

Programmieren in Java

SS 2019

Organisation

- ▶ Moodle <http://lms.fh-kiel.de/>
 - ▶ FB IuE/Bachelor/Programmieren in Java SS19
 - ▶ Gruppenwahl/Teamwahl
 - ▶ Projektbeschreibung
 - ▶ Materialien zum Labor
 - ▶ Abgaben zum Labor
 - ▶ Ankündigungen
 - ▶ Moodle-Benachrichtigungen werden an Ihre FH-Mail-Adresse verschickt!

Struktur der Veranstaltung

▶ Vorlesung

- ▶ Inhalte, die Hilfestellung für das Labor bieten

▶ Labor

- ▶ Bearbeitung eines Projekts im Team (6-8 Studierende)
- ▶ 9 Wochen a 4 SWS
- ▶ 2 Labortermine pro Woche ab 4. VL
 - ▶ 1 Labortermin: Mo 5. oder 6. Block
 - ▶ 1 weiterer Labortermin am Mi 1. Block
 - ▶ Zeitgleich für beide Laborgruppen

Ergebnis

► Modulnote

- Protokolle der Teamsitzungen 40%
 - Anwesenheit (3x darf man im Labor fehlen)
 - Kurzer Bericht:
 - Wer hat was getan? Welche Hindernisse gab es? -> jeder berichtet was
 - Projektplanung:
 - Wer wird was tun? -> jeder tut was
- Abgabe Projektdokumente/Programm 40%
 - Qualität, Vollständigkeit
 - Termineinhaltung
 - Eigene Entwicklungsarbeit
- Endpräsentation 20%
 - Alle präsentieren

Laborübung – Ablauf

▶ Laborübungen

- ▶ Teamsitzung in jeder Übung
 - ▶ Verteilung der Teams auf Räume C12-2.70/2.69/2.01/1.91
- ▶ Programmieren (im Labor)

▶ Letzte Laborübung

- ▶ Abschlusspräsentation
 - ▶ **alle** Gruppenmitglieder präsentieren
 - ▶ findet am Mittwoch, 19.06.2019 statt

Laborübung - Teamsitzungen

▶ In den Sitzungen:

- ▶ Kurzer Bericht (15min):
 - ▶ was hat jeder getan
 - ▶ welche Hindernisse gab es
- ▶ Gemeinsame Zeitplanung des Projekts
- ▶ Gemeinsame Analyse/Entwurf:
 - ▶ Use Cases, Aktivitätsdiagramme, Klassendiagramm, ...
- ▶ Aufteilung der Aufgaben auf die Teammitglieder

Heute

- ▶ Einteilung/Übersicht der Laborgruppen
 - ▶ 26 Studierende
 - ▶ Verteilung abhängig von Montag-Termin
 - ▶ Mo 5. Block
 - ▶ Mo 6. Block
- ▶ Abgabedokumente/Termine
- ▶ Vorstellung des Projektthemas
- ▶ Entwicklungsumgebung + weitere nützliche Tools
- ▶ Entwicklungsprozess
- ▶ Vorstellung der Inhalte der Vorlesung
 - ▶ Was kann ich? Was sollte ich noch wiederholen?
 - ▶ Was wird noch erklärt werden?
- ▶ Vorstellung ausgewählter Kapitel

Abgabedokumente -> Termine

- ▶ Use Cases -> 15. KW
- ▶ Aktivitätsdiagramme -> 16. KW
- ▶ Problembereichsmodell (Klassendiagramm) -> 16. KW
- ▶ Produktbeschreibung -> 24. KW
- ▶ Sequenzdiagramme -> 24. KW
- ▶ Analysemodell (Klassendiagramm + GUI-Klassen) -> 24. KW
- ▶ Designmodell (Beschreibung der Klassen mit Javadoc) -> 24. KW
- ▶ Klassentests, Produkttests 24. KW
- ▶ Fertiges Programm 19.06.19 (25.KW)

Die Abgaben sind in Moodle hochzuladen!

Projektthema

- ▶ Tool zur Zeiterfassung
- ▶ Heute Kurzvorstellung
- ▶ detaillierte Aufgabenbeschreibung in Moodle
- ▶ Alle Teams haben das gleiche Projektthema

Projekt Zeiterfassung

- ▶ Übersicht Arbeitszeit
- ▶ Entwickler mit mehreren Projekten parallel beschäftigt
- ▶ wie viel Zeit für welches Projekt
- ▶ Beginn der Arbeit/Aufgabe wird vom Benutzer „gestartet“ ; Beenden wird vom Benutzer gestoppt
- ▶ Projekte
 - ▶ sind Bereichen zugeordnet, haben Unterprojekte
 - ▶ Bereich „FH“, Projekt „Zeiterfassung“, Aufgabe „Aktivitätsdiagramm erstellen“
- ▶ graphische Darstellung der verwendeten Zeit
- ▶ Gegenüberstellung geschätzte/geleistete Arbeitszeit
- ▶ Stundenzettel
- ▶ Rechnungen mit unterschiedlichen Stundensätze
- ▶ Export/Import der Daten



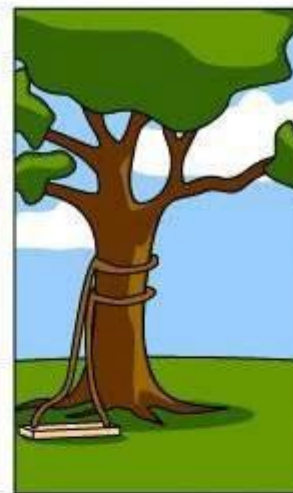
Was der Kunde erklärte



Was der Projektleiter
verstand



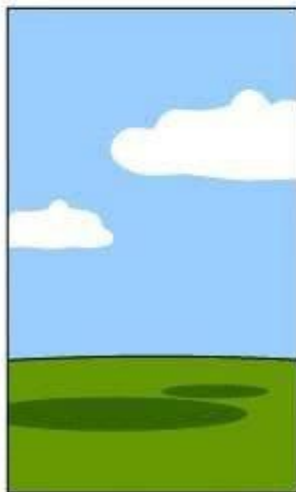
Wie es der Analytiker
entwarf



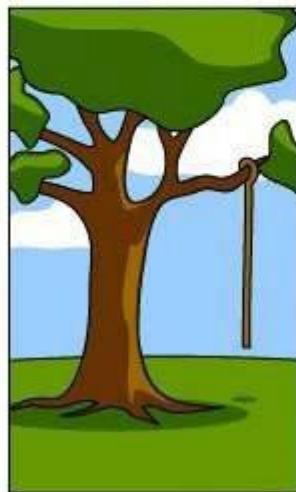
Was der Programmierer
programmierte



Was der Berater definierte



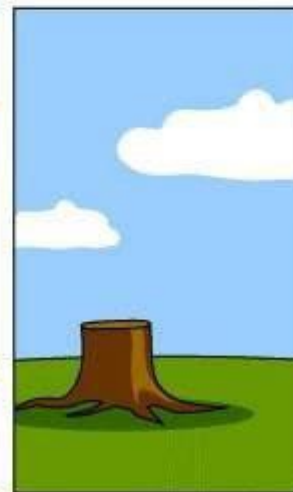
Wie das Projekt
dokumentiert wurde



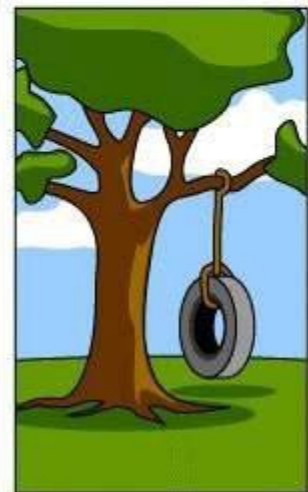
Was installiert wurde



Was dem Kunden in
Rechnung gestellt wurde



Wie es gewartet wurde



Was der Kunde wirklich
gebraucht hätte

Entwicklungsprozess

► Scrum als agiler Prozessrahmen

- Kenntnisse aus Modul Agile Softwareentwicklung nützlich
- Literatur: „Der Scrum Guide“
<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-DE.pdf>

► Warum Scrum?

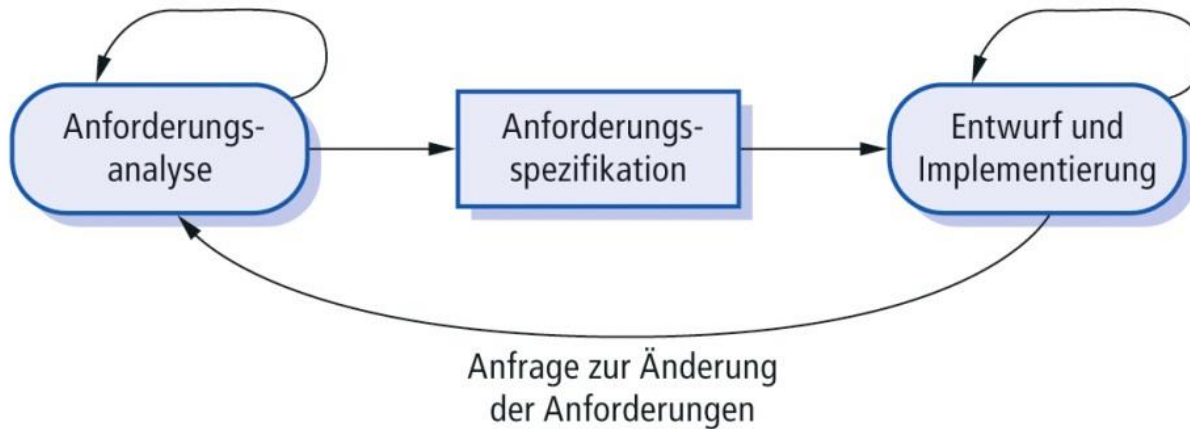
- Leichtgewichtiger Rahmen
- Schnelle Ergebnisse
- Einfach zu erlernen
- Guter Startpunkt für agiles Vorgehen

Entwicklungsprozess

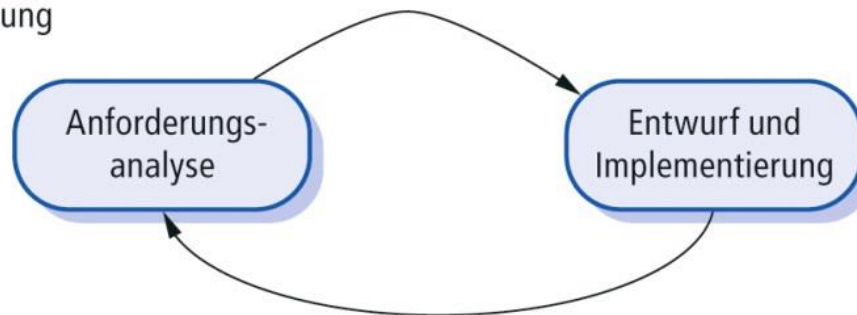
- ▶ Scrum angepasst an unsere studentische Veranstaltung
 - ▶ Wöchentliche Treffen (2 mal in der Woche)
 - ▶ Arbeit zu Hause
 - ▶ Zeitanteil nicht ausschließlich für dieses Modul
 - ▶ Literatur:
 - ▶ „Adaption von SCRUM für verteilte Teams in Teilzeitarbeit“ in Informatik-Spektrum 05/2017, Springer Verlag
<https://link.springer.com/content/pdf/10.1007%2Fs00287-016-1003-4.pdf>
 - ▶ Die Abgabedokumente geben zum Teil ein planbasiertes Vorgehen vor

Plangesteuerte und agile Entwicklung

planbasierte Entwicklung



agile Entwicklung



Scrum

► Merkmale

- Teams sind selbstgesteuert
- Produkt schreitet in Sprints (zeitl. Zeitspanne) voran
- Sprints haben eine konstante Dauer
- Anforderungen sind im Product Backlog festgehalten
- Priorisierung der Anforderungen im Sprint
- Keine Anforderungsänderungen während des Sprints
- Innerhalb eines Sprints sind Teilfunktionalitäten realisiert
- Keine spezifische Entwicklungsmethode

Scrum

► Rollen

► Product Owner

- erfasst Bedürfnisse der Kunden
- passt Prioritäten für jeden Sprint an

► Scrum Master

- verantwortlich für die Einhaltung der Scrum-Technik
- Initiiert die Selbstorganisation des Teams
- moderiert die Meetings
- arbeitet nicht inhaltlich (in unserem Projekt schon)
- Keine Projektleitung

► Team

- Teammitglieder übernehmen selbst die Aufgaben und bekommen sie nicht zugeteilt

Scrum – Meetings

▶ Sprint Planung

- ▶ Team wählt User Stories aus dem Product Backlog aus
- ▶ Sprint Backlog wird erstellt
- ▶ Tätigkeiten gemeinschaftlich formuliert und geschätzt 1-8 h

▶ „Daily“

- ▶ Jeder berichtet kurz:
 - ▶ Was hat man seit dem letzten mal geschafft?
 - ▶ Was wird man bis zum nächsten mal tun?
 - ▶ Welche Hindernisse sind da?

Scrum – Meetings

▶ Sprint-Reviewmeeting

- ▶ Team präsentiert, was es im Sprint erreicht hat (Demo/Modell)

▶ Sprint-Retrospektive

- ▶ Frage an jeden
 - ▶ Was lief gut?
 - ▶ Was kann verbessert werden?
- ▶ Diskussion der Probleme

Entwicklung

▶ Werkzeuge

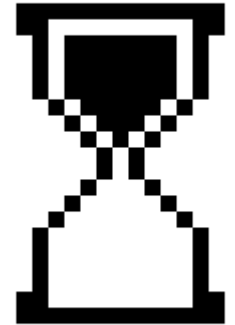
- ▶ Confluence /Jira /Bitbucket
 - ▶ Teams/Mitglieder müssen eingerichtet werden
- ▶ Eclipse
- ▶ ...

▶ Hilfsmittel

- ▶ Beispielprojekt in Confluence

Was kann ich? Was wird erklärt?

Begriff	kann ich	wird erklärt
Grundkenntnisse OOP	x	
Grundkenntnisse Java	x	
Grundkenntnisse UML	(x)	x
Packages		x
Generische Typen		x
Collections		x



Was wird noch erklärt werden?

Begriff		wird erklärt
Javadoc		x
GUI (Swing/ JavaFX)		x
Model-View-Controller		x
Ereignisbehandlung		x
Java<->XML		x
Testen/ JUnit		x

