

Agile Softwareentwicklung im Informatikunterricht – Ein Best-Practice-Beispiel am Spiel „Pengu“

Petra Kastl, Silva März und Ralf Romeike¹

Abstract: Im Beitrag wird der Einsatz agiler Methoden im Informatikunterricht an einem konkreten Beispiel illustriert. Am Spiel „Pengu“, welches mit der Programmierumgebung Greenfoot implementiert werden soll, werden agile Praktiken wie User Stories, Tasks, Zeit- und Prioritätenabschätzung, Iterationen und ein mögliches Project Board exemplarisch dargestellt. Das Beispiel kann als Orientierung genutzt werden, um agile Methoden zu verstehen und in eigenen Projekten umzusetzen.

Keywords: Projektunterricht, Agile Methoden der Softwareentwicklung

1 Einleitung

Eine wesentliche Herausforderung bei der Implementierung innovativer Unterrichtsmethoden stellt das Finden geeigneter Beispiele dar, an denen sowohl die Lehrkraft, als auch die Schülerinnen und Schüler Kernideen und Umsetzungsmöglichkeiten konkret nachvollziehen und erfassen können. Eine solche Innovation stellen im Moment agile Methoden der Softwareentwicklung dar, die Potential besitzen, Probleme zu vermeiden, die Projektunterricht, der herkömmlichen, linearen Prozessmodellen folgt, regelmäßig offenbart. Unter agilen Methoden ist eine Sammlung von Artefakten und Praktiken zu verstehen, die den Projektlauf strukturieren und dabei den Beteiligten konkrete Handlungsoptionen und Kommunikationsanlässe zur Hand geben. Die Praktiken müssen nicht starr verwendet werden, sondern können passend zu dem jeweiligen Projekt neu zusammengestellt werden. Etablierte Frameworks agiler Methoden sind zum Beispiel Scrum, eXtreme Programming oder auch Kanban.

Im Folgenden werden für Unterrichtsprojekte sinnvolle agile Praktiken kurz charakterisiert und am Beispiel des Spiels „Pengu“ exemplarisch umgesetzt. Der Ablauf orientiert sich dabei am agilen Projektmodell für den Informatikunterricht AMoPCE ([RG12], vgl. Abb. 1). Hierbei werden zu Beginn alle bisher bekannten und gewünschten Anforderungen an das System bzw. die Software definiert und in diesem Projekt als sogenannte User Stories festgehalten. Diese als „Geschichten“ beschriebenen Funktionen werden noch weiter hinsichtlich ihrer Umsetzung spezifiziert, Prioritäten und Aufwand abgeschätzt und auf einem Project Board festgehalten. Der weitere Prozess gliedert sich in mehrere Iterationen, wobei jede Iteration ein „Mini-Projekt“ darstellt, in welchem die

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen
petra.kastl@fau.de, silva.maerz@fau.de, ralf.romeike@fau.de

Phasen des Planens, Entwerfens, Implementierens und Testens jeweils abschließend durchlaufen werden. Innerhalb der Iterationen wird zu Beginn jeder Unterrichtseinheit ein Standup-Meeting durchgeführt, die Projektorganisation wird unterstützt durch ein Project-Board.

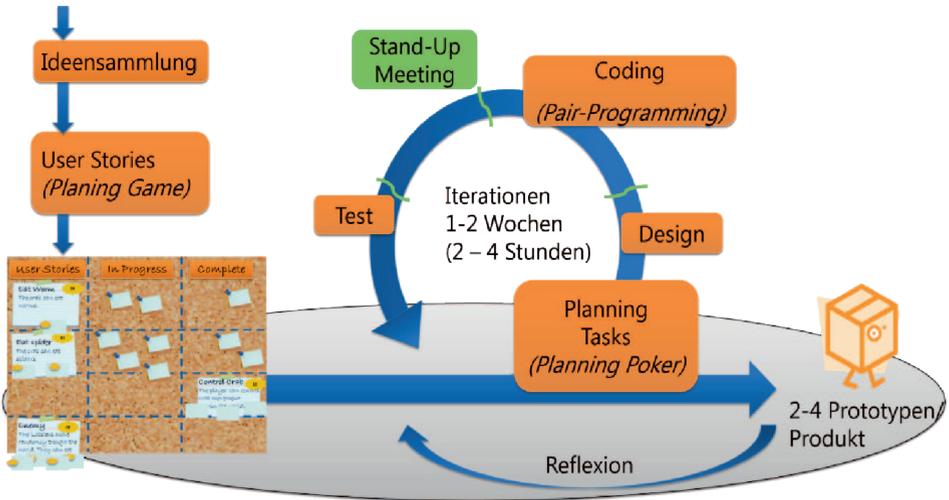


Abb. 1: Agiles Projektmodell für den Informatikunterricht AMoPCE

2 Agile Methoden am Beispiel von „Pengu“

Das Spiel „Pengu“ ist ein Jump 'n' Run-Spiel, bei welchem es die Aufgabe des Spielers ist, eine Spielfigur auf einer bewegten Wolke über einen Abgrund zu steuern (siehe Abb. 2). Umgesetzt wird dieses Szenario in diesem Beispiel mit Greenfoot. Greenfoot stellt auf der Programmiersprache Java basierende Miniwelten zur Verfügung, welche insbesondere für zweidimensionale Spiele und Simulationen geeignet sind. Greenfoot ermöglicht Programmieranfängern, die objektorientierte Programmierung auf interaktive Weise kennenlernen.

Im Folgenden soll nun die Spielidee konkret mit Hilfe agiler Methoden umgesetzt werden. Hierzu wird das Szenario zuerst in sogenannte User Storys und Tasks aufgeteilt. Anschließend wird das weitere Projektvorgehen beschrieben.



Abb. 2: Spiel „Pengu“ in Greenfoot

2.1 User Stories und Tasks

User Stories beschreiben in kurzer Form Funktionalitäten und Fähigkeiten der zu entwickelnden Software, die dem Nutzer zur Verfügung stehen sollen. Jede User Story beschränkt sich dazu auf eine konkrete Aktivität und wird aus Sicht des Kunden beschrieben. Entsprechend sollten sie so kurz sein, dass sie auf eine Karteikarte passen und auch ohne Programmierkenntnisse verständlich sind. Auf den Karteikarten kann später auch der prognostizierte Aufwand sowie die Priorität vermerkt werden.

Die Erstellung der User Stories erfolgt unter Einbeziehung des Auftraggebers und erfordert domänenspezifisches Wissen aus dem Kontext der zu erstellenden Software. So ist es in unserem Beispiel erforderlich, dass das Team eine Vorstellung vom Spielfluss und Möglichkeiten solcher Jump-and-Run-Spiele besitzt, um die Anforderungen klar aufzustellen und auch eigene Ideen mit einzubringen.

Eine User Story besteht aus einem Titel, einer Beschreibung, einer zeitlichen Abschätzung und einer Priorität. Sie ist nicht länger als drei Sätze, verwendet keine technischen Begriffe und spezifiziert auch keine Werkzeuge. Auch im professionellen Bereich werden für User Stories häufig Karteikarten verwendet, da diese ein passendes Format haben und gut zu handhaben sind.

Alle User Stories zusammen spezifizieren das finale Produkt. Deshalb bilden diese auch die Grundlage für die Präzisierung des Projektziels und die Verständigung mit dem Kunden und der Projektgruppe. Gemeinsam mit dem Kunden werden anschließend die einzelnen Prioritäten festgelegt, aus welchen hervorgeht, welche User Stories zuerst umgesetzt werden sollen und welche ggf. auch nicht umgesetzt werden (vgl. Abschnitt 2.3).

In unserem „Pengu“-Beispiel helfen die User Stories, das umfangreiche Spiel in mehrere Teilbereiche zu gliedern und aufzuteilen. Weiterhin wird durch User Stories leichter deutlich, ob etwas vergessen wurde. Als Grundsatz sollte für jede Rolle (hier z. B. Pinguin bzw. Wolke) und jede einzelne Aufgabe (hier z. B. Pinguin kann sich bewegen) eine eigene User Story geschrieben werden (vgl. [Wo11]).

| | | |
|--|---|---|
| Titel: Spielfläche | Titel: Pinguin bewegen | Titel: Pinguin fällt |
| Beschreibung: Ein Pinguin steht auf einer Klippe die durch einen Abgrund von einer zweiten Klippe getrennt ist. Zwischen den Klippen ist eine Wolke. | Beschreibung: Der Spieler kann den Pinguin nach rechts und links bewegen. Der Pinguin blickt dabei immer in Laufrichtung. | Beschreibung: Wenn der Pinguin über den Abgrund kommt, fällt er hinunter und das Spiel ist zu Ende. |
| Priorität: 10 | Priorität: 20 | Priorität: 30 |

Abb. 3: Die ersten drei User Stories von „Pengu“

Die ersten drei User Stories dienen in unserem Beispiel der Grundfunktionalität des Spiels: „Spielfläche“, „Pinguin bewegen“ und „Pinguin fällt“. Weitere User Stories beinhalten mögliche Ausbaufunktionen. Abb. 3 zeigt die entsprechenden Beschreibungen der User Stories, sowie die Prioritäten. Da alle drei User Stories für das Spiel grundlegend sind, wurden sehr hohe Prioritäten festgelegt (vgl. Abschnitt 2.2), aus denen auch implizierte Abhängigkeiten ersichtlich sind (ehe das Szenario steht, kann keine Aktivität implementiert werden). Nach der Festlegung der User Stories kann mithilfe des Planungsspiels (Planning Poker) der zeitliche Aufwand der Umsetzung der User Stories abgeschätzt werden. Möglichkeiten zur Umsetzung werden im nachfolgenden Abschnitt beschrieben.

Nach der ersten Zeitabschätzung und Priorisierung werden die User Stories der ersten Iteration in Tasks überführt. Ein Task ist eine grobe Beschreibung eines Arbeitspakets, das von einem Entwickler oder Entwicklerpaar (beim Pair Programming) umgesetzt werden kann. Entsprechend kann eine User Story als eine Sammlung von Tasks angesehen werden. Während User Stories die Ziele des „Pengu“-Spiels aus Sicht des Kunden beschreiben, müssen die Schüler nun ihre Perspektive ändern und die Ziele aus Sicht des Entwicklers betrachten. Beim Aufteilen der User Stories zu Tasks sind bereits verschiedene Designentscheidungen zu treffen. Ein prototypisches Beispiel, wie eine umfangreichere User Story in Tasks überführt werden kann, zeigt Abb. 4. Die Überführung der ersten drei User Stories von „Pengu“ in Tasks ist in Abb. 5 dargestellt. Hierbei wird deutlich, dass die Tasks bereits konkrete Umsetzungsangaben beinhalten, woran die

Entwicklersicht deutlich wird. Tasks dürfen auch konkrete, technische Begriffe oder kurze Quellcodehinweise enthalten. Notiert werden die Tasks in der Regel auf Klebezetteln (Post-Its). Jeder Task erhält eine eigene zeitliche Abschätzung, die wieder über ein Planungsspiel erarbeitet werden kann. Bei größeren Projekten bietet es sich außerdem an, das Namenskürzel der Schüler zu notieren, die diesen Task bearbeiten. Lehrer und Schüler können hierdurch genau sehen, welcher Schüler welchen Tasks übernommen hat. Weiterhin wird es für die Schüler leichter, am Ende eine Soll-Ist-Gegenüberstellung vorzunehmen.



Abb. 4: Aufteilen von User Stories in Tasks

2.2 Prioritäten- und Zeitabschätzung

Die erste Prioritätenschätzung wird in der Regel gemeinsam mit dem Kunden vorgenommen, wofür sich im Unterricht der Lehrer anbietet. Hierbei ist zu berücksichtigen, was dem Kunden besonders wichtig ist und was gegebenenfalls nur ein „nice to have“ wäre. Hier kommt der flexible Charakter des agilen Modells zum Tragen: Entsprechend dem agilen Manifest (vgl. [Be01]) ist es wichtiger, auf Veränderungen reagieren zu können, als nur strikt einem Plan zu folgen. Da – wie jeder Lehrer im Unterricht regelmäßig erlebt – organisatorische, technische oder gesundheitliche Probleme den geplanten Unterrichtsverlauf stören können, wird durch die Festlegung und Umsetzung höher priorisierter Aufgaben ermöglicht, dass selbst bei reduzierter Projektlaufzeit ein lauffähiger und getesteter Prototyp vorhanden ist – und selbst die Prioritäten im Laufe des Projekts noch angepasst werden können. Die Prioritäten werden in Vielfachen von 10 angegeben,

wobei 10 die höchste Priorität hat, bei unserem Spiel ist es die Spielfläche, und nach oben hin die Priorität abnimmt. Verschiedene User Stories können auch die gleiche Priorität besitzen.

Die zeitliche Einschätzung von Tasks wird von den Schülern eigenständig ohne Kunden vorgenommen, es ist aber wichtig, dass alle Teammitglieder in den Schätzprozess mit eingebunden sind. Als Grundlage sollten sinnvolle Zeiteinheiten gewählt werden. Im professionellen Bereich werden Tage als Basis genommen, in Unterrichtsprojekten haben sich Einheiten von 15 Minuten als sinnvoll und gut handhabbar erwiesen. Eine gute Möglichkeit, Vorgaben durch nur wenige erfahrene Schüler zu vermeiden und alle Schüler in die Kommunikationsprozesse einzubinden, stellt das Planungsspiel (Planning Poker, vgl. [PM08]) dar. Hierbei legen sich alle Teammitglieder unter Verwendung eines speziellen Kartenspiels verdeckt fest, welcher Aufwand ihrer Meinung nach für einen bestimmten Task erforderlich ist. Nach Aufdecken der Einschätzungen müssen die extremen Schätzungen begründet werden und es wird versucht, in der Diskussion einen Konsens zu finden. Alternativ wird ein Durchschnittswert der Schätzungen herangezogen. Der Aufwand einer User Story ergibt sich aus den addierten Schätzwerten der einzelnen Tasks. Die Diskussion zur Einschätzung ist gerade für unerfahrene Schüler sehr wichtig, da die Schüler hierbei auch mögliche Strategien austauschen und voneinander lernen können. Gleichzeitig ist dies eine gute Kontrolle, ob alle das gleiche Verständnis von dem haben, was sie mit den Tasks geplant haben. Wenn die Schätzungen auseinander gehen, werden ggf. Unklarheiten in der Diskussion aufgedeckt.

2.3 Iterationen und Prototypen

In Schulsoftwareprojekten stellt sich das Planen und Durchführen langer Projektphasen regelmäßig als Hauptschwierigkeit heraus. Insbesondere ist es schwierig, die Motivation der Schüler aufrecht zu erhalten, wenn über lange Strecken der Erfolg nicht sichtbar wird und das Produkt erst in späten Implementierungsphasen oder zum Ende des Projekts ausprobiert werden kann. Durch die iterative Entwicklung haben die Schüler nun die Möglichkeit, Prototypen des „Pengu“-Spiels in verschiedenen Entwicklungsphasen auszuprobieren und dabei zu überprüfen, ob die Teilziele erreicht wurden.

Im „Pengu“-Beispiel wird die erste Iteration (vgl. Abb. 5) bestimmt durch die Herstellung der Grundfunktionalitäten des Spiels. Der Vorteil der iterativen Vorgehensweise wird unmittelbar deutlich: Bereits früh im Projekt ist das Spiel „spielbar“, der erste Erfolg kann bereits getestet und gegenüber Mitschülern und dem Lehrer demonstriert werden. Sogar das Hinzufügen weiterer Spielideen (Features) ist vor einer Iteration noch möglich, was bei linearen Vorgehensmodellen nahezu ausgeschlossen wäre. Zum Abschluss jeder Iteration ist es möglich, zu reflektieren, wie gut der Prozess bis dahin gelaufen ist und ob sich das Team auf dem richtigen Weg befindet.

In den weiteren Iterationen kommen nun nach und nach, der zuvor bestimmten Priorität entsprechend, weitere Funktionalitäten hinzu, wie sie auf den User Stories festgehalten wurden, z. B. das Bewegen der Wolke, Springen, Überqueren des Abgrunds in Iteration

2, Punktezähler, Sternenhimmel und Sternesammeln in Iteration 3 sowie das Berücksichtigen verschiedener Leben, Gewinnen und Spielende in Iteration 4. Erweiterungen des Spiels, wie bspw. die Eiszapfen, dargestellt in Abb. 4, können auch später noch hinzugefügt werden.

So stellt sich jede Iteration als Mini-Projekt dar, in welchem jede Phase des Softwareentwicklungsprozesses (Anforderungen, Entwurf, Implementieren und Testen) einmal durchlaufen wird und aufgrund der geringeren Komplexität einfacher zu handhaben ist. Hierdurch erhalten die Schüler die Gelegenheit, den gesamten Prozess mehrfach in einem Projekt zu durchlaufen, daran zu lernen, bisherige Erfahrungen zu reflektieren und verschiedene Aufgaben im Team zu übernehmen.



Abb. 5: Tasks für die erste Iteration

2.4 Project Board und Standup-Meetings

Das Project Board ist der zentrale Informations- und Organisationsort des Projekts. Es visualisiert die Ziele und den Status der aktuellen Iteration und unterstützt zielgerichtete Diskussionen anhand der angebrachten User Stories und Tasks. Hier finden sich im Wesentlichen drei Bereiche: Der Vorrat der User Stories mit Tasks, Tasks in Bearbeitung, sowie erledigte Tasks und User Stories. Zu Beginn einer Iteration befinden sich alle Karten auf der linken Seite. Sobald ein Schüler oder Schülerpaar mit einer Aufgabe beginnt, wird ein Post-it mit dem entsprechenden Task aus der Vorratsspalte genommen, mit Namenskürzel versehen und in die Spalte „in Bearbeitung“ gehängt. Sobald der Task bearbeitet wurde, wird er auf „erledigt“ weitergehängt. Ein Beispiel für ein Project Board in der ersten Iteration des „Pengu“-Spiels zeigt Abb. 6.

Am Project Board finden auch die regelmäßigen Standup-Meetings statt, in welchen zu Beginn einer Unterrichtsstunde organisatorische Aspekte der Projektdurchführung be-

Schwierigkeiten ergaben sich regelmäßig aus der Notwendigkeit, die agilen Praktiken zuerst selbst erfassen zu müssen. Wie bspw. User Stories formuliert sein sollen und wie man diese in Tasks überführt, stellte sich mangels Erfahrung im ersten Workshop als eine Herausforderung dar. Mit diesem Beispiel ist nun eine Vorlage gegeben, die zur Orientierung für das Erlernen agiler Praktiken und die Durchführung agiler Softwareentwicklungsprojekte im Informatikunterricht dienen kann.

Literaturverzeichnis

- [Be01] Beck, K.; Beedle, M.; Bennekum, A.v. et al.: Manifesto for Agile Software Development. <http://www.agilealliance.org/the-alliance/>, abgerufen am 10.04.2015.
- [PM08] Pilone, D.; Miles, R.: Softwareentwicklung von Kopf bis Fuß. Beijing: O'Reilly, 2008.
- [RG12] Romeike, R.; Göttel, T.: Agile Projects in High School Computing Education – Emphasizing a Learners ' Perspective. In: Proceedings of the 7th Workshop in Primary and Secondary Computing Education (WiPSCE'12): ACM, Hamburg, 2012.
- [Wo11] Wolf, H.: Agile Softwareentwicklung (2., aktualisierte und erweiterte Aufl). Heidelberg: Dpunkt.Verlag, 2011.

Anhang: User Stories des „Pengu“-Spiels

Titel: Spielfläche

Beschreibung:

Ein Pinguin steht auf einer Klippe die durch einen Abgrund von einer zweiten Klippe getrennt ist. Zwischen den Klippen ist eine Wolke.

Priorität: 10

Titel: Pinguin bewegen

Beschreibung:

Der Spieler kann den Pinguin nach rechts und links bewegen. Der Pinguin blickt dabei immer in Laufrichtung.

Priorität: 20

Titel: Pinguin fällt

Beschreibung:

Wenn der Pinguin über den Abgrund kommt fällt er hinunter und das Spiel ist zu Ende.

Titel: Wolkenbewegung

Beschreibung:

Die Wolke bewegt sich automatisch zwischen den Klippen hin und her.

Priorität: 30

Titel: Pinguin überquert Abgrund

Beschreibung:

Der Pinguin kann auf der Wolke stehen und bewegt sich mit ihr mit. Der Spieler kann ihn auf der Wolke wie auf den Klippen nach rechts und links bewegen.

Priorität: 40

Titel: Pinguin springt

Beschreibung:

Der Spieler kann den Pinguin springen lassen. Wenn der Pinguin fällt reagiert er nicht. Während des Sprungs behält der Pinguin eine anfängliche horizontale Bewegung und seine Blickrichtung bei.

Priorität: 50

Titel: Punktezähler

Beschreibung:

Das Spiel hat einen Punktezähler, den die Welt verwaltet.

Priorität: 55

Titel: Pengu gewinnt

Beschreibung:

Auf der rechten Klippe steht eine Tür. Solange der Spieler mehr als 15 Punkte hat ist sie offen, sonst ist sie geschlossen. Erreicht Pengu die Tür, wenn sie offen ist ertönt eine Fanfare und der Spieler hat gewonnen.

Titel: Sternenhimmel

Beschreibung:

Es erscheinen Sterne am Himmel, stehen dort unterschiedlich lang und verschwinden dann wieder.

Priorität: 60