

Agil und spielerisch: Neue Methoden der Software-Entwicklung in der Praxis und ihr Potential für den Schulunterricht

Dr. Stefanie Scherzinger

Google Germany GmbH
Dienerstraße 12
80331 München
steffis@google.com

„Agile Software-Entwicklung“ bezeichnet eine iterative Vorgehensweise, die sowohl in Silicon Valley Startups als auch großen Software-Häusern erfolgreich eingesetzt wird. Musste nach dem Wasserfall-Modell jede Eventualität im Voraus akribisch bedacht werden, sind agile Ansätze deutlich flexibler: In kurzen Zyklen wird eine einfache, prototypische Lösung zum vollwertigen Produkt ausgebaut. Am Ende jedes Zyklus steht das fertig getestete Produkt; auch wenn es möglicherweise noch nicht über sämtliche gewünschten Funktionalitäten verfügen sollte, ist es für die wichtigsten Zwecke bereits einsatzbereit. Somit können Auftraggeber zeitnah Änderungswünsche einbringen und das Produkt noch in der Entstehungsphase auf das Nutzungsziel ausrichten. Das Entwicklungsteam wiederum kann auf Unvorhergesehenes flexibel reagieren und Zwischenerfolge entsprechend verbuchen. Die kritische Retrospektive spielt dabei eine tragende Rolle.

Methoden wie Agiles Planen, Scrum Meetings und Test Driven Development zählen zu den festen Lehrinhalten an Hochschulen. Dieser Vortrag zeigt den Fortschritt durch die agile Software-Entwicklung auf. Bei der Darstellung der neuen Konzepte wird vor allem auf deren Praxisorientierung hingewiesen. So betrachten wir Beispiele aus dem Arbeitsalltag einer Software Entwicklerin und diskutieren, wie diese Konzepte auch im schulischen Informatikunterricht spielerisch erfahrbar gemacht werden können:

- Bei der Projektplanung spielen Zeitabschätzungen eine wesentliche Rolle. Im Kartenspiel “Planning Poker” pokert das Team um die Aufwände einzelner Funktionalitäten.
- In täglichen Lagebesprechungen, den Scrum Meetings, moderiert ein Scrum Master gezielt den Informationsfluss. Mit Antworten auf einfache Fragen (was habe ich gestern erledigt, woran arbeite ich heute, und wo „hakt es“ bei meiner Arbeit?) wird der Stand des Projekts für das Team erkennbar. Spielerisch kann ein solcher Ablauf am Beispiel gemeinschaftlichen Kochens geübt werden.

- Test Driven Development ist ein bewährtes Prinzip, bei dem erst die Tests und dann der eigentliche funktionale Teil des Projekts programmiert werden. So wird sichergestellt, dass das Programm auch gut zu testen ist. Im Gruppenspiel "Spaghetti Challenge" bauen kleine Teams in kürzester Zeit Türme aus Spaghettis, Bindfaden und Klebeband. Es gewinnt das Team, dessen Turm ein Stück Marshmallow auf dem höchsten Punkt balancieren kann. Doch Vorsicht, wer das scheinbar leichte Marshmallow-Stück erst zuletzt auf den Turm setzt, überschätzt in der Regel die Tragkraft der gebauten Struktur, - wie auch bei der Software Entwicklung die Belastungstests gerne unterschätzt werden.
- Vier Augen sehen mehr als zwei. Nach diesem Prinzip funktionieren Code Reviews, bei denen alle Änderungen am Programm erst von einer zweiten Person akzeptiert werden müssen, bevor sie umgesetzt werden dürfen. Im Pair Programming wird gar zu zweit am Rechner entwickelt. Was zunächst als Einschränkung der Kreativität und als Kontrolle empfunden werden könnte, wirkt in der Praxis aber sehr effektiv. Sobald das Ziel gemeinsam erarbeitet ist, kommt auch der Spaß an der Arbeit nicht zu kurz.

Projektmanagement und die Fähigkeit zur Selbstorganisation sind wichtige Schlüsselkompetenzen in der Berufspraxis. Indem Schüler und Schülerinnen erste Erfahrungen in der Software Entwicklung sammeln, erhalten sie Einblick in die Berufsbilder von Software-Entwicklung und Projektleitung. Unter Betonung der kommunikativen Komponenten der Programmierung möchte der Vortrag nicht nur häufigen Vorbehalten gegenüber der Informatik, wie das negative Image „des sozial vereinsamten Hackers“, entgegen wirken, sondern auch neues Interesse wecken. Zudem erhalten Schüler und Schülerinnen Handwerkszeug für ihr eigenes Zeitmanagement, wie es für die Persönlichkeitsentwicklung wertvoll ist.

Die Referentin greift bei ihren Vorschlägen für den Schulunterricht auf ihren Erfahrungsschatz als Software Entwicklerin bei IBM und Google zurück, sowie auf ihr nebenberufliches Engagement in der fachspezifischen Didaktik. Dazu gehören die Gestaltung von Mädchen-Techniktagen oder Veranstaltungen bei der Informatik-Sommerschule "Informatica Feminale", die sich an Informatikerinnen in Studium und Beruf wenden.