiches Projekt im Rahmen der SE-Ausbildung, das unbedingt fortgeführt werden sollte.

Literatur

- [Bad00] Badcoe, I., 2000. The Computer Game Industry: Current State of Play. *Virtual Reality*, Volume 5, Number 4. Springer, London, 204-214
- [Bjö05] Björk, S. and Holopainen, J., 2005. *Patterns in Game Design*. Charles River Media, Boston, MA
- [Bur08] Burns, B., 2008. Teaching the Computer Science of Computer Games. *Journal of Computing Sciences in Colleges*, Volume 23, Issue 3. Consortium for Comp. Sciences in Colleges, USA, 154-161
- [Cal05] Callele, D., Neufeld, E., Schneider, K., 2005. Requirements Engineering and the Creative Process in the Video Game Industry. In *Proc. of the 2005 13th IEEE International Conference on Requirements Engineering*. IEEE, 240-252
- [Cha06] Chandler, H., 2006. *The Game Production Handbook*. Charles River Media, Inc., Hingham, MA
- [Dem99] Demuth, B. Hußmann, H. Zschaler, S. Schmitz, L., 1999. Erfahrungen mit einem frameworkbasierten Softwarepraktikum. In *SEUH'99*. Teubner, Wiesbaden, Germany
- [Dem05] Demuth, B. Schmitz, L. Wittek, B., 2005. Softwaretechnologie-Praktikum im Grundstudium: universitäres oder reales Projekt? In *SEUH'05*. Teubner, Wiesbaden, Germany
- [Fic05] Ficocelli, L., Gregg, D., 2005. A Group Project – with a twist! In *Proc. of the 2005 Western Canadian Conference on Computing Education (WCCCE)*. At <http://web.unbc.ca/wccce05/html/proceedings.html> (last accessed 20.10.08)
- [Fly04] Flynt, John P., Salem, O., 2004. *Software Engineering for Game Developers*. Thomson Course Technology, Boston, MA
- [Gam01] Gamma, E., Helm, R., Johnson, R., Vlissides, J., 2001. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley
- [Igd03] IGDA Education Committee, 2003. *IGDA Curriculum Framework: The Study of Games and Game Development*. Version 2.3 beta, Feb 25, 2003
- [Jtf04] The Joint Task Force on Computing Curricula, ACM, IEEE-CS, 2004. *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. Aug 23, 2004

- [Mcc04] McCallum, S., Mackie, J., Nacke, L., 2004. Creating a Computer Game Design Course. In Proc. of the New Zealand Game Developers Conference. NZGDC, Dunedin, New Zealand
- [Raj05] Rajagopalan, M., Schwartz, D.I., 2005. Game Design and Game-Development Education. *Phi Kappa Phi Forum*, Volume 85, Summer 2005, at <http://www.highbeam.com/doc/1G1-135022702.html> (last accessed 20.10.08)
- [Rez06] Rezk-Salama, C., et al., 2006. Game Development as Part of the Computer Science Education. In *Proc. International Digital Games Conference (iDiG)*. Springer, 15-24
- [Rol03] Rollings, A., Adams, E., 2003. *Andrew Rollings and Ernest Adams on Game Design*. New Riders Publishing, Berkeley, CA
- [Rol04] Rollings, A., Morris, D., 2004. *Game Architecture and Design: A New Edition*. New Riders Publishing, Berkeley, CA
- [Swe05] Sweedyk, E., Keller, R.M., 2005. Fun and Games: A New Software Engineering Course. *SIGCSE Bulletin*, Volume 37, Number 3. ACM, NY, 138-142
- [Tkd09] Team KAROSHI [Dev]. Volk, D., 2009. Das Projekt AntZeit. In *SEUH'09*. dpunkt.verlag, Heidelberg, Germany
- [Wal06] Wallace, S.A., Nierman, A., 2006. Addressing the Need for a Java Based Game Curriculum. *Journal of Computing Sciences in Colleges*, Volume 22, Issue 2. Consortium for Computing Sciences in Colleges, USA, 20-26