

Softwareentwicklungspraktikum für Fortgeschrittene

Einführungsveranstaltung

Prof. Dr. Martin Wirsing, Prof. Dr. Rudolf Haggenmüller,
Dr. Andreas Schroeder, Christian Kroiß



I. Was wir bieten

- Produkt
- Prozess
- Technologien

II. Was wir erwarten

- Fundierte Programmierkenntnisse
- Eigenverantwortung
- Zeit

III. Ablauf des Praktikums

- Zeitplan
- Benotung
- Test

Teil I. Was wir bieten

■ Ziele

- Modernes Produkt
- Spaß
- Klare Anforderungen

■ Das Produkt: „The Bug Is A Lie“

- Ein Kartenspiel, welches sich in satirischer Weise der Softwareentwicklung in der Industrie nähert
- Eigenentwicklung des Lehrstuhls
- Umsetzung als **Multiplayer-(„Roleplaying“)-Onlinespiel**





■ Spielablauf

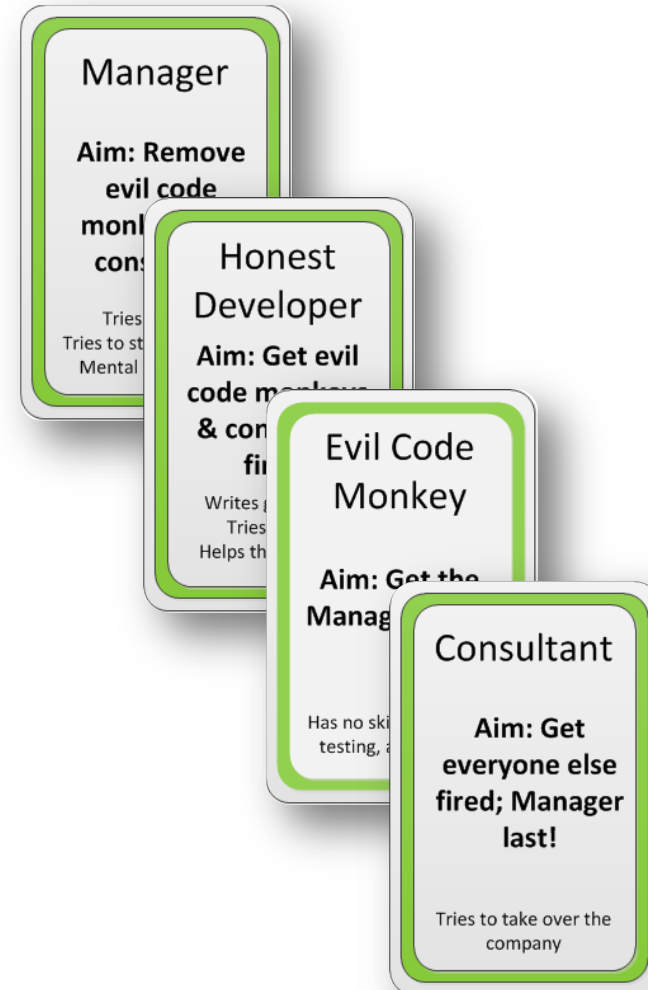
- Spieler schieben sich gegenseitig **bug reports** zu, die mittels **solution** bearbeitet oder mit einer **lame excuse** abgelehnt werden müssen.



- Spieler besitzen eine gewisse Menge **mental health**; diese verringert sich bei nicht bearbeiteten bug reports. Bei **mental health == 0** wird der Spieler gefeuert.

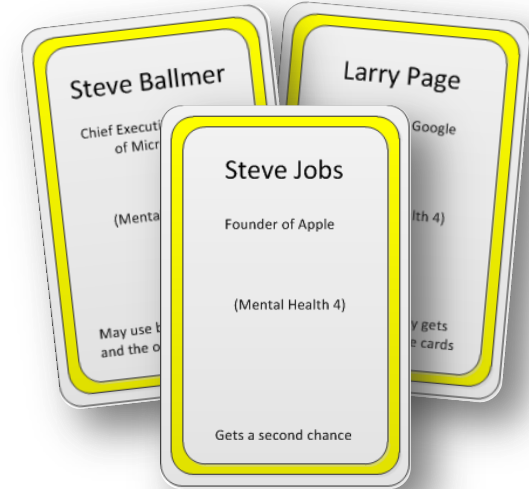


- Jeder der 4-7 Spieler erhält eine Rolle.
 - **Manager**
 - **Honest Developer**
 - **Evil Code Monkey**
 - **Consultant**
- Rollen (bis auf **Manager**) sind am Anfang unbekannt.
- Ziel des Spiels
 - Manager und Honest Developers: Evil Code Monkey und Consultant feuern
 - Evil Code Monkeys: Manager feuern
 - Consultant: Projekt übernehmen (= > alle anderen feuern)





- Zusätzliche **Persönlichkeitskarten** geben Sonderfähigkeiten.
- Verschiedene Zusatzkarten wie Prestige, Fortbildungen etc. fügen zusätzliche Aspekte hinzu.





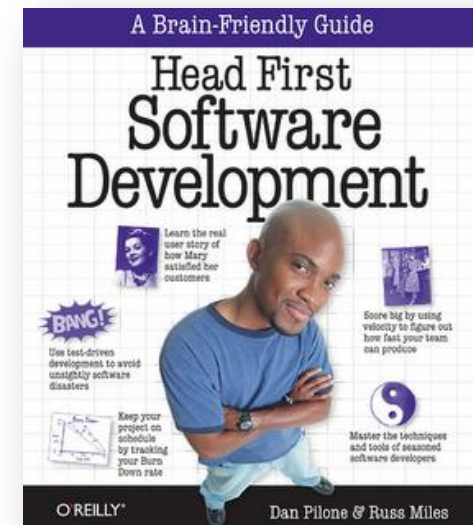
- Detaillierte Regeln werden am 18.04. geliefert.
- **Umsetzung** erfolgt
 - als Web-basiertes Onlinespiel
 - Lobby, Spieltisch, Chat, ggf. Statistiken
 - Implementierung des Browser-Clients mittels Wicket, jQuery, html 5 und CSS
 - Implementierung des Servers mit Java Servlets (Anbindung an UI durch Wicket)

■ Ziele

- Die Idee ist, den Sinn eines Software-Entwicklungsprozesses zu erkennen.
- Der Prozess ist daher leichtgewichtig und einfach zu erlernen.
- Erlernen von Prozess-Techniken (Planung, Testen, Bugtracking, ...)

■ Der eingesetzte Prozess ist ein **agiler Softwareentwicklungsprozess**

- Bekannte Vertreter: Scrum, XP, Kanban
- Hier: Scrum + XP-Elemente
- Vorstellung am 15.04.;
Buch dazu ist (sehr!) empfohlen
- Einübung in einer Lego4Scrum Session am 22.04.





- Prozessablauf
 - Der Prozessablauf ist in vier **Iterationen** (2-3 Wochen) und ein **Release** (3 Monate) aufgeteilt.
 - Jede Iteration und jedes Release produziert einen **voll lauffähigen Zwischenstand** (wenn auch natürlich noch nicht mit kompletter Funktionalität versehen).
- Handling von Funktionalität
 - Umzusetzende Funktionalität wird in **User Stories** beschrieben.
 - User Stories werden **priorisiert** und **geschätzt**.



■ Teamwork

- Das Team organisiert sich im Prozess selbst.
 - Volle Autorität über Planung und Umsetzung
 - Aber auch volle Verantwortlichkeit für das Produkt
- Offenheit und Ehrlichkeit prägen die Kommunikation.
- Das Team hat Scrum Master (Coach) und Product Owner (Kunde) als Ansprechpartner.



- **Ziel:**
 - Einsatz **produktiver** Werkzeuge (IDEs, Git, ...)
 - Einsatz **moderner** Technologien (Wicket, Mockito, ...)
- Im Speziellen:
 - **Eclipse** als IDE und Build-System
 - **Technologien:**
 - Java 7
 - Wicket, jQuery
 - JUnit, Mockito, JaCoCo (Testing, Mocking, Coverage)
 - Selenium (System-Tests)
 - **Verwaltungstools**
 - Git, Redmine (Issue-Tracker), Jenkins (CI), Mylyn



- Bootstrapping
 - Ein Code-Spike für wird am 18.04. bereitgestellt.
 - Infrastruktur (Git, Redmine, Jenkins und Eclipse) wird zur Verfügung gestellt.
- Einführung in Techniken, Technologien und Tools
 - Die Technologien werden am 18.04. und 25.04. in eigenen Vorlesungsstunden vorgestellt.
 - Eine Coding Session am 2.05. führt in das Codegerüst ein.
 - Zusätzlich gibt es Sessions zu OO und agile Designprinzipien.

Teil II. Was wir erwarten



- Fundierte Programmierkenntnisse in Java
 - Dies ist kein Java-Kurs!
 - Wir erwarten solide Java-Kenntnisse.
 - Sie müssen in der Lage sein, Funktionalität umzusetzen und sich mit neuen Bibliotheken vertraut zu machen.
- Begründung
 - Fokus soll auf neue Technologien und dem Prozess liegen, nicht auf der Sprache.
 - Niveau der Teilnehmer soll ungefähr gleich und adäquat für die Aufgabe sein.
 - Teamarbeit sollte nicht zu stark durch Unterschiede in der Vorbildung beeinflusst werden.

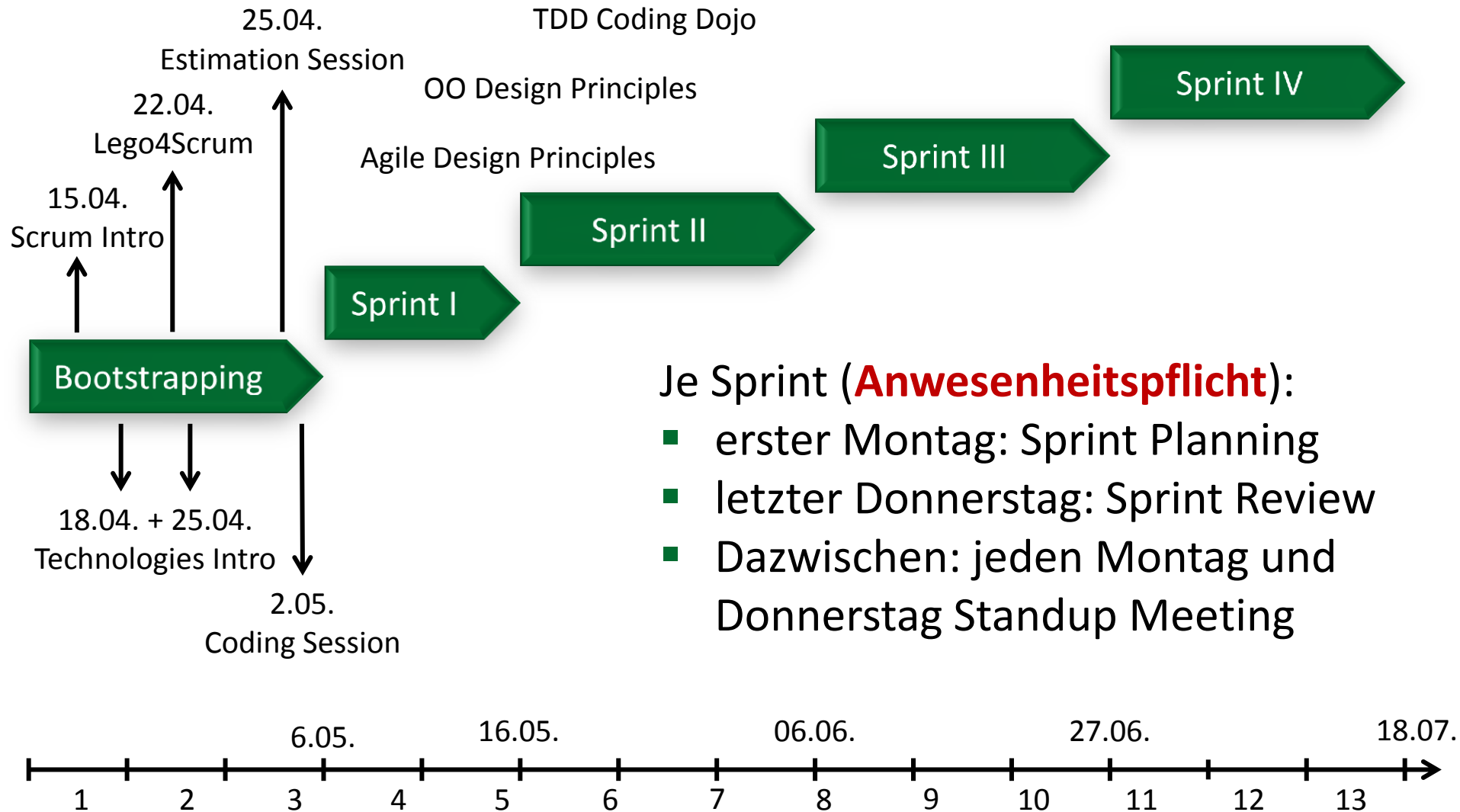


- Wir erwarten **Interesse am Prozess** (und Produkt).
- Darüber hinaus erwarten wir
 - Befolgung der Regeln des Prozesses (nur so lernen Sie etwas)
 - **Selbstmanagement** des Teams:
Das Team setzt sich seine Ziele selbst und überwacht die Zieleinhaltung
 - Übernahme von **Verantwortung für das Produkt**
 - **Offenheit** und **Respekt**: Kommunikation von Problemen und Erfolgen, respektvoller Umgang mit anderen Team-Mitgliedern
- Wichtig: die Betreuer sind sowohl **Kunde** (Product Owner) als auch **Berater** (Scrum Master).



- Zeitmanagement
 - Zeitplanung: Ehrliche Schätzungen der Zeitaufwände
 - Zeiteinsatz: Nutzung nur der vorhandenen Arbeitszeit
 - Dies sind gemäß der ECTS 13 Stunden (=1,6 Tage) pro Woche
 - **Wir erwarten den Einsatz dieser 13 Stunden!**
(aber auch nicht mehr – das ist auch eines der Ziele des Prozesses)
 - Controlling: Tracking der eingesetzten Zeit und Korrektur eigener Schätzungen

Teil III. Ablauf des Praktikums





- Zwei Teams zu jeweils 4-7 Teilnehmern
 - Beide Teams entwickeln gemeinsam am gleichen Produkt.
 - Die Teams verfolgen unterschiedliche Sprint-Ziele (= Produkt-Teilziele).
- Wöchentliche Meetings
 - 2 Standup Meetings (15 Minuten, **Anwesenheitspflicht**)
 - 1 Scrum of Scrums Meeting (15 Minuten, 2 Teilnehmer je Team)
 - Jede Woche 30-Minuten Sessions zu gewählten Themen
 - Abstrakt: Scrum, Design Patterns, Testen, Dokumentation, Tools, ...
 - Konkret: Design-Sitzungen, Team-Besprechungen, Code-Reviews, ...



Nicht Umfang, sondern Vorgehensweise zählt

1. Umsetzung des Prozesses

- **Ehrlichkeit** und **Offenheit**
- Zeitmanagement
- Projektplanung
- Berücksichtigung aller Schritte bei der Umsetzung (Entwurf, Test, Review, Integration, Dokumentation)
- Vollständiger Iterationsabschluss

2. Vorgehen in der Entwicklung

- Architekturentwurf und Systembeschreibung
- Angemessene Testabdeckung
- Dokumentation des Codes
- Code-Qualität (Verständlichkeit und Änderbarkeit)



- Ziele des Test
 - 1) Erfüllen Ihre Programmierkenntnisse unsere Anforderungen?
(bezüglich Java, Nutzung von Bibliotheken, IDEs, ...)
 - 2) Wo stehen unsere Teilnehmer insgesamt?
(Feedback für uns)
- Wir erwarten keine perfekte, sondern eine **gut designte** Lösung.
- Dauer des Tests: 60 Minuten
- Platzvergabe
 - Erfolgt nach Anmeldezeitpunkt
 - Bedingung ist das erfolgreiche Bestehen des Tests
- Benachrichtigung: Samstag per E-Mail (UniWorX-Anmeldung!)

Zusammenfassung



- Überlegen Sie sich,
 - ... ob das Praktikum für Sie interessant ist
 - ... ob Sie unsere Erwartungen erfüllen

- **15 Minuten Pause**

- Danach: Test!