

Peter Brichzin

Agile Softwareentwicklung – Erfahrungsbericht eines Oberstufenprojekts im Wahlpflichtunterricht



Methoden und Rollen bei den Profis



3 Rollen (Wer?)

Team

Scrum Master

Product Owner



4 Ergebnisse (Was?)

Product Backlog

Sprint Backlog

Product Increment

Burn-down Chart



5 Ereignisse (Wie?)

Product Planning

Sprint Planning

Daily Scrum

Sprint Review

Sprint Retro

User Stories	Tasks In Progress	Done	Sprint ends
Ideas			
Problems			

Projekte benötigen Zeit um Irrwege gehen zu können – 26 Std. klingen viel, lassen aber nicht viel Luft.

Inf 11.2 Softwaretechnik (ca. 26 Std.)

Das Arbeiten in Projekten ist die typische Vorgehensweise bei der Entwicklung großer Systeme. Mit den bisher erworbenen Kenntnissen und Fertigkeiten sind die Schüler nun in der Lage, größere Softwaresysteme (z. B. Geschäftsabläufe einer Bank, Autovermietung) eigenständig zu gestalten. Hierbei bauen sie auch ihre Fähigkeit zur Planung und Durchführung von Projekten aus. Die Jugendlichen übernehmen verstärkt persönliche Verantwortung und erfahren die Notwendigkeit, eigene Ansichten und Ideen vor anderen darstellen und vertreten zu können. Sie setzen alle bisher erlernten Beschreibungstechniken der Informatik ein ...

Inf 11.2.1 Planung und Durchführung kooperativer Arbeitsabläufe

Inf 11.2.2 Praktische Softwareentwicklung

Bisher haben die Schüler verschiedene Modellierungstechniken der Informatik einzeln angewandt. Nun erkennen sie, dass eine angemessene Beschreibung größerer Systeme nur durch die kombinierte Verwendung aller bisher erlernten Modellierungstechniken möglich ist. Bei der Implementierung ihrer Modelle setzen sie bekannte Datenstrukturen situationsgerecht ein und achten bei der Gestaltung der Benutzeroberfläche insbesondere auf Benutzersfreundlichkeit.

- Zusammenspiel der verschiedenen Beschreibungstechniken beim Systementwurf: Datenmodellierung – Ablaufmodellierung – funktionale Modellierung – Objektmodellierung
- Implementierung des Systementwurfs unter Nutzung rekursiver Datenstrukturen; Anwendung des Softwaremusters „Model-View-Controller“
- Test der Komponenten und des Gesamtsystems, Überprüfung der Vollständigkeit und Korrektheit des Systementwurfs
- Dokumentation des Softwareprodukts



Lerngruppe ist typisch für den Wahlpflichtunterricht

Stellung der Informatik:	Wahlpflichtfach
Vorwissen:	OOM/OOP 1,75 Schuljahre
Entwicklungsumgebung:	BlueJ (Eclipse, Netbeans)
Schüleranzahl:	23 Schüler
Leistungsstärke:	extrem inhomogen
Stundenzahl pro Woche:	3 Std. (davon eine Doppelstunde)



Die Berücksichtigung von Schülerwünschen bei der Themenwahl erhöht die Motivation, ...

Spiele:

Mario Spiel, Brettspiel TAC, Flappy Stein, Jump ,n‘ Run, Black Jack, Poker, Schafkopf, Space Invaders, Asteroids, Stein, Schere und Papier und Rubiks Cube

Anwendungssoftware (ohne Spiele):

individualisierte Anzeige des Vertretungsplanes auf dem Handy/Computer, Terminverwaltungsprogramm (z. B. mit Hausaufgabenliste), Kassensystem für Pausenverkauf, Mediensammlung, Datenverwaltungssoftware, Nachhilfevermittlung, Bibliothekssoftware für die Lehrmittelbibliothek (zusätzlicher Vorschlag vom Lehrer)



... führt aber meist zu anspruchsvollen Themen.

Spiele:

Mario Spiel, Brettspiel TAC, Flappy Stein, **Jump ,n‘ Run**, Black Jack, Poker, Schafkopf, Space Invaders, Asteroids, Stein, Schere und Papier und Rubiks Cube

Anwendungssoftware (ohne Spiele):

individualisierte Anzeige des Vertretungsplanes auf dem Handy/Computer, Terminverwaltungsprogramm (z. B. mit Hausaufgabenliste), Kassensystem für Pausenverkauf, Mediensammlung, Datenverwaltungssoftware, Nachhilfevermittlung, **Bibliotheksoftware für die Lehrmittelbibliothek** (zusätzlicher Vorschlag vom Lehrer)



„Selbstorganisation der Teams bei Planung und Umsetzung“ ist ein agiles Prinzip.



3 Rollen (Wer?)

Team

Scrum Master

Product Owner

Team:

Gruppengröße: 6-8

Organisation: Paare

→ Pair Programming

!Wichtig! Das Team organisiert sich selbst!

Lehrer übernimmt die Aufgabe des Scrum Masters
Hindernisse (impediments) aus dem Weg zu räumen

User Stories	(Task) In Progress	Done	Sprint ends
Task 1	Task 2	Task 3	
Task 4	Task 5	Task 6	
Task 7	Task 8	Task 9	
Task 10	Task 11	Task 12	

Sprint als Iterationsintervall wie bei den Profis



Sprint ist ein Zeitintervall für einen Iterationsschritt

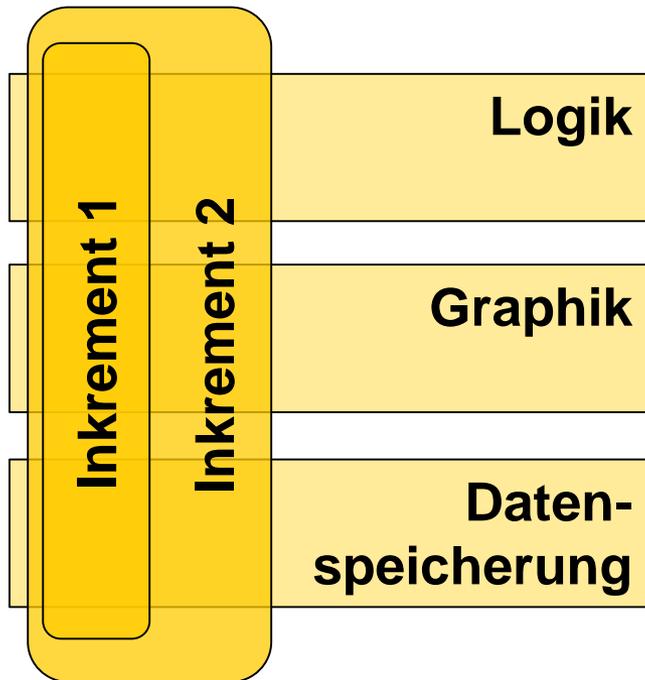
- entspricht einem Meilenstein
- beginnt mit einer Planung
- Schließt mit einem Prototyp ab
- (hat am Ende eine Reflexionsphase)

Die Sprintdauer wird i.A. einmal festgelegt und beibehalten

Info 11 – 2 Wochen = 6 Std.



Prototyping ist zentraler Erfolgsfaktor in Hinblick auf Reduzierung, Erfolgserlebnisse und Schnittstellen.



Nach jedem Sprint (iterativer Prozess) gibt es einen lauffähigen und getesteten Prototyp, der mehr Funktionalitäten bietet, als der vorhergehende Prototyp (inkrementell)

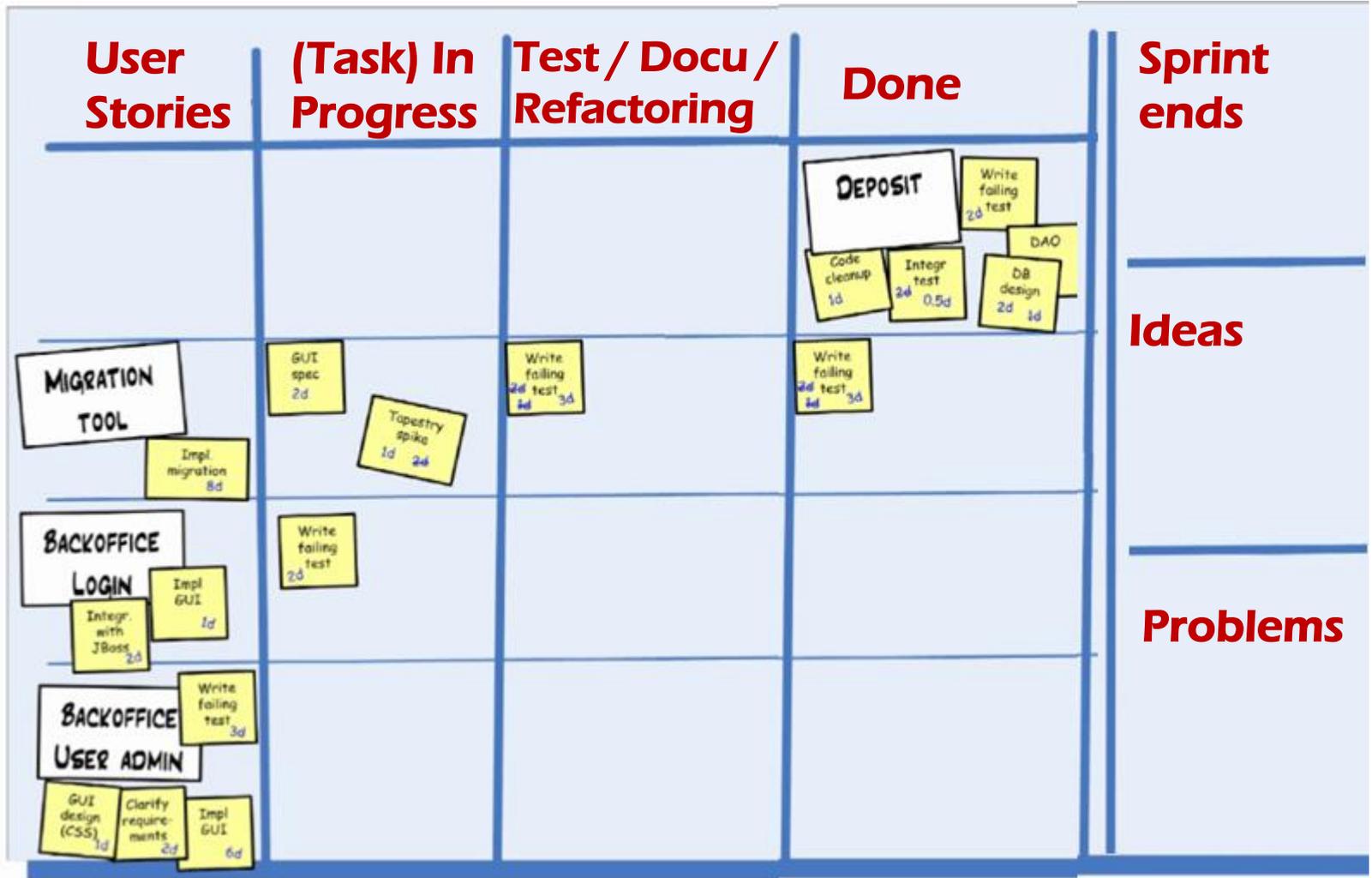
Vorteile

- Kunde und Entwickler sehen das Produkt wachsen
- Schnittstellenproblematik wird vermieden
- Produktausrichtung und Reihenfolge der umzusetzenden Funktionalitäten kann im laufenden Prozess verändert werden, die Entwicklung bleibt beweglich (lat. agilis: flink, beweglich)



Das Taskboard ist das zentrale Kommunikationsmittel - kleine didaktische Anpassungen lenken.

absteigende
Priorität





Modeling Story – Der Lehrer nutzt User Stories um zu lenken

Titel:

Klassendiagramm

Beschreibung:

*Erstellt ein
Klassendiagramm
um einen Überblick
über Eure Klassen zu
erhalten. Legt ein
besonderes
Augenmerk auf die
Beziehungen.*

Priorität: 1

**Eine lange Planungsphase zu
Beginn ist nicht motivierend
und überfordert Schüler.**

**Mit zunehmenden
Projektfortschritt fehlt der
Schülern der Überblick.**

**Der Nutzen von Diagrammen
ist sichtbar.**



Tasks – der Blick ins Innere

Titel:

Lebensenergie

Beschreibung:

Wenn ein Avatar mit einem giftigen/ energispendendem Element in Berührung kommt, werden ihm Lebensenergiepunkte abgezogen/ addiert, um das Spiel attraktiver zu machen.

Priorität: 5

Titel:

Energieelement

Beschreibung:

ENERGIEELEMENT

punkte

// neg. Wert bei gift. Elem.

PunkteGeben()

PunkteSetzen(punkteNeu)

Bearbeiter: *Bernd & Peter*

Titel:

Lebensenergie

Beschreibung:

Ergänzung eines Attributs lebensenergie mit Getter und Setter Methode in der Klasse Avatar

Bearbeiter: *Birgit & Paola*

Titel:

Anzeige Energieelement

Beschreibung:

graph. Anzeige von Energieelementen

Bearbeiter: *Evelyn & Andi*

Titel: *Ereignis Berührung Energieelement*

Beschreibung:

Punktanzug/-addition bei Berührung

Bearbeiter: *Silke & Tom*

Titel:

Anzeige Lebensenergie

Beschreibung:

Anzeige der Lebensenergie mit einem entsprechendem Update, wenn sich der Wert ändert

Bearbeiter:



Nicht verwendete agile Methoden waren:

Aufwandsaufschätzung (Planning Poker, Burn-down Chart) wird reduziert auf die Anzahl an Tasks

Retrospektive – Einsteuerung durch Fragen des Lehrers bei der Sprintplanung



Projektergebnisse

Spielentwicklung

Ein ausreichendes Ergebnis

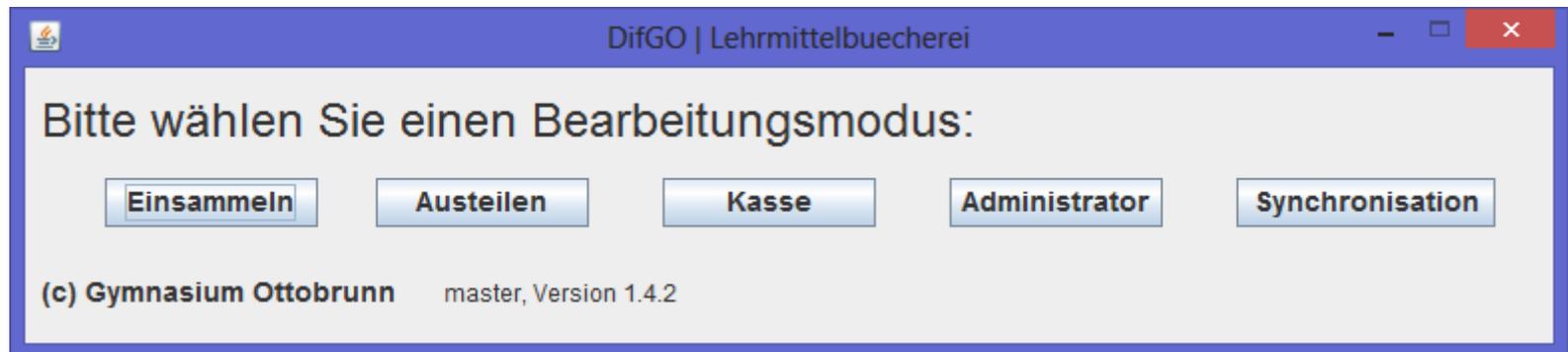
Problem: Mangelnde Kommunikation und Teamarbeit

Personalisierter Vertretungsplan:

Herausragendes Ergebnis mit sehr gutem internen Wissenstransfer ohne Hilfestellungen des Lehrers

Bibliotheksoftware:

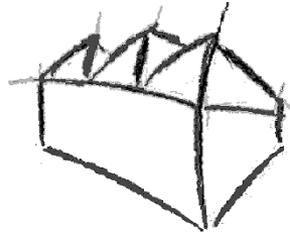
Mit Engagement und Hilfestellungen des Lehrers ein sehr gutes Ergebnis





Und dieses Schuljahr ...

- Einführung und Nutzen verschiedener Techniken wie User Stories, Tasks, Pair Programming schon im nicht projektorientierten Unterricht
- Unterstützung des kollaborativen Arbeitens durch Repositories
- Massiveres Eingreifen durch den Lehrer bei Fehlentwicklungen



Peter Brichzin

Agile Methoden kommen (nicht nur) bei Softwareprojekten im Informatikunterricht der Arbeitsweise von Schülerinnen und Schülern entgegen und sind ein echter Mehrwert im Hinblick auf die Unterstützung der Selbstorganisation.