

Projektmanagement spielend lernen

Alexander Nassal, Institut für Programmiermethodik und Compilerbau, Universität Ulm

alexander.nassal@uni-ulm.de

Zusammenfassung

Um den Studierenden schon früh ein Gefühl für das Projektmanagement und den damit verbundenen Aufgaben und Probleme zu vermitteln, haben wir ein Spiel entwickelt, das auf dem Projektmanagementwerkzeug MS Project basiert. Unter Verwendung einer umfangreichen Simulation können damit die in MS Project erstellten Projektpläne simuliert werden. Die Studierenden können dadurch sofort erkennen, wie gut ihre Planung ist und wo Probleme auftreten. Ziel dabei ist es, den Studierenden eine Möglichkeit anzubieten, spielerisch und ohne Risiko verschiedene Projektmanagementstrategien unter verschiedenen Bedingungen auszuprobieren. Dabei können die Spieler alle in MS Project vorhandenen Funktionen und Werkzeuge nutzen.

In diesem Beitrag beschreiben wir das Spiel in seinem Aufbau und Ablauf. Um unsere Arbeit besser bewerten zu können, haben wir das Spiel mittels einer Befragung evaluiert. Eine kurze Zusammenfassung dieser Evaluation findet sich am Ende dieses Beitrags.

Motivation

Die Lehre im Bereich der Softwaretechnik, insbesondere im Bereich des Projektmanagements, stellt die Lehrenden vor die große Herausforderung, den Studierenden dieses wichtige Thema verständlich und nachvollziehbar zu vermitteln. Neben der Kenntnis der verschiedenen Methoden und ihren Einsatzfeldern ist es vor allem die Erfahrung, die einen guten Softwareingenieur ausmacht. Diese lässt sich jedoch nicht durch Theorie im Rahmen einer Vorlesung vermitteln, sondern setzt persönliche Aktivität der Lernenden voraus.

Vor allem im Bereich des Projektmanagements ist es aufwändig, den Studierenden dafür ein entsprechendes Übungsangebot zu machen. Ziel des Übungsangebots sollte es sein, den Studierenden die komplexen Zusammenhänge in einem Projekt zu verdeutlichen und es ihnen zu ermöglichen ein Gefühl für die Probleme und Schwierigkeiten im Gesamten zu entwickeln.

Im Gegensatz zu technischeren Fächern lassen sich für Projektmanagement nur schwer praxisnahe Beispiele und Übungen finden, an denen die Teilnehmer das zuvor erworbene theoretische Wissen selbst ausprobieren können. Ein reales Projekt benötigt neben dem organisatorischen Aufwand vor allem viel Zeit und zusätzliche Arbeitskapazität zur eigentlichen,

vom Projektmanagement unabhängigen Bearbeitung der Projektkinhalte. Daher sind solche Projekte als Grundlage einer Übung im Bereich des Projektmanagements nur sehr bedingt geeignet.

Ein möglicher Lösungsansatz ist es, Ausschnitte aus einem fiktiven oder realen Projekt zu verwenden und diese Teilprobleme als Übung zu bearbeiten. Da viele Aspekte sich nicht nur auf einzelne Teile sondern auf das ganze Projekt beziehen, lässt sich damit allerdings nicht alles abdecken.

Da Projektmanagement viel mit Mitarbeiterführung und den damit verbundenen weichen Faktoren zu tun hat, ist es oft schwierig konkrete Regeln oder Handlungsvorgaben für einzelne Situationen anzugeben. Die Studierenden müssen lernen, entsprechende Situationen zu erkennen, zu bewerten, zu lösen und dabei das Projekt als Gesamtes zu berücksichtigen. Dabei geht es nicht nur um das zu erstellende Produkt, sondern auch um die beteiligten Mitarbeiter und weitere Aspekte, wie beispielsweise die Marktrepputation des Unternehmens.

Als einen ersten Ansatz zur Bewältigung der oben beschriebenen Problematik haben wir das hier vorgestellte Planspiel entwickelt. Dazu haben wir die eigentliche, sonst aufwändige und teure Projektarbeit als Simulation in das Projektmanagementwerkzeug *Microsoft Project* (Microsoft Corporation, 2013) von Microsoft (im Folgenden als *MS Project* bezeichnet) integriert. Damit ermöglichen wir den Studierenden, das in MS Project geplante Projekt auch tatsächlich durchzuführen und durch geeignetes Feedback des Spiels zu sehen, wie erfolgreich die eigene Planung war. Im Gegensatz zu einem realen Projekt können dabei problemlos und in kurzer Zeit verschiedene Lösungsansätze ausprobiert und verglichen werden.

Durch die Integration des Spiels in eine umfangreiche und weit verbreitete Software stehen den Spielern viele ausgereifte Werkzeuge der modernen Projektplanung zur Verfügung, ohne dass diese speziell für das Spiel entwickelt werden müssen. Den Spielern ist es so beispielsweise möglich, die in MS Project vorhandene Ressourcenplanung zu verwenden, um zu sehen, wie gut die einzelnen Mitarbeiter ausgelastet sind, und um schnell zu erkennen, ob einzelnen Mitarbeitern zu viele Aufgaben aufgetragen wurden. Als positiven Nebeneffekt lernen die Studierenden dabei auch den Umgang mit MS Project.

Dieser Beitrag ist wie folgt strukturiert: Nach einem kurzen Überblick über bestehende Ansätze und verwandte Arbeiten folgt eine Beschreibung der Plattform und des verwendeten Simulationsframeworks. Der Hauptteil beschreibt den Aufbau und Ablauf des Spiels und liefert ein mögliches Beispielszenario, auf dem die anschließende Evaluation basiert. Zum Schluss bilden wir ein kurzes Fazit und diskutieren mögliche Verbesserungen und Erweiterungen, die bislang noch nicht umgesetzt wurden.

Verwandte Arbeiten

Vorhandene Arbeiten zeigen, dass die Idee, Simulationen und Spiele in der Softwaretechniklehre einzusetzen, nicht neu ist. Die meisten Projekte fokussieren dabei einen bestimmten Schwerpunkt. So konzentrieren sich beispielsweise *SimjavaSP* (Shaw u. Dermoudy, 2005) auf Wasserfall- und Spiralmodell und *SimVBSE* (Jain u. Boehm, 2006) auf die Thematik des Value-Based Engineering.

Durch eine Storyline und zusätzliche Spielinhalte kann der Spielspaß entsprechend erhöht werden, wie *The Incredible Manager* (de Oliveira Barros u. a., 2006) zeigt. Es gab in der Vergangenheit auch Versuche, Lernaspekte in bestehende Spiele wie beispielsweise *Second Live* zu integrieren (Ye u. a., 2007).

Andere Projekte verfolgen den Ansatz, eine Plattform zur Verfügung zu stellen, mit der eigene Lerninhalte selbst gestaltet und – meist in Form eines Simulationsmodells und zusätzlichem Inhalt für die geeignete textuelle und graphische Präsentation – zu einem Spiel zusammengesetzt werden können. Eines der ersten Projekte dieser Kategorie war das *SESAM* Projekt (Ludewig u. a., 1992) mit einem Schwerpunkt auf dem Qualitätssicherungsaspekt im Softwareentwicklungsprozess. Das umfangreichste uns bekannte Projekt auf diesem Gebiet ist *SimSE* (Navarro, 2006) der University of California, Irvine, das neben vielen Beispielszenarien für die unterschiedlichsten Vorgehensmodelle auch umfangreiche Editoren zur Erstellung eigener Inhalte bereitstellt.

Alle oben genannten Arbeiten versuchen einen oder mehrere Schwerpunkte des Software Engineering mit Hilfe einer eigens dafür entwickelten Spielumgebung zu vermitteln. Dabei können Schwierigkeitsgrad und benötigte Hilfestellungen gut an die Vorkenntnisse und Bedürfnisse des Spielers angepasst werden. Durch die künstlich geschaffene Spielumgebung wirkt ein solches Spiel aber oft aufgesetzte und realitätsfern. Dieser Effekt kann durch ein einfach gehaltenes Simulationsmodell verstärkt werden, das sich auf den Kern des zu vermittelnden Schwerpunkts beschränkt und keine Randeffekte berücksichtigt.

In der Arbeit *Alles nur Spielerei? Neue Ansätze für digitales spielbasiertes Lernen von Softwareprozessen* (Pieper, 2013) werden die oben genannten Projekte anhand verschiedener Kriterien verglichen. Die Arbeit

gibt außerdem einen tieferen Einblick in die Vor- und Nachteile der einzelnen Spiele und zeigt ungenutztes Potential auf.

Einer von Piepers Kritikpunkten ist die oft fehlende Integration von Werkzeugen aus der realen Softwareentwicklung in die Simulationsspiele. Für unser Projekt haben wir den umgekehrten Weg gewählt und die Simulation direkt in das Werkzeug MS Project integriert, um den Aspekt des Projektmanagements, insbesondere der Aufgabenplanung, für die Studenten erfahrbar zu machen.

Plattform und Simulationsframework

Um die Simulation nahtlos in MS Project zu integrieren, wurde das Spiel als AddIn unter Verwendung der Office-API entwickelt. Nach der Installation des AddIns stehen damit alle Funktionen direkt in MS Project zur Verfügung, weitere Werkzeuge werden nicht benötigt. Das AddIn kann über die API auf nahezu alle Funktionen und Daten von MS Project zugreifen und somit die Verbindung zwischen MS Project und dem Simulationsframework herstellen.

Um eine umfassende und realitätsnahe Simulation der Vorgänge in einem Projekt durchführen zu können, haben wir das in (Nassal, 2014) vorgestellte Framework für Planspiele im Bereich des Softwareengineering verwendet. Dieses Framework wurde speziell für den Einsatz in Lernspielen im Bereich der Softwaretechniklehre entwickelt und enthält neben den notwendigen Simulationsmodellen alle weiteren Elemente, die wir für die Simulation in unserem Spiel benötigen.

Das Simulationsmodell des Frameworks geht deutlich über die einfachen Berechnungen hinaus, wie sie in einigen vergleichbaren Spielen zu finden sind. Stattdessen wurden Modelle basierend auf Erkenntnissen der Arbeits- und Lernpsychologie verwendet, und zu einem komplexeren Gesamtmodell kombiniert. So werden beispielsweise die charakterlichen Eigenschaften der Mitarbeiter, ihr Lernverhalten, ihr Gesundheitszustand und ihre Belastbarkeit berücksichtigt. Ein eigenes Modell für die Motivation regelt die Arbeitsleistung und das Verhalten der Mitarbeiter. Der Projektplan wird dabei lediglich als Vorgabe verwendet, das tatsächliche Verhalten der Mitarbeiter kann je nach Situation entsprechend davon abweichen. Das spiegelt die Realität besser wider, als wenn der Projektplan strikt nach Vorgabe abgearbeitet wird.

Durch die Verwendung eines detaillierten Modells wird es für den Spieler notwendig, sich neben der Aufwandsschätzung, Abhängigkeits- und Terminplanung auch Gedanken über den richtigen Einsatz seiner Mitarbeiter zu machen, um ein vorgegebenes Problem erfolgreich lösen zu können.

Das Framework arbeitet auf einer Datenstruktur, die das Vorgehensmodell, die Rahmenbedingungen des

Projekts, die dazugehörigen Mitarbeiter mit ihrem jeweiligen Charakter und weiteren Eigenschaften, sowie das zu entwickelnde Produkt festlegt. Die Datenstruktur wird dabei in Teilen von der Simulation bei deren Ausführung fortgeschrieben. Des Weiteren kann diese Datenstruktur auch während der Simulation verändert werden, um beispielsweise besondere Ereignisse wie den Ausfall eines Mitarbeiters oder veränderte Anforderungen an das Produkt zu realisieren.

Das Vorgehensmodell legt fest, welche Aktivitäten unter welchen Bedingungen durchgeführt werden können. Als Basisaktivitäten stehen *Entwicklungsarbeit*, *Qualitätsarbeit* und *Kommunikation* zur Verfügung. Jeder Aktivitätstyp wirkt sich unterschiedlich auf das zu entwickelnde Produkt und die Mitarbeiter aus. Entwicklungsarbeit bewirkt einen Fortschritt bei den Produktartefakten. Qualitätsarbeit findet und behebt Fehler, die durch Entwicklungsarbeit in den Artefakten entstanden sind und erhöht damit die Qualität der Artefakte. Kommunikationsarbeit hat keinen Einfluss auf das Produkt, sondern dient dazu, Wissen zwischen den einzelnen Mitarbeitern oder Gruppen von Mitarbeitern auszutauschen.

Wir erzeugen die benötigte Datenstruktur aus einem Szenario, das über das AddIn geladen werden kann. So können ohne großen zusätzlichen Aufwand unterschiedlichste Szenarien für unser Spiel erstellt werden, ohne das AddIn anpassen zu müssen.

Let's play

Im Folgenden wird das Spiel in seinem Aufbau und Ablauf beschrieben und die Entwurfsentscheidungen, die dazu geführt haben, erläutert.

Da ein Wettbewerb zwischen den Studierenden oft eine gute Motivation ist, haben wir für das Spiel ein Score entwickelt, der den Projekterfolg misst und vergleichbar macht. Er liefert außerdem ein gutes Feedback für die Studierenden und ermöglicht es ihnen, ihr Vorgehen besser zu bewerten. Der Aufbau dieses Score wird im Anschluss an die Beschreibung des Spiels erläutert.

Am Ende dieses Abschnitts steht ein Beispielszenario, das den Spielbau noch einmal verdeutlichen soll und außerdem als Grundlage für die Evaluation des Spiels gedient hat.

Spielbau und Ablauf

Zu Beginn eines Spiels muss ein vorgefertigtes Szenario geladen werden. Ein Szenario enthält neben den Simulationsparametern und den Rahmenbedingungen des Projekts das zu verwendende Vorgehensmodell, welches wiederum die möglichen Aktivitäten, Rollen und Artefakttypen vorgibt. Es enthält außerdem das zu entwickelnde Produkt und damit die Menge der Artefakte, die während des Spiels erstellt werden müssen.

Je nach Lernziel kann im Szenario ein Projektteam definiert sein, mit dem der Spieler das Spiel beginnt.

Sind keine Mitarbeiter vorgegeben, muss der Spieler sich diese über den Arbeitsmarkt zusammenstellen.

Der Arbeitsmarkt wird ebenfalls über das Szenario konfiguriert. Hier können Parameter wie Durchschnittsgehalt, Durchschnittsfähigkeiten, die Anzahl der verfügbaren Arbeitskräfte und Ähnliches festgelegt werden. Der Arbeitsmarkt kann auch vollständig deaktiviert werden. In diesem Fall ist der Spieler auf den vorgegebenen Mitarbeiterstamm beschränkt und kann diesen nicht verändern.

Eines der Lernziele des Spiels ist es, dem Spieler ein bestimmtes Vorgehensmodell und die damit verbundenen Aktivitäten und Rollen zu vermitteln. Das Vorgehensmodell ist im Szenario definiert und kann vom Spieler im Spiel nachgeschlagen werden. Dafür kann im Szenario neben den Parametern der Aktivitäten und Rollen jeweils ein Beschreibungs- und Erklärungstext hinterlegt werden. Zusätzlich zu den Texten werden auch alle relevanten Abhängigkeiten angezeigt. Dargestellt werden beispielsweise die Artefakttypen, die mittels einer bestimmten Aktivität bearbeitet werden können, die Rollen, die für eine Aktivität zuständig sind und weitere Artefakttypen, die einen potentiellen Einfluss auf die Bearbeitung haben können, falls das im Produkt entsprechend definiert wurde.

Abbildung 1 zeigt die *Entwurfsaktivität* des weiter unten beschriebenen Beispielszenarios. Bearbeitet werden können hier sowohl Artefakte mit dem Typ *Entwurfsdokument*, als auch Artefakte mit dem Typ *UI Spezifikation*. Außerdem ist zu erkennen, dass es sich um eine Entwicklungsaktivität handelt und damit ein Dokument neu erstellt oder weiterentwickelt wird.

Analog zur Beschreibung des Vorgehensmodells kann der Spieler auch eine Beschreibung des Produkts einsehen. Hier werden alle Teilartefakte des Produkts aufgelistet und beschrieben.

Neben dem Beschreibungstext – der erklärt, um was für ein Dokument es sich handelt, für was es verwendet wird und wie es erstellt werden kann – ist auch der Artefakttyp angeben, der den Bezug zum Vorgehensmodell herstellt.

Weiterhin sind die im Szenario definierten Kontextartefakte und etwaige Voraussetzungen für die Bearbeitung aufgelistet. Ein Kontextartefakt ist ein anderes Artefakt, dessen Vollständigkeit und Qualität einen Einfluss auf die Erstellung des neuen Artefakts hat. So hat beispielsweise der Architektorentwurf einen Einfluss auf die Implementierung. Eine Voraussetzung ist eine Bedingung, die vorgibt, welchen Zustand andere Artefakte haben müssen, damit das neue Artefakt bearbeitet werden kann. So können logische und zeitliche Abfolgen in der Entwicklung sichergestellt werden. Beispielsweise kann Quellcode erst dokumentiert werden nachdem er erstellt wurde.

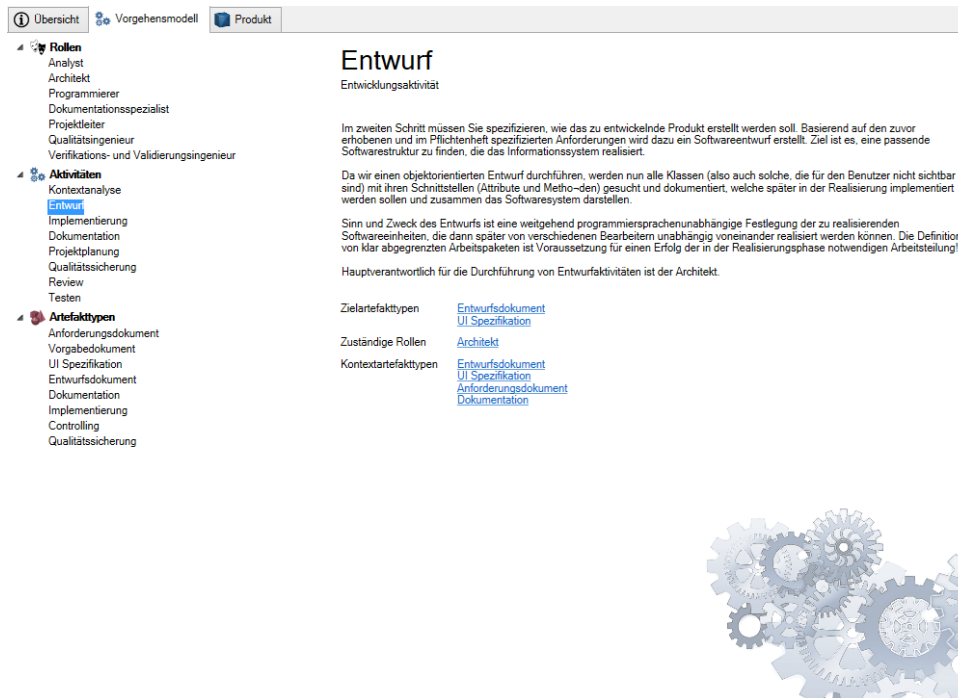


Abbildung 1: Erläuterung zur Entwurfsaktivität in der Szenariobeschreibung

Abbildung 2 zeigt das Produktartefakt *Dialoggestaltung* aus dem unten beschriebenen Beispielszenario. Es liefert dem Spieler alle Informationen, die notwendig sind, um das zu entwickelnde Artefakt zu verstehen und es in den Produkt- und Projektkontext einordnen zu können. Dazu werden neben einem beschreibenden Text auch die relevanten Kontextartefakte des Produktartefakts, sowie die für die Bearbeitung notwendigen Voraussetzungen und relevanten Fachbereiche aufgeführt.

Vorgangsplanung

Der Kern des Spiels ist der Projektplan, anhand dessen die Mitarbeiter entscheiden, wann sie welche Tätigkeit durchführen. Die simulierten Mitarbeiter entscheiden hauptsächlich anhand dieses Plans, welche Basisaktionen sie in der ihnen vorhandenen Zeit durchführen.

Den Projektplan erstellen die Spieler mit den in MS Project vorhandenen Werkzeugen. Das Ergebnis ist die Menge der Vorgänge, welche in MS Project typischerweise als Gantt-Diagramm dargestellt wird (Abbildung 3).

Das AddIn wandelt die Vorgänge aus MS Project in simulationsinterne Aufgaben des Frameworks um. Ein Vorgang in MS Project besitzt folgende für die Simulation wichtige Attribute:

- Einen Start- und Endzeitpunkt
- Eine Tätigkeit die ausgeführt werden soll
- Ein Artefakt auf dem die Tätigkeit ausgeführt werden soll
- Einen oder mehrere Mitarbeiter, die dem Vorgang zugewiesen wurden

Tätigkeit und Artefakt können dabei aus einer vorgegebenen Liste ausgewählt werden, die über das Szenario festgelegt ist. Mitarbeiter werden einem Vorgang, wie in MS Project üblich, als Ressource zugeordnet. Das erlaubt es, die in MS Project vorhandenen Werkzeuge und Ansichten zur Ressourcenplanung zu nutzen, um damit leichter eine optimale Auslastung zu erreichen und eine Überlastung der Mitarbeiter zu verhindern.

Neben der Unterstützung zur Ressourcenplanung können auch alle anderen in MS Project integrierten Werkzeuge zur Vorgangsplanung verwendet werden. Dazu gehören unter anderem die zeitlichen Abhängigkeiten der einzelnen Vorgänge, sowie deren Gruppierung und Hierarchisierung. Auch Hilfsmittel wie Beschriftung, Einfärbung und Kommentierung von Vorgängen stehen zur Verfügung.

Die Steuerung der Simulation ist sehr einfach. Der in MS Project vorhandene Cursor, der das aktuelle Datum in einem Projektplan anzeigt und somit Vergangenheit und Zukunft voneinander trennt, markiert den aktuellen Zeitpunkt der Simulation. Wird ein neues Zeitintervall simuliert, läuft der Cursor bis zu diesem Zeitpunkt vor.

Die Simulation wird über das Simulationsribbon (siehe Abbildung 4) in MS Project gesteuert. Neben der Möglichkeit ein neues Szenario zu laden, bzw. das geladene Szenario auf die Ausgangssituation zurückzusetzen, beinhaltet es verschiedene Möglichkeiten die Simulation bis zu einem gewünschten Zeitpunkt auszuführen. Dabei werden die vom Spieler festgelegten Vorgänge verwendet, um den Fortschritt und

Übersicht
Vorgehensmodell
Produkt

Artefakte

- Produktskizze
- Kundengespräch
- Überblick
- Fachwissen
- Anwendungsfälle
- Szenarien
- Systemaufgaben
- Nichtfunktionale Anforderungen
- Dialogstruktur
- Dialoggestaltung
- Nutzungskonzept
- Systemarchitektur Teilentwurf (Kommunikationsdiagramm)
- Systemarchitektur Teilentwurf (Klassendiagramm)
- Systemarchitektur Teilentwurf (Methodenbeschreibung)
- Durchstich Teilentwurf
- Datenbanklayout Teilentwurf
- Systemarchitektur Teilentwurf (Kommunikationsdiagramm)
- Systemarchitektur Teilentwurf (Klassendiagramm)
- Systemarchitektur Teilentwurf (Methodenbeschreibung)
- Rollenverteilung
- Projektplan
- Programmierkonventionen
- Systemkern
- Testplan Systemkern
- Tools
- Testplan Tools
- Datenbankanbindung
- Testplan Datenbankanbindung
- Benutzeroberfläche Komponente I
- Testplan Benutzeroberfläche Komponente I
- Logik Komponente I
- Testplan Logik Komponente I

Dialoggestaltung

[UI Spezifikation](#)

In der Dialoggestaltung werden die einzelnen Dialoge genauer spezifiziert. Dazu wird zu jedem Dialog ein Prototy (bspw. als MockUp) erstellt und zusätzlich textuell beschrieben. Dazu gehören auch die Interaktionen (mit ihren Vor- und Nachbedingungen), die der Benutzer auf dem Dialog ausführen kann, sowie diverse Eigenschaften des Dialogs (modal, nichtmodal, etc.).

Vorgehensweise bei der Dialoggestaltung

- Beschreiben Sie den konzeptionellen Aufbau und die Interaktionsmöglichkeiten aller Dialoge
- Erstellen Sie einen GUI-Prototypen, indem Sie für jeden Dialog seine grafische Repräsentation (Komponenten inklusive Beispielinhalt) erstellen. (Für ähnliche Dialoge (z.B. Ok/Abbrechen oder Ja/Nein Dialoge) müssen Sie nur jeweils einen Screenshot erstellen. Ggf. führen Sie die unterschiedlichen Meldungstexte dazu auf.
- Die GUI-Prototypen sind MockUps, also nur Entwürfe ohne echte Funktionalität. Verwenden Sie ein entsprechendes Werkzeug zur Erstellung (z.B. MS Visio oder entsprechende GUI-Builder). Verzichteten Sie auf Styles und andere Gestaltungsmerkmale, die keine Informationsträger sind (Schatten, runde Ecken, etc.). Sie können zusätzlich Stylevorschläge erstellen (hier reicht meistens schon ein einziger Dialog aus).
- Beschreiben Sie die Benutzerinteraktionen, die von den Dialogen unterstützt werden, im Stil einer Benutzeranleitung. Interaktionen sollten dabei möglichst abstrakt bleiben und nicht bis zu einzelnen Mausclicks oder Tastendruckern verfeinert werden.
- Dokumentieren Sie sowohl die Interaktion selbst, also die konkreten Aktionen eines Benutzers, der die Interaktion durchführen will, als auch die ausgelagerte Aktivität des Systems oder den durchgeführten Dialogwechsel.
- Beschreiben Sie alle Ausgabedokumente (Ausdrucke oder Export-Daten).
- Prüfen ob Ihre Lösung die Richtlinien der Software-Ergonomie (ISO 9241) erfüllt.

Relevante Fachbereiche [Gestaltung](#)

Kontextartefakte [Dialogstruktur](#)
[Nichtfunktionale Anforderungen](#)
[Szenarien](#)

Voraussetzungen [Dialogstruktur](#)

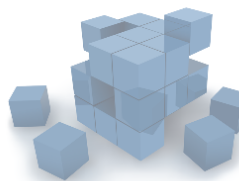


Abbildung 2: Erläuterung zur Dialoggestaltung in der Szenariobeschreibung

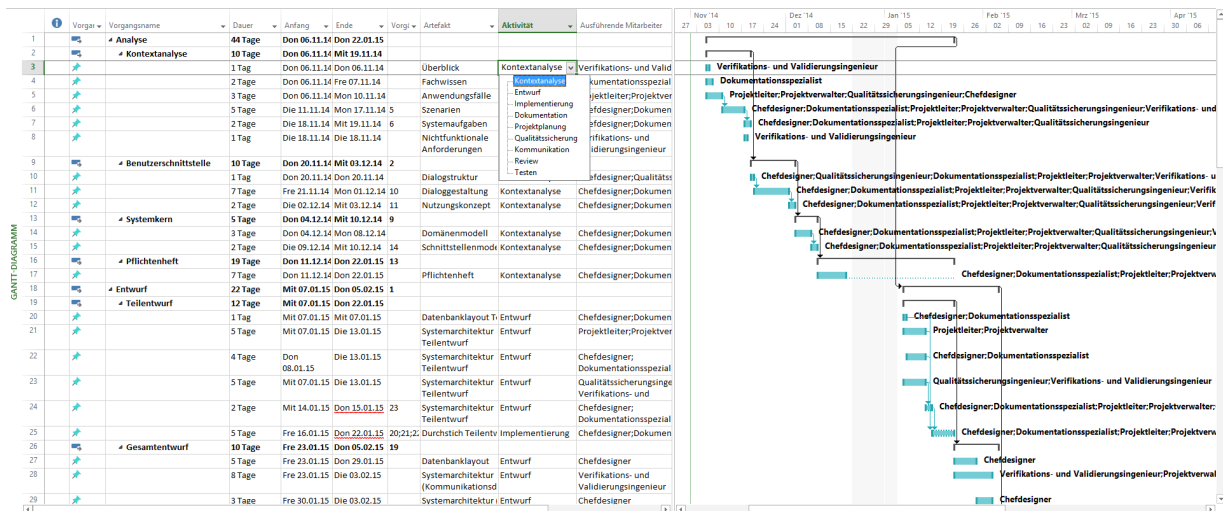


Abbildung 3: Vorgangsplanung in MS Project

die Qualität der Artefakte entsprechend des Simulationsmodells und der Parameter, welche im Szenario festgelegt wurden, zu aktualisieren.

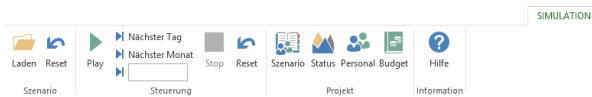


Abbildung 4: Steuerelemente der Simulation

Um dem Spieler zu ermöglichen, aus seinen Fehlern zu lernen, diese zu korrigieren und die Auswirkungen der Korrekturen zu sehen, kann die Simulation auf den Anfangszustand zurückgesetzt werden. Geplante Vorgänge bleiben dabei, im Gegensatz zum Zurücksetzen des Szenarios, erhalten. Merkt der Spieler beispielsweise, dass er seine Mitarbeiter in

der Vergangenheit besser hätte auslasten müssen, oder dass er einige Vorgänge besser in einer anderen Reihenfolge hätte planen sollen, kann er diese Fehler korrigieren und die Simulation erneut von Beginn an ausführen. Da sich die Simulation bei mehrfachem Ausführen jeweils gleich verhält, ist das Springen zu einem bestimmten Punkt in der Vergangenheit als zusätzliche Option nicht notwendig. Statt dessen kann die Simulation auf den Anfang zurückgesetzt und dann erneut bis zu dem gewünschten Zeitpunkt ausgeführt werden.

Personalmanagement

Neben der Reihenfolge der Vorgänge und ihrer Dauer, hat auch die Auswahl der Mitarbeiter einen erheblichen Einfluss auf das Ergebnis. Jedem einzelnen

Mitarbeiter kann über eine Vielzahl verschiedener Parameter ein ganz individueller Charakter zugewiesen werden. Dieser Charakter bestimmt das Arbeits- und Lernverhalten, die soziale Interaktion mit der Umwelt und die Fähigkeiten in Bezug auf das Arbeitsfeld eines Mitarbeiters. Die Werte der Parameter können entweder im Szenario definiert, oder von der Simulation während des Spiels zufällig gewählt werden.

Im realen Leben sind diese Parameter nur schwer messbar und selten in irgendeiner Art direkt sichtbar. Ein Projektleiter muss deshalb seine Mitarbeiter kennen und ein Gefühl dafür entwickeln, wie er sie entsprechend ihres Charakters am besten einsetzt. Da in einem Spiel diese Art des Kennenlernens nur schwer umsetzbar ist, müssen wir dem Spieler entsprechende Hilfestellungen geben. Das können beispielsweise zusätzliche beschreibende Texte sein, die im Szenario enthalten sind. Eine weitere Möglichkeit ist das Anzeigen einiger ausgewählter Eigenschaften der Mitarbeiter. Für unser Spiel haben wir eine Kombination aus beiden Wegen gewählt. Da die Fähigkeiten der Mitarbeiter einen hohen Einfluss auf die Produktivität und die Qualität der Arbeit haben, kann zu jedem Mitarbeiter das Fähigkeitsprofil angeschaut werden.

Abbildung 5 zeigt die Mitarbeiterverwaltung. Bei Auswahl eines Mitarbeiters werden rechts dessen Fähigkeiten zu den einzelnen Aufgaben und den produktspezifisch relevanten Domänen als Balkendiagramme angezeigt. Zusätzlich wird rechts oben durch ein Icon ein Hinweis auf den gesundheitlichen Zustand des Mitarbeiters eingeblendet. So kann der Spieler erkennen, wie fit seine Mitarbeiter sind, und diese Information bei der Planung der Arbeitspakete berücksichtigen.

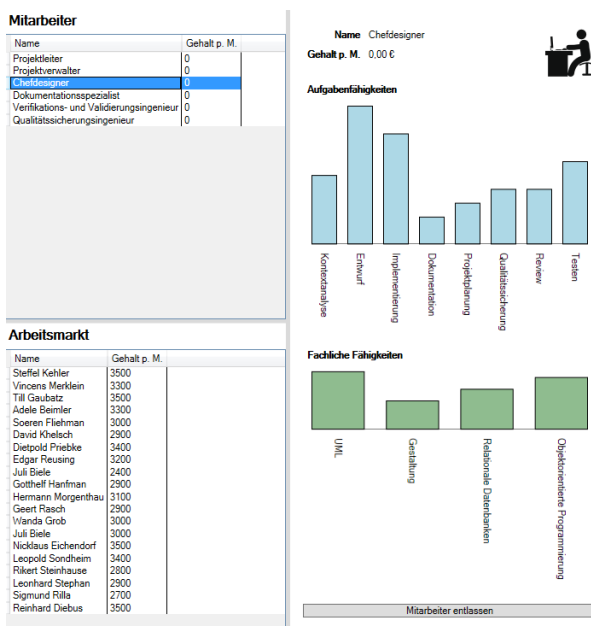


Abbildung 5: Mitarbeiterübersicht

Wird ein Bewerber auf dem Arbeitsmarkt ausgewählt, werden statt den realen Werten die Werte aus dessen Portfolio angezeigt. Das Portfolio eines Mitarbeiters entspricht einer Bewerbung, in der der Mitarbeiter sich selbst beschreibt. Diese Beschreibung kann je nach Charakter des Mitarbeiters entsprechend von den tatsächlichen Werten abweichen. So kann ein Mitarbeiter nicht in der Lage sein, seine Fähigkeiten entsprechend gut einzuschätzen, oder er stellt sich absichtlich besser oder schlechter dar, als er tatsächlich ist. Diese Diskrepanz zeigt sich erst nach der Einstellung des Mitarbeiters. Da eine Einstellung immer mit einem finanziellen Aufwand und einer zeitlichen Verzögerung verbunden ist, kann der Spieler nicht einfach alle Mitarbeiter ausprobieren. Es besteht deshalb – wie auch in der Realität – bei der Einstellung eines neuen Mitarbeiters immer ein gewisses Risiko, einen ungeeigneten Mitarbeiter einzustellen.

Budget und Finanzen

Ein relevanter Teilaspekt, der zunächst nichts mit der Softwareentwicklung an sich zu tun hat, aber in nahezu jedem Projekt berücksichtigt werden muss, sind die Projektfinanzen. Vor allem bei Szenarien, in denen der Spieler seine Mitarbeiter selbst über den Arbeitsmarkt bezieht, sind die dadurch entstehenden Kosten relevant und wirken sich teils erheblich auf den Projekterfolg aus.

Jeder Mitarbeiter kostet Geld. Wie viel wird entweder im Szenario festgelegt oder durch den Arbeitsmarkt definiert. Bessere Mitarbeiter kosten tendenziell mehr als Mitarbeiter mit geringeren Fähigkeiten. Es gibt dabei allerdings auch Ausnahmen. Der Spieler muss lernen, die Fähigkeiten der Kandidaten in Bezug auf sein Projekt zu beurteilen, um so die richtigen Mitarbeiter auswählen zu können. Dabei muss er berücksichtigen, welche Vorteile ein Mitarbeiter für das Projekt hat, aber auch welche Kosten dabei entstehen. In manchen Fällen kann es sinnvoll sein, zwei günstigere Mitarbeiter einzustellen als einen teureren, oder umgekehrt. Der Spieler muss dabei auch berücksichtigen, dass durch die Kommunikation zwischen den Mitarbeitern Arbeitsleistung verloren gehen kann.

Mitarbeiter, die nicht ausgelastet sind, verursachen trotzdem Kosten. Daher muss der Spieler gut überlegen, wann er zusätzliche Mitarbeiter einstellt und wann er Mitarbeiter entlässt. Sowohl Einstellung als auch Entlassungen verursachen zusätzliche Kosten.

Neben dem Gehalt der Mitarbeiter, das sich von Mitarbeiter zu Mitarbeiter unterscheiden kann und mitunter von deren Tagesarbeitszeit abhängig ist, haben Mitarbeiter auch davon unabhängige, laufende Fixkosten. Zusätzlich gibt es weitere, laufende Fixkosten für das Projekt an sich, z.B. für Büroräume, Softwarelizenzen und Büromaterial. Zusätzliche, einmalige Kosten, ebenso wie zusätzliche Einnahmen, können im Szenario festgelegt werden. Zusätzliche

Einnahmen können beispielsweise die Anschubfinanzierung des Projekts oder regelmäßige Zahlungen des Kunden sein. Alle Kosten und Einnahmen kann der Spieler über den Budgetdialog einsehen und mitverfolgen (siehe Abbildung 6).

Kontostand 112.813,26 €	
Einnahmen	50.000,00 € 22.10.2014 Monatliche Tranche 50.000,00 € 01.11.2014 Monatliche Tranche 50.000,00 € 01.12.2014 Monatliche Tranche
Summe	150.000,00 €
Laufende Kosten	-3.968,83 € Okt 2014 -20.239,04 € Nov 2014 -3.078,87 € Dez 2014
Summe	-27.286,74 €
Einmalige Kosten	-9.900,00 € 05.11.2014 Abfindung für Henning Hümmel
Summe	-9.900,00 €
Aktuelle laufende Kosten pro Monat	Mitarbeiter -5.950,00 € Krista Lang -6.120,00 € Martin Schwacke Material- und Lizenzkosten -300,00 € Krista Lang -300,00 € Martin Schwacke Fixkosten -10.000,00 €
Summe	-36.360,00 €
Aktuelle Monatsbilanz	Einnahmen 50.000,00 € Regelmäßige Ausgaben -36.207,12 € Einmalige Ausgaben 0,00 €
Summe	13.792,88 €

Abbildung 6: Budgetdialog

Ein negatives Budget kostet zusätzliches Geld für die dadurch fälligen Zinsen. Der Spieler sollte deshalb möglichst gut mit den ihm zur Verfügung stehenden Mitteln haushalten. Finanzielle Gewinne und Verluste haben keine direkte Auswirkung auf das Projekt an sich. Sie gehen aber in die abschließende Bewertung des Projekts ein.

Wer gewinnt? – Score und Projektbericht

Der Spieler entscheidet selbst, wann das Projekt endet. Typischerweise wurde entweder das Produkt vollständig und in einer annehmbaren Qualität entwickelt, oder das im Szenario festgelegte Enddatum für das Projekt wurde überschritten. Das Spiel selbst kann trotz Projektende beliebig fortgesetzt werden. So kann die Projektlaufzeit bei Bedarf auch überschritten werden.

Wichtig für den Lernerfolg ist ein konstantes Feedback an den Spieler. Dafür verwenden wir die Projektbewertung. Sie besteht aus einem Score und einem zusätzlichen kurzen Projektbericht, der die einzelnen Teilaspekte zusammenfasst und bewertet. Dieser, in Abbildung 7 dargestellte Bericht, kann vom Spieler jederzeit eingesehen werden. So bekommt der Spieler auch schon während des Spiels eine Rückmeldung, wo sein Projekt steht.

Die Bewertung enthält die wesentlichen Projektbewertungskategorien: Vollständigkeit, Qualität, Kosten und Dauer. Vollständigkeit und Qualität können zu einer Produktbewertung zusammengefasst werden. Um einen, auch über mehrere Projekte hinweg vergleichbaren Wert für die einzelnen Aspekte zu erhalten,

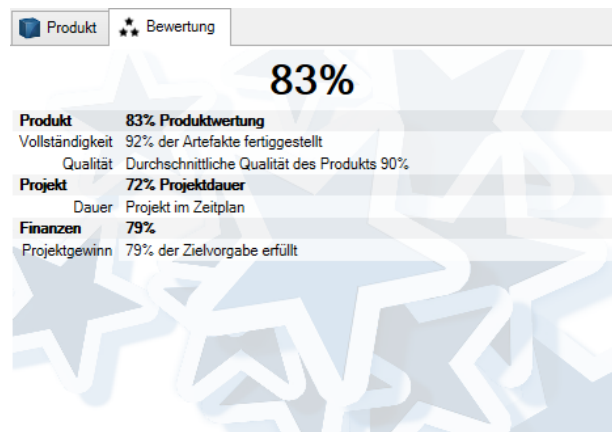


Abbildung 7: Score und Projektbericht

werden die Bewertungen in einem Prozentsatz ausgedrückt. Dabei steht 100% für ein optimales Ergebnis. In wenigen Fällen, wenn beispielsweise beim Projektgewinn ein Optimum aufgrund der nach oben offenen Skala nicht definiert werden kann, sind auch Werte größer 100% möglich.

Die Vollständigkeit c wird, wie in Formel (1) gezeigt, anhand des bearbeiteten Umfangs der einzelnen Artefakte und des Gesamtumfangs des Produkts berechnet.

Die Qualität des Produkts wird als die mit dem Umfang der Artefakte gewichtete Summe der einzelnen Artefakte berechnet (2). Jedes Artefakt besitzt dabei ein Qualitätslevel a_q zwischen 0 und 1. Dabei steht 1 für absolute Fehlerfreiheit bzw. ein optimales Ergebnis, 0 für ein vollständig falsches, bzw. aufgrund der hohen Fehlerzahl unbrauchbares Artefakt. Ein Qualitätswert von 1 bzw. 100% entspricht damit einem Produkt mit maximaler Qualität.

Die Produktwertung ist die Kombination aus Qualität und Vollständigkeit der Artefakte (3). Da die Vollständigkeit absolut in Arbeitsstunden angegeben ist, gehen die Artefaktattribute entsprechend ihres Umfangs gewichtet in die Produktwertung ein. Kleine Artefakte haben daher eine geringere Auswirkung auf die Produktwertung als Artefakte mit größerem Umfang.

Durch das Szenario wird für jedes Projekt eine Projektlaufzeit mittels Starttermin und Endtermin vorgegeben. Eine Überschreitung des Termins ist möglich, wirkt sich aber negativ auf die Wertung aus. Die Projektdauer berechnet sich wie in (4) gezeigt, aus der vorgegebenen und der tatsächlichen Projektlaufzeit.

Die Kostenwertung wird, wie in (5) gezeigt, anhand des aktuellen Kontostands berechnet. Dabei wird zusätzlich der bereits umgesetzte Produktumfang berücksichtigt. Das verhindert eine hohe Wertung, die zu Beginn des Projekts oder durch das Ansammeln von Finanzmitteln, ohne die Investition in das Produkt entstehen könnte. Entsprechend der Annahme, dass ein Produkt mit einem höheren Umfang auch einen

$$c = \frac{\sum_{a \in A} a_c}{\sum_{a \in A} a_s} \quad (1)$$

$$\text{Qualität} = \frac{\sum_{a \in A} a_q \cdot a_s}{\sum_{a \in A} a_s} \quad (2)$$

$$\text{Produkt} = \frac{\sum_{a \in A} a_c \cdot a_q}{\sum_{a \in A} a_s} \quad (3)$$

$$\text{Dauer} = \frac{t_{\text{Heute}} - t_{\text{Start}}}{t_{\text{Ende}} - t_{\text{Start}}} \quad (4)$$

$$\text{Kosten} = \max \left(0, 0.5 + \frac{k \cdot b}{\sum_{a \in A} a_s} \cdot \begin{cases} c & b > 0 \\ 1 & \text{sonst} \end{cases} \right) \quad (5)$$

$$\text{Score} = \frac{\text{Produkt} \cdot 0.5}{\max(1, \text{Dauer})} + \text{Kosten} \cdot 0.5 \quad (6)$$

A : Menge aller Produktartefakte

a_c : Fortschritt von Artefakt a in Arbeitsstunden

a_q : Qualität des Artefakts $\in [0,1]$

a_s : Umfang von Artefakt a in Arbeitsstunden

t_{Heute} : Aktuelles Datum der Simulation

t_{Start} : Datum des Projektbeginns

t_{Ende} : Datum des vorgegebenen Projektendes

b : aktueller Kontostand in €

k : Faktor entsprechend der Gewinnerwartung

entsprechend höheren Umsatz und Gewinn bedeutet, wird die Kostenwertung über den Produktumfang normiert.

Um einen relativen Wert für die Wertung zu bekommen, muss die Gewinnerwartung k berücksichtigt werden. Diese ist im Szenario definiert und hängt maßgeblich von den Entwicklungskosten und den Zahlungen des Kunden ab. Ein kostenneutrales Projekt wird mit 50% bewertet. Ein Projekt, das die Gewinnerwartung erfüllt mit 100%. Projekte, die Verluste verursachen, erhalten eine Wertung unter 50%.

Die Gesamtwertung (6) setzt sich aus den oben beschriebenen Teilwertungen zusammen. Dabei enthält die Produktwertung die Vollständigkeit und Qualität, die Kostenwertung die Vollständigkeit und den Projektgewinn bzw. -verlust. Die Kostenwertung enthält indirekt die Projektdauer, da eine längere Projektlaufzeit aufgrund der laufenden Kosten den Gewinn reduziert. Produkt- und Kostenwertung gehen zu gleichen Teilen in die Gesamtwertung ein. Da in der Produktwertung die Projektlaufzeit nicht enthalten ist, muss diese hier besonders berücksichtigt werden. Dazu wird die Produktwertung durch die zur veranschlagten Projektlaufzeit relativen tatsächlichen Laufzeit geteilt.

Um zu verhindern, dass Anomalien in der Bewertung entstehen, weil ein Projekt sehr früh beendet wird, wird eine Projektlaufzeit unter der im Szenario

veranschlagten Zeit nicht zusätzlich belohnt. Da durch ein frühes Beenden des Projekts die Kostenwertung verbessert werden kann, geht dieser Aspekt trotzdem indirekt in die Gesamtwertung ein.

Es besteht die Möglichkeit, Szenarien ohne Budgetaspekte zu definieren, wenn diese für das Lernziel nicht relevant sind. In diesem Fall entfällt die Kostenwertung und die Gesamtwertung entspricht nur dem Quotienten aus Produktwertung und Projektdauer.

Beispielszenario: Softwaregrundprojekt

Ein wesentlicher Meilenstein der Informatikausbildung an der Universität Ulm ist das Softwaregrundprojekt. Hier kommen die Studierenden zum ersten Mal mit der Softwaretechnik in Berührung. Während sie zu Beginn des Studiums lediglich kleinere Übungsaufgaben implementiert haben, müssen sie im Softwaregrundprojekt zum ersten Mal ein vollständiges System nach ausgewählten Prinzipien der Softwaretechnik entwickeln. Dieses Praktikum geht über zwei Semester und wird von einer Vorlesung zur Softwaretechnik begleitet.

Wir verwenden im Softwaregrundprojekt eine angepasste Form der Fusion-Methode nach Coleman et al. (Coleman u. a., 1994) mit einer UML-Erweiterung angelehnt an die Ideen in (Bittner, 2006). Da die Studierenden noch keinerlei Erfahrung mit Fusion oder anderen Methoden der Softwaretechnik gemacht haben, fällt es ihnen anfänglich oft schwer, sich in der Vielzahl der unterschiedlichen Aktivitäten und zu erstellenden Artefakte zurechtzufinden.

Um den Studierenden möglichst früh einen Überblick über die wichtigen Elemente des Projekts und deren Zusammenhänge zu ermöglichen, haben wir das Softwaregrundprojekt als Szenario für unser Spiel modelliert. Während im Praktikum selbst ein konkretes System entwickelt wird, ist das Produkt im Spiel sehr abstrakt gehalten und von der konkreten Aufgabenstellung des Praktikums unabhängig.

Wenn im Folgenden vom *Softwaregrundprojekt* gesprochen wird, ist immer die tatsächliche Lehrveranstaltung gemeint. Dagegen beschreiben wir mit *Spiel* oder *Szenario* das simulierte Projekt in der Spielumgebung. Sowohl die Aktivitäten, als auch die Artefakte des Szenarios leiten sich direkt aus der Aufgabenstellung und den zu erstellenden Teilergebnissen im Softwaregrundprojekt ab.

Vorgehensmodell

Das Vorgehensmodell definiert die Rollen, Aktivitäten und Artefakttypen im Projekt. Da das Softwaregrundprojekt so ausgelegt ist, dass jeder der Studierenden jede der unterschiedlichen Aktivitäten selbst mindestens einmal durchgeführt hat, benötigen wir kein besonderes Rollenkonzept, welches die einzelnen Mitarbeiter auf ihre Aufgaben einschränkt.

Die Studierenden müssen im Laufe des Projekts sieben unterschiedliche Aktivitäten ausführen:

- **Kontextanalyse:** Zu Beginn des Projekts muss anhand von Produktskizzen und Kundengesprächen herausgefunden werden, was der Kunde für ein System wünscht. Diese Erkenntnisse müssen geeignet dokumentiert bzw. modelliert werden. Die Kontextanalyse bearbeitet Artefakte des Typs *Anforderungsdokument* und *UI Spezifikation*.
- **Entwurf:** Mittels der Entwurfsaktivitäten wird auf Basis der Dokumente, die während der Kontextanalyse erstellt wurden, das Architekturdesign des Systems entwickelt. Beim Entwurf werden Artefakte des Typs *Entwurfsdokument* bearbeitet.
- **Implementierung:** Die Implementierungsaktivität beschreibt sowohl das Erstellen des Quellcodes, als auch dessen Dokumentation. Dabei werden Artefakte des Typs *Implementierung* bearbeitet.
- **Dokumentation:** Neben der inhärenten Dokumentation der einzelnen Artefakte, ist für viele Bereiche eine zusätzliche Dokumentation, wie beispielsweise ein Benutzerhandbuch, notwendig. Dabei werden Artefakte des Typs *Dokumentation* erstellt.
- **Projektplanung:** Diese Aktivität befasst sich mit allem, was zur Planung und Steuerung des Projekts gehört. Der dazugehörige Artefakttyp ist *Controlling*.
- **Qualitätssicherung:** Vorbereitende und begleitende Maßnahmen zur Qualitätssicherung. Erstellt werden Artefakte des Typs *Qualitätssicherung*.
- **Testen:** Dient zur Verbesserung der Qualität von Artefakten des Typs *Implementierung*.

Alle genannten Aktivitäten, mit Ausnahme von *Testen*, sind Entwicklungsaktivitäten, die dazu dienen, verschiedene Artefakte zu erstellen. Testen ist eine Qualitätssicherungsaktivität, die dazu dient, die Qualität von bestehenden Artefakten zu verbessern, indem die darin enthaltenen Fehler aufgedeckt und behoben werden.

Produkt

Das zu entwickelnde Produkt besteht aus ca. 60 Artefakten. Jedes dieser Artefakte ist einem der Artefakttypen zugeordnet. Die einzelnen Artefakte haben einen individuellen Schwierigkeitsgrad und stellen unterschiedliche Anforderungen an die fachlichen Fähigkeiten der Mitarbeiter. Der Umfang der einzelnen Artefakte ist ebenfalls unterschiedlich.

Den Artefakten sind teilweise ein oder mehrere der vier Fachbereiche *UML*, *Gestaltung*, *Relationale Datenbanken* und *Objektorientierte Programmierung* als Schwerpunkt zugeordnet. Je besser sich diese Fachbereiche mit den entsprechenden Fähigkeiten der zuständigen Mitarbeiter decken, umso besser sind die Ergebnisse der Aktivitäten.

Alle im Szenario definierten Artefakte müssen von den Studierenden später auch im Softwaregrundpro-

jekt erstellt werden. Sie bauen zum Teil aufeinander auf bzw. beeinflussen sich gegenseitig. Diese Abhängigkeiten sind auch im Szenario als Voraussetzungen und Einflüsse modelliert worden. So beeinflussen beispielsweise die Programmierkonventionen die Qualität der Implementierung und Tests können nicht ohne das vorherige Erstellen der entsprechenden Testpläne durchgeführt werden.

Das Produkt besteht aus folgenden Artefakten:

- **Anforderungsdokumente:** Überblick, Fachwissen, Anwendungsfälle, Szenarien, Systemaufgaben, Nichtfunktionale Anforderungen, Domänenmodell, Schnittstellenmodell und Pflichtenheft
- **UI Spezifikationen:** Dialogstruktur, Dialoggestaltung und Nutzungskonzept
- **Entwurfsdokumente:** Datenbanklayout, Kommunikationsmodell, Klassenmodell und Methodenbeschreibung
- **Implementierung:** Systemkern, Tools, Datenbankbindung, Übergreifende UI Konzepte und fünf Komponenten mit jeweils einem Artefakt für Benutzeroberfläche und Logik.
- **Controlling:** Projektplan und Rollenverteilung
- **Qualitätssicherung:** Allgemeine Testrichtlinien und jeweils ein Testplan zu jedem Implementierungsartefakt.
- **Dokumentation:** Programmierkonventionen, Architekturbeschreibung, Komponentendiagramm, Beschreibung des Gesamtkonzepts, Produkt- und Lizenzlisten, Datenbankbeschreibung, Installationsanleitung, Benutzerdokumentation und Projekttagebuch.

Mitarbeiter

Das Softwaregrundprojekt wird in Teams zu jeweils sechs Studenten durchgeführt. Zu Beginn arbeiten dabei alle gemeinsam an der Kontextanalyse. Nachdem diese Phase mit der Erstellung des Pflichtenhefts abgeschlossen wurde, werden die Verantwortlichkeiten für die folgenden Aufgaben unter den Teammitgliedern aufgeteilt. Es gilt aber weiterhin, dass sich jedes Teammitglied an allen Aufgaben beteiligen soll, um in jeden Bereich Einblick zu erhalten. Lediglich die Schwerpunkte der einzelnen Teammitglieder können sich unterscheiden.

Die Verantwortlichkeitsbereiche sind *Systemarchitektur*, *Qualitätssicherung*, *Verifikation und Validierung*, *Dokumentation*, *Infrastruktur und Projektverwaltung* und *Projektmanagement*. Die Implementierung ist kein eigener Bereich, da sich hier alle Studenten in gleichem Umfang beteiligen sollen.

Im Szenario ist jeweils ein Mitarbeiter definiert, der von seinem Fähigkeitsprofil gut zu einem der Bereiche passt. Der Bereich Projektmanagement wird vom Spieler selbst übernommen und bedarf deshalb keinem simulierten Mitarbeiter. Die Projektverwaltung lässt sich im Szenario nicht sinnvoll umsetzen, da es sich hierbei um eine übergeordnete Aufgabe handelt. Daher wurde Projektverwalter und Projektleiter jeweils

ein Fähigkeitsprofil zugewiesen, das die der anderen Mitarbeiter möglichst gut ergänzt. Dadurch steht dem Spieler ein ausgewogenes Team zur Verfügung.

Da das Softwaregrundprojekt mit einem festen Projektteam und ohne Finanzplanung durchgeführt wird, wurden die Budgetplanung sowie der Arbeitsmarkt deaktiviert.

Dieses Szenario stellt die Grundlage für unsere Evaluation dar.

Evaluation

Um den Lerneffekt, die Akzeptanz und die Bedienbarkeit des Spiels zu testen, haben wir eine Evaluation in Form einer Onlinebefragung unter den Spielteilnehmern durchgeführt.

Da wir das Spiel bisher nur einer kleinen Gruppe von 14 Studierenden zugänglich gemacht haben, sind die quantitativen Ergebnisse dieser Befragung nicht ausreichend belastbar und wurden deshalb durch eine qualitative Evaluation in Form von detailliertem persönlichem Feedback ergänzt.

Da die quantitative Auswertung der Ergebnisse der Onlinebefragung sowohl mit unseren Erwartungen, als auch mit den Ergebnissen des persönlichen Feedbacks nahezu übereinstimmt, sind wir momentan dabei, die Evaluation auf eine weitere Gruppe mit ca. 150 Studierenden auszuweiten. Ziel dabei ist es, die vorliegenden Ergebnisse auch quantitativ bestätigen zu können.

Der verwendete Fragebogen enthält 30 Fragen, die in sieben Blöcke aufgeteilt sind:

1. Fragen zur Erfassung der Rahmensituation. Dazu wurden Fragen zum allgemeinen Spielverhalten und zum Umgang mit komplexen Problemen gestellt.
2. Fragen zum Spielerlebnis. Dazu gehören beispielsweise Spielzeit, Spielspaß und Wiederspielwert des Spiels.
3. Fragen zur Erfassung des Lernfortschritts in Bezug auf das Werkzeug MS Project. Dazu wurde unter anderem der diesbezügliche Kenntnisstand vor und nach dem Spiel erfasst.
4. Fragen zur Erfassung des Lernfortschritts in Bezug auf das Vorgehensmodell. Dazu wurde wie bei 3. der entsprechende Kenntnisstand vor und nach dem Spiel erfasst. Anschließend wurden Fragen gestellt, bei denen die Spieler ihren Kenntnisstand in Bezug auf das im Szenario verwendete Vorgehensmodell einschätzen sollten.
5. Analog zu 4. wurde der Lernfortschritt in Bezug auf das Produkt, d.h. die Dokumente und Modelle, die während des Softwaregrundprojekts erstellt werden müssen, evaluiert.
6. Fragen zur Erfassung allgemeiner Lernerfolge im Projektmanagementbereich.

7. Abschließende Fragen zur Darstellung der Informationen im Spiel und der Bedienbarkeit von Spiel und MS Project.

Im Folgenden liefern wir eine kurze Zusammenfassung der Ergebnisse der Befragung.

Der allgemeine Spielspaß wurde von den Teilnehmern als eher gering eingestuft. Dieses Ergebnis ist nicht weiter überraschend, da sich das Spiel auf die reinen Lernaspekte beschränkt und keine zusätzlichen Elemente eingebaut wurden, die dem Spielspaß dienen. Der Aspekt des Lernens steht bei diesem Spiel eindeutig im Vordergrund und das wird von den Studierenden auch so erkannt. Trotzdem würden die meisten Teilnehmer das Spiel mit einem anderen Szenario erneut spielen und es auch ihren Kommilitonen weiterempfehlen. Das spricht in unseren Augen für die Akzeptanz des Spiels als Lernmittel und lässt einen gewissen Wiederspielwert erkennen.

Anhand der Fragen zu MS Project konnten wir feststellen, dass alle Spieler ihre Fähigkeiten in Bezug auf das Werkzeug nach dem Spiel höher einschätzen als vor dem Spiel. Daraus schließen wir, dass die Benutzung von MS Project im Rahmen des Spiels auch einen Trainingseffekt auf den Umgang mit diesem und ähnlichen Werkzeugen hat. Das wird auch von den Spielern selbst so gesehen. Alle haben angegeben, durch das Spiel im Umgang mit MS Project etwas gelernt zu haben, wenn auch unterschiedlich viel. Alle Teilnehmer trauen sich jetzt zu, ein einfaches Projekt mit MS Project zu managen. Auch diejenigen, die angegeben haben, vor dem Spiel mit MS Project nicht zurechtgekommen zu sein.

Ein ähnliches Ergebnis konnten wir bei den allgemeinen Fähigkeiten ein Projekt zu planen beobachten. Auch hier trauen sich jetzt alle Spieler zu, eigenständig ein Projekt zu planen. Allerdings haben die meisten angegeben, diese Fähigkeit auch schon vor dem Spiel besessen zu haben. Trotzdem haben alle Spieler angegeben, durch das Spiel ihre Projektmanagementfähigkeiten verbessert zu haben.

Im Bezug auf die Aktivitäten und Dokumente, die im Softwaregrundprojekt benötigt werden, haben nahezu alle Spieler nach Abschluss des Spiels angegeben, die Aktivitäten und Artefakte benennen, erklären und in ihren Kontext einordnen zu können.

Mit der Bedienung des Spiels sind die Teilnehmer gut zurechtgekommen. Für die Darstellung des Szenarios gab es einige Anregungen zur Verbesserung der Übersichtlichkeit der Zusammenhänge von Aktivitäten und Artefakten. Die Meinungen zur Bedienbarkeit von MS Project, unabhängig des Spiels, waren sehr unterschiedlich und gingen von *problemlos* bis hin zu *nur schlecht bedienbar*.

Die Ergebnisse zeigen uns, dass die grundsätzliche Konzeption des Spiels erfolversprechend ist. Es ist notwendig, einige kleinere Anpassungen vorzunehmen, um das Spiel und dessen Bedienbarkeit weiter

zu optimieren. Sollte eine umfangreichere Evaluation diese Ergebnisse bestätigen, ist geplant, das Spiel zusätzlich um neue Konzepte zu erweitern. Eines der Hauptziele wird es dann sein, den Spielspaß zu erhöhen.

Fazit und Ausblick

Die Ausbildung von Studierenden im Bereich des Projektmanagements stellt für die Lehrenden eine große Herausforderung dar, da die notwendige praktische Erfahrung nur durch geeignete eigenständige Projektarbeit gewonnen werden kann.

Mit unserem Planspiel haben wir eine Möglichkeit gefunden, praktische Erfahrung spielerisch zu gewinnen. Ziel dabei war es, eine möglichst natürliche Spielumgebung zu schaffen, die es dem Spieler ermöglicht, neben den Erfahrungen im Bereich des Projektmanagements gleichzeitig den Umgang mit den dazu benötigten Werkzeugen zu üben. Dazu haben wir ein Simulationsframework als AddIn in MS Project integriert, so dass der Spieler die mit MS Project geplanten Projektvorgänge direkt simulieren und dadurch die Ergebnisse seiner Planung sofort sehen kann.

Die von uns durchgeführte Evaluation hat gezeigt, dass wir mit dieser Art von Lernspiel einen erfolgversprechenden Weg gewählt haben. Um das volle Potential des Ansatzes nutzen zu können, muss diese Idee noch weiter verfeinert und ausgebaut werden.

Als erster Schritt soll das Feedback, das der Spieler durch das Spiel erhält, erweitert werden. Um seine Mitarbeiter besser beobachten, einschätzen und steuern zu können, ist es sinnvoll, dem Spieler weitere Informationen über den Zustand der Mitarbeiter und dem Team im Gesamten zu geben. Dazu gehören zum Beispiel die Motivation der Mitarbeiter, das soziale Gefüge im Team, Vorlieben und Abneigungen der einzelnen Mitarbeiter für bestimmte Aufgaben, sowie diverse weiche Eigenschaften wie Lerngeschwindigkeit und Kommunikationsfähigkeit. All diese Parameter werden momentan zwar von der Simulation berücksichtigt, dem Spieler aber nicht mitgeteilt. Durch eine geeignete Aufbereitung dieser Informationen und einer Integration in die Spielumgebung wird das Spiel für den Spieler realistischer und besser beherrschbar.

Ein ähnlicher Ansatz gilt für die Darstellung des Produkts, den Anforderungen für die Erstellung der einzelnen Artefakte, sowie deren Abhängigkeiten und Einflüsse untereinander. Hier kann vor allem eine geeignete Visualisierung helfen, das zu entwickelnde Produkt besser zu verstehen und damit auch das Projekt besser steuern zu können.

Die Berechnung der Gesamtwertung des Projekts berücksichtigt momentan lediglich den Umfang der einzelnen Produktartefakte. In der Realität sind nicht alle Artefakte gleich wichtig für den Erfolg

eines Projekts. Die Wichtigkeit eines Artefakts ist nicht ausschließlich an dessen Umfang zu erkennen, sondern setzt zusätzliche Informationen über die Bedeutung bzw. Aufgabe des Artefakts im Produkt voraus. Ein möglicher Ansatz für eine Verbesserung der Berechnung der Score ist es, einen entsprechenden zusätzlichen Gewichtungsfaktor für die einzelnen Artefakte einzuführen. Eine mögliche weitere Verfeinerung dieses Ansatzes ist es, die Qualität und den Umfang getrennt zu gewichten. So kann bei für das Produkt kritischen Artefakten ein höheres Augenmerk auf die Qualität gefordert werden.

Am Ende des Spiels soll für den Spieler, neben dem kurzen Übersichtsbericht, ein zusätzlicher Projektbericht über den Verlauf seines Projekts erstellt werden. Je nach Szenario kann dieser dem Spieler auch schon während des Spiels zugänglich gemacht werden. Ein solcher Bericht kann den zeitlichen Fortschritts- und Qualitätsverlauf der einzelnen Artefakte enthalten. Ebenso hilfreich ist eine Verfolgung der Eigenschaften der einzelnen Mitarbeiter. So kann schnell erkannt werden, wie sich Projektentscheidungen auf beispielsweise die Motivation, die Fitness oder die Fähigkeiten der Mitarbeiter ausgewirkt haben. All diese Informationen müssen geeignet aufbereitet und dargestellt werden.

Bislang unterstützt das Spiel ausschließlich Szenarien, die auf linearen Prozessmodellen basieren. Grund dafür ist die, über die Szenarien definierte, statische Konfiguration des Produkts, bestehend aus seinen einzelnen Artefakten. Bei iterativen und agilen Prozessmodellen ist diese Aufteilung zu Beginn des Projekts noch nicht vollständig bekannt. Um nichtlineare Prozessmodelle zu unterstützen, muss der Spieler während des Projekts entscheiden können, welche neuen Artefakte zu erstellen sind oder welche bestehenden Artefakte verworfen werden sollen. Damit gibt er vor, wie er die Entwicklung des Produkts strukturieren möchte. Das kann beispielsweise durch das Erstellen einer neuen Iteration bei iterativen Prozessmodellen, oder innerhalb der Planung eines neuen Sprints bei agilen Prozessmodellen wie SCRUM geschehen.

In wie weit sich eine solche zusätzliche Funktionalität in das Spiel integrieren lässt, haben wir bislang noch nicht getestet. Da das verwendete Simulationsframework die Möglichkeit bietet, die Struktur des Produkts während der Simulation zu verändern, sind zumindest die technischen Grundlagen dafür vorhanden.

Die Evaluation hat uns gezeigt, dass noch Potential zur Steigerung des Spielspaßes vorhanden ist. Beim Design unseres Testszenarios haben wir darauf bewusst keinen Wert gelegt. Langfristig macht es aber Sinn ein Spiel zu entwickeln, das dem Spieler auch

Spaß macht. Ein möglicher Ansatzpunkt dafür ist eine detailliertere Gestaltung der Szenarien. Diese können mit zusätzlichen Spielelementen, wie beispielsweise zufälligen Spielereignissen erweitert werden. Eine weitere Möglichkeit das Spiel abwechslungsreicher zu gestalten ist es, das Szenario um eine Storyline zu ergänzen und diese medial mit Texten, Grafiken und Videos ansprechend zu gestalten. Diese Spielelemente zielen primär darauf ab, das Spiel abwechslungsreicher und spannender zu machen.

Die Erfahrungen, die wir mit diesem Projekt gemacht haben, zeigen uns, dass es sinnvoll ist, bei der Entwicklung von Lernspielen eine möglichst realitätsnahe Spielumgebung zu schaffen. Es hat sich herausgestellt, dass sich dies durch das Einbeziehen der Werkzeuge, die in der realen Entwicklung verwendet werden, umsetzen lässt. Da diese Werkzeuge häufig eine gute Erweiterbarkeit mit entsprechenden Schnittstellen aufweisen, lässt sich eine Integration häufig ohne großen Aufwand realisieren.

In unserem Spiel haben wir uns auf den Aspekt des Projektmanagements beschränkt. Ein interessanter Ansatz für weitere Arbeiten ist es, andere Aspekte des Software Engineering in ähnlicher Weise in die passenden Werkzeuge zu integrieren, und diese auf einer höheren Ebene zu einem gemeinsamen Softwareprojekt zusammenzufügen. So können die Studierenden alle relevanten Aspekte im Bereich des Software Engineering an einem einzigen durchgehenden Projekt praktisch üben.

Literatur

- [Bittner 2006] BITTNER, M.: *Enhancing the Fusion Method to Fusion B: Requirements Engineering and Formal Specification*, Technischen Universität Berlin, Diss., 2006
- [Coleman u. a. 1994] COLEMAN, Derek ; ARNOLD, Patrick ; BODOFF, Stephanie ; DOLLIN, Chris ; GILCHRIST, Helena ; HAYES, Fiona ; JEREMAES, Paul: *Object-oriented Development: The Fusion Method*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 1994
- [Jain u. Boehm 2006] JAIN, Apurva ; BOEHM, Barry W.: *SimVBSE: Developing a Game for Value-Based Software Engineering*. In: *CSEE&T*, IEEE Computer Society, 2006, S. 103–114
- [Ludewig u. a. 1992] LUDEWIG, J. ; BASSLER, T. ; DEININGER, M. ; SCHNEIDER, K. ; SCHWILLE, J.: *SESAM-simulating software projects*. In: *Software Engineering and Knowledge Engineering, 1992. Proceedings., Fourth International Conference on*, 1992, S. 608–615
- [Microsoft Corporation 2013] MICROSOFT CORPORATION: *Microsoft Project*. Version 2013. 2013
- [Nassal 2014] NASSAL, Alexander: *A General Framework for Software Project Management Simulation Games*. In: *Information Systems and Technologies (CISTI) 2014, 9th Iberian Conference on Information Systems and Technologies*, 2014, S. 321–325
- [Navarro 2006] NAVARRO, Emily: *SimSE: A Software Engineering Simulation Environment for Software Process Education*, University of California, Irvine, Diss., 2006
- [de Oliveira Barros u. a. 2006] OLIVEIRA BARROS, Márcio de ; DANTAS, Alexandre R. ; VERONESE, Gustavo O. ; WERNER, Cláudia Maria L.: *Model-driven Game Development: Experience and Model Enhancements in Software Project Management Education*. In: *Software Process: Improvement and Practice* 11 (2006), Nr. 4, S. 411–421
- [Pieper 2013] PIEPER, Jöran: *Alles nur Spielerei? Neue Ansätze für digitales spielbasiertes Lernen von Softwareprozessen*. In: SPILLNER, Andreas (Hrsg.) ; LICHTER, Horst (Hrsg.): *Tagungsband des 13. Workshops "Software Engineering im Unterricht der Hochschulen" (SEUH) 2013, Aachen*, CEUR Workshop Proceedings Vol. 956, 2013, 131–139
- [Shaw u. Dermoudy 2005] SHAW, Katherine ; DERMOUDY, Julian R.: *Engendering an Empathy for Software Engineering*. In: YOUNG, Alison (Hrsg.) ; TOLHURST, Denise (Hrsg.): *ACE Bd. 42*, Australian Computer Society, 2005 (CRPIT), S. 135–144
- [Ye u. a. 2007] YE, En ; LIU, Chang ; POLACK-WAHL, J.A.: *Enhancing Software Engineering Education Using Teaching Aids in 3-D Online Virtual Worlds*. In: *Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, 2007