

PSE „Studienplanung als Generierung von Workflows mit Compliance-Anforderungen: Planerstellung und Visualisierung“

*Jutta Mülle, muelle@kit.edu
IPD, Lehrstuhl Prof. Dr. Klemens Böhm*

Tutor: Simon Dürr, simon.duerr8@gmail.com

Überblick

- Motivation und Aufgabenstellung
- Vorarbeiten
- Detaillierung der Aufgabenstellung
- Organisatorisches
 - Teams
 - Zeitplan und Termine
 - Tools
- Phasen
- Phase 1: Pflichtenheft erstellen
- Was steht an?

- **Motivation und Aufgabenstellung**
- Vorarbeiten
- Detaillierung der Aufgabenstellung
- Organisatorisches
 - Teams
 - Zeitplan und Termine
 - Tools
- Phasen
- Phase 1: Pflichtenheft erstellen
- Was steht an?

Praxis der Software-Entwicklung (PSE)

- Ziel: Entwicklung eines mittelgroßen Systems im Team mit objektorientierter Softwaretechnik
- Zielsystem: max 10kLOC
- objektorientierter Entwurf (UML),
- Implementierung (Java/C++/C#): Java
- Qualitätssicherung (z. B. Jcov, Junit)

- Umfang: 8 LP,
ca. 270 Arbeitsstunden / Teilnehmer,
ca. 2 Arbeitstage / Woche / Teilnehmer

Motivation

- **Ausgangsbasis:** Manuelle Studienplanerstellung, umfangreiche Modulhandbücher und einzuhaltende Regelungen.
- Studienpläne beschreiben den vorgesehenen Studienablauf:
 - Workflows zur Modellierung von Abläufen.
 - Viele Änderungen der Modulhandbücher und Regelungen, sowie Neu- oder Umplanungen der Studierenden.
- Compliance von Studienplänen mit Modulhandbüchern, Regelungen und Zielen des Studierenden erforderlich.
- **Probleme:** große Anzahl Module und Constraints, sowie häufige Änderungen.

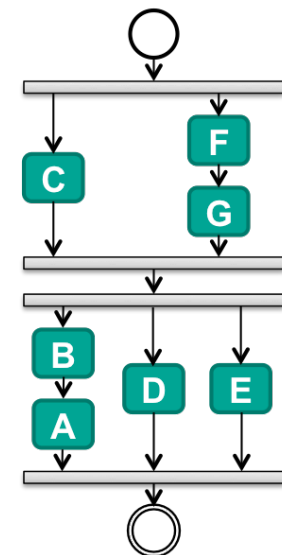


Existierende Methoden zur Prozessverifikation und Prozessgenerierung nutzen.

Prozess-
Generierung:

Abhängigkeiten

- Vorlesung X führt zu Praktikum Y
- Z vor (ist-Voraussetzung-für) X
- ...



Aufgabenstellung

- **Entwicklung einer Web-basierten Benutzeroberfläche eines Systems für die Studienplanung**
 - Werkzeuge zur Generierung von Workflows und zur Prozessverifikation von Constraints (Eigenschaften und Ziele) sind vorgegeben.
 - Sie konzipieren und realisieren für einige ausgewählte Fälle Studienplanungen mit Modulen, Lehrveranstaltungen, Constraints und Planungszielen.
 - Lösung soll modular sein, um Einbindung neuer Algorithmen und Tools zu ermöglichen.
- **Kick-Off-Treffen:**

Mittwoch 9.11.2016, 15:45 Uhr, Informatikgebäude 50.34, Raum 010
- **Weitere Infos** unter dbis.ipd.kit.edu/2436.php

- Motivation und Aufgabenstellung
- **Vorarbeiten**
- Detaillierung der Aufgabenstellung
- Organisatorisches
 - Teams
 - Zeitplan und Termine
 - Tools
- Phasen
- Phase 1: Pflichtenheft erstellen
- Was steht an?

Inbetriebnahmeabläufe in der Automobilindustrie

Automatische Verifikation und Generierung

- *Ausgangsbasis:* Manuelle Erzeugung von Abläufen zur Überprüfung von Fahrzeugkomponenten
- Abläufe beschreiben welche Arbeitsschritte zum Überprüfen/Konfigurieren eines Fahrzeugs notwendig sind
- *Problem:* Wachsende Komplexität und Anzahl der Inbetriebnahmeabläufe

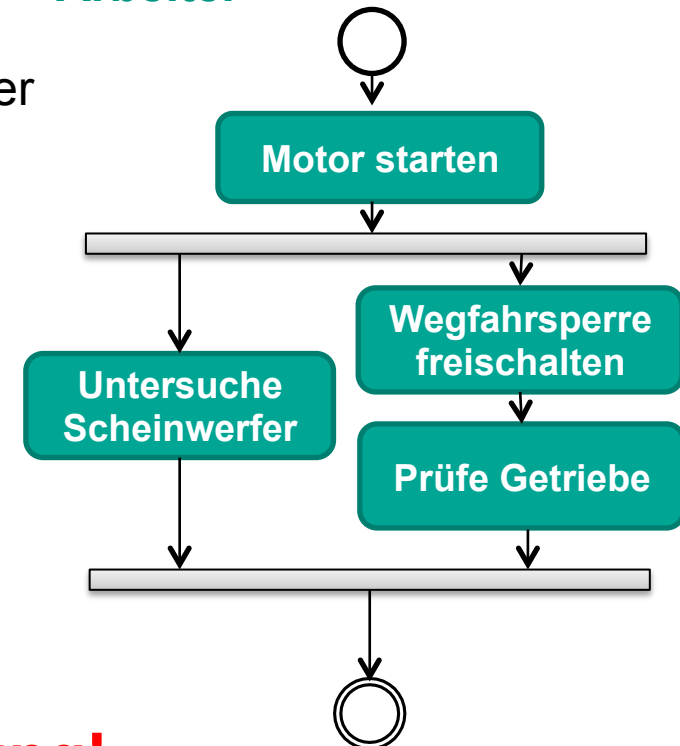
➡ Fehler und aufwändige Entwicklung von Testworkflows

Herausforderungen

- Wissen über die Abläufe ist häufig nur in den Köpfen von Mitarbeitern vorhanden
- Komplexität der Inbetriebnahmeabläufe

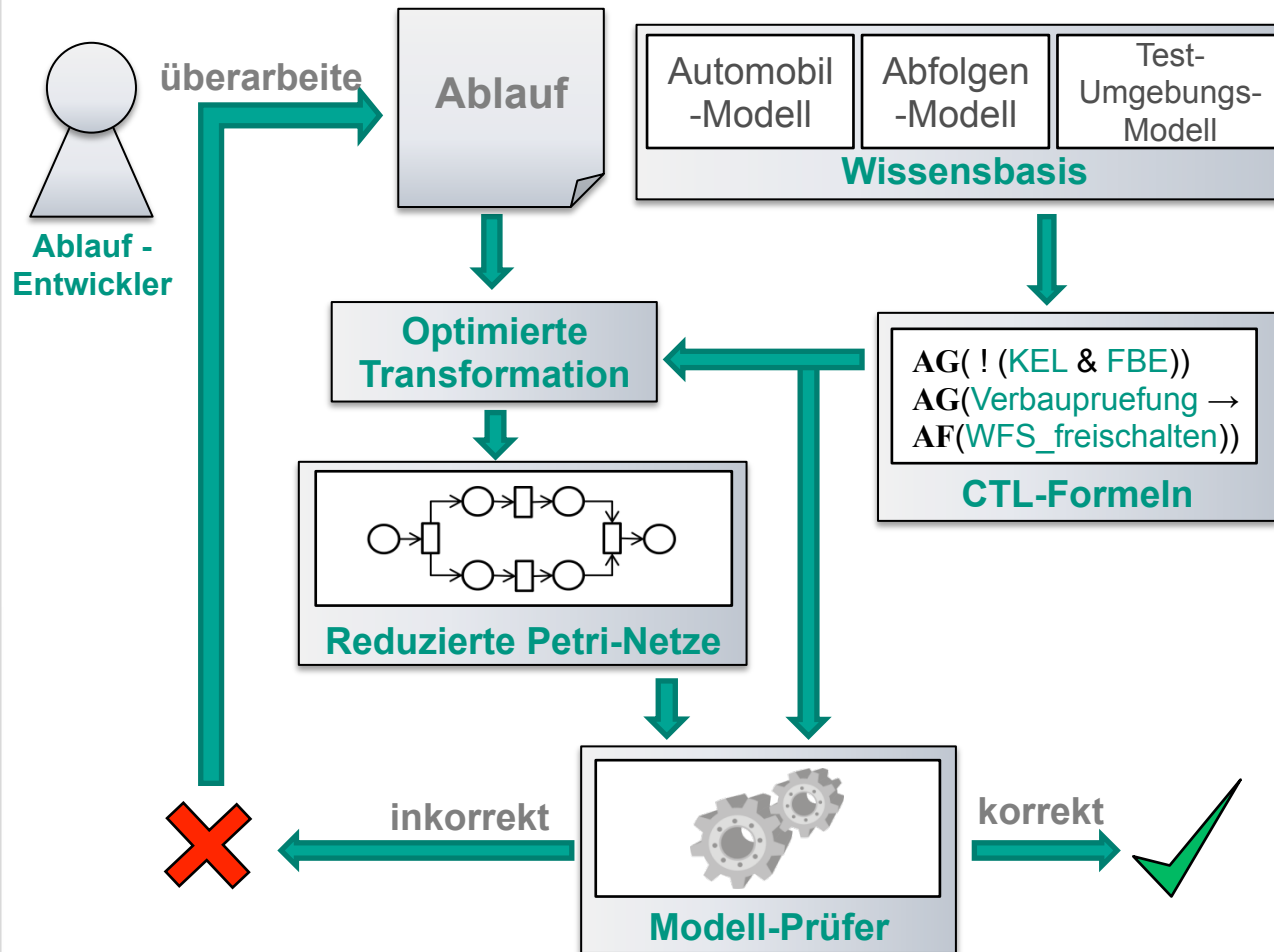


Arbeiter



Anwendung auf Studienplanung!

Verifikation von Inbetriebnahmeabläufe



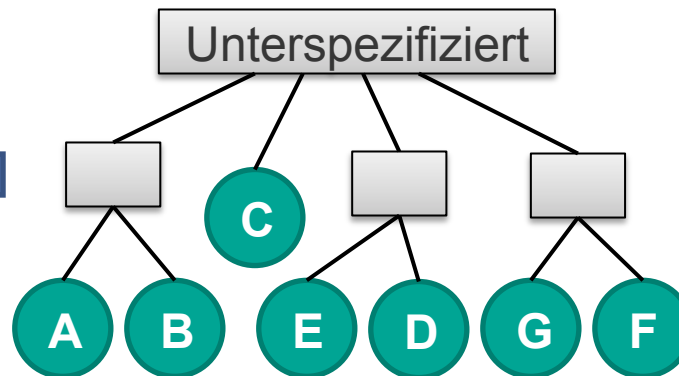
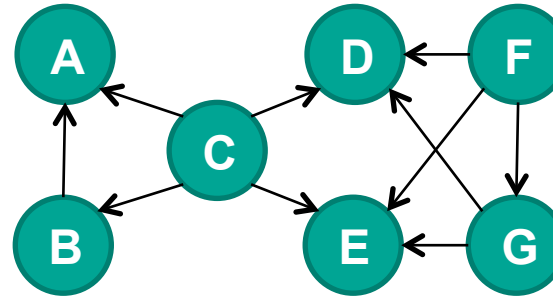
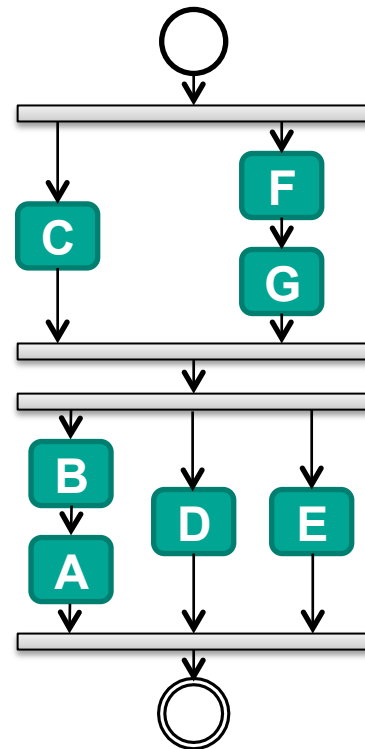
- System zur automatischen Verifikation von Abläufen durch Modell-Prüfung
- Automatische Erzeugung von Abhängigkeiten aus einer Datenbank mit Domänenwissen
- Optimierung der Laufzeit durch Reduktion der Abläufe auf relevante Regionen
- Effizienzsteigerung der Verifikation von
Ø 94,39%
- Prototyp im Einsatz

➡ Durch das System konnte die Qualität der Abläufe entscheidend verbessert werden.

Erzeugung von Inbetriebnahmeabläufe

Abhängigkeiten

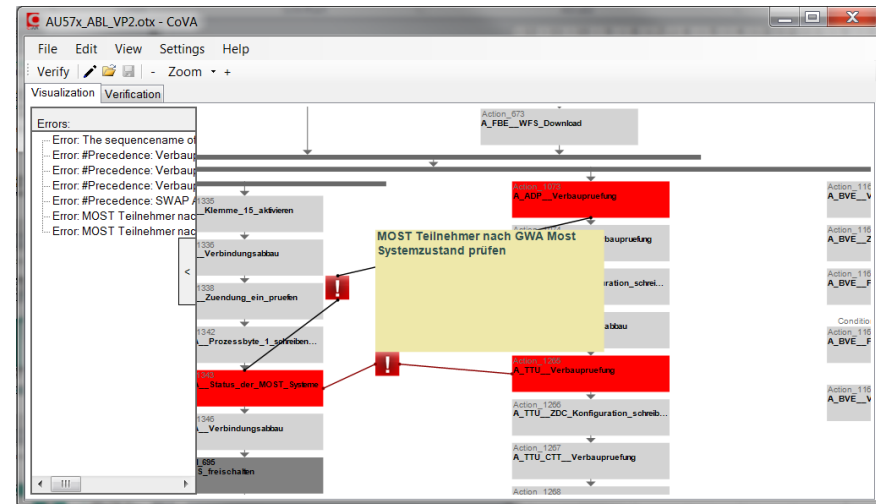
- X führt zu Y
- Z vor X
- ...



- Aus der Beschreibung der Abhängigkeiten wird ein Abhängigkeitsgraph erzeugt
- Dieser Graph wird bezüglich des Kontext reduziert
- Aus dem Graph werden die Unterspezifizierten Regionen isoliert
- Eine probabilistische Optimierung wird für die unterspezifizierten Regionen angewandt
- Ein Ablauf wird automatisch erzeugt

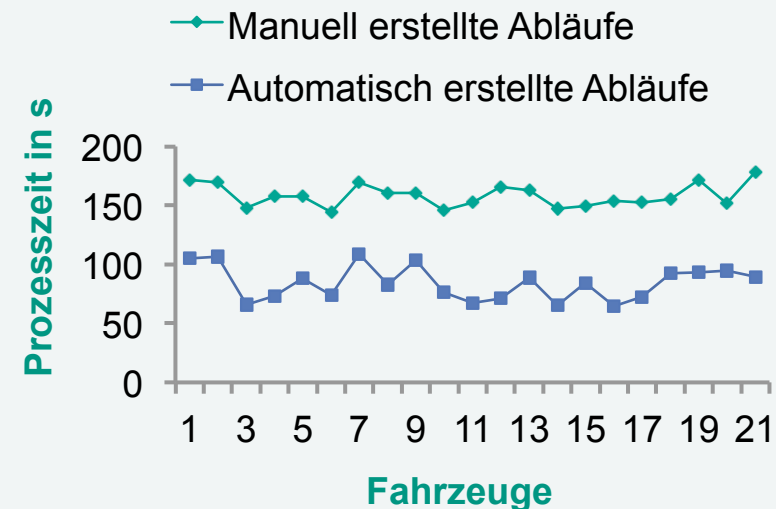
Verifikation

- Prototyp des Frameworks wird im Werk Neckarsulm bei der Entwicklung von Abläufen verwendet
- Expertenbefragung ergab gute Werte bezüglich der Benutzerfreundlichkeit und der Nützlichkeit des Tools
- Eine Verwendung in anderen Werken/Standorten ist in Planung



Generierung

- Erste Experimente zeigen ein großes Potential
- Verglichen mit manuell erzeugten Inbetriebnahmeabläufe konnte eine Laufzeitverbesserung von durchschnittlich 47% erreicht werden



- Motivation und Aufgabenstellung
- Vorarbeiten
- **Detaillierung der Aufgabenstellung**
- Organisatorisches
 - Teams
 - Zeitplan und Termine
 - Tools
- Phasen
- Phase 1: Pflichtenheft erstellen
- Was steht an?

Detailliertere Aufgabenstellung

- Das System soll intuitiv bedienbar sein und eine graphische Oberfläche haben
- Benutzerzentriert, d. h. benutzeradäquate Darstellung und komfortable Benutzerunterstützung
- Benutzer sind vorrangig für dieses Projekt Studierende, Nutzung durch Dozenten ist für eine Erweiterung des Systems zu beachten (modulare Systemarchitektur!)
- Die Nutzung von vorhandenen Tools zur Überprüfung (Verifikation) und zur Erzeugung (Generierung) von Prozessen soll vorgesehen werden. Diese Tools sind Voraussetzung und sollen nicht im Projekt entwickelt werden.
- Studienpläne für zukünftige Semester sollen vorgeschlagen oder überprüft werden
- Der bisherige Studienverlauf soll berücksichtigt werden

Detailliertere Aufgabenstellung

- Studienplan entspricht einem Ablauf, d.h. Modellierung als Workflow:
 - Aktivitäten sind Module und verschiedene Arten von Lehrveranstaltungen
 - Graphische Darstellung der Studienpläne als Workflow/ Graph
- Anordnung der Module in Semester
- Pflichtveranstaltungen und Wahlveranstaltungen können unterschieden werden
- Es gibt Vertiefungsfächer (denen Module zugeordnet sind)

Detailliertere Aufgabenstellung

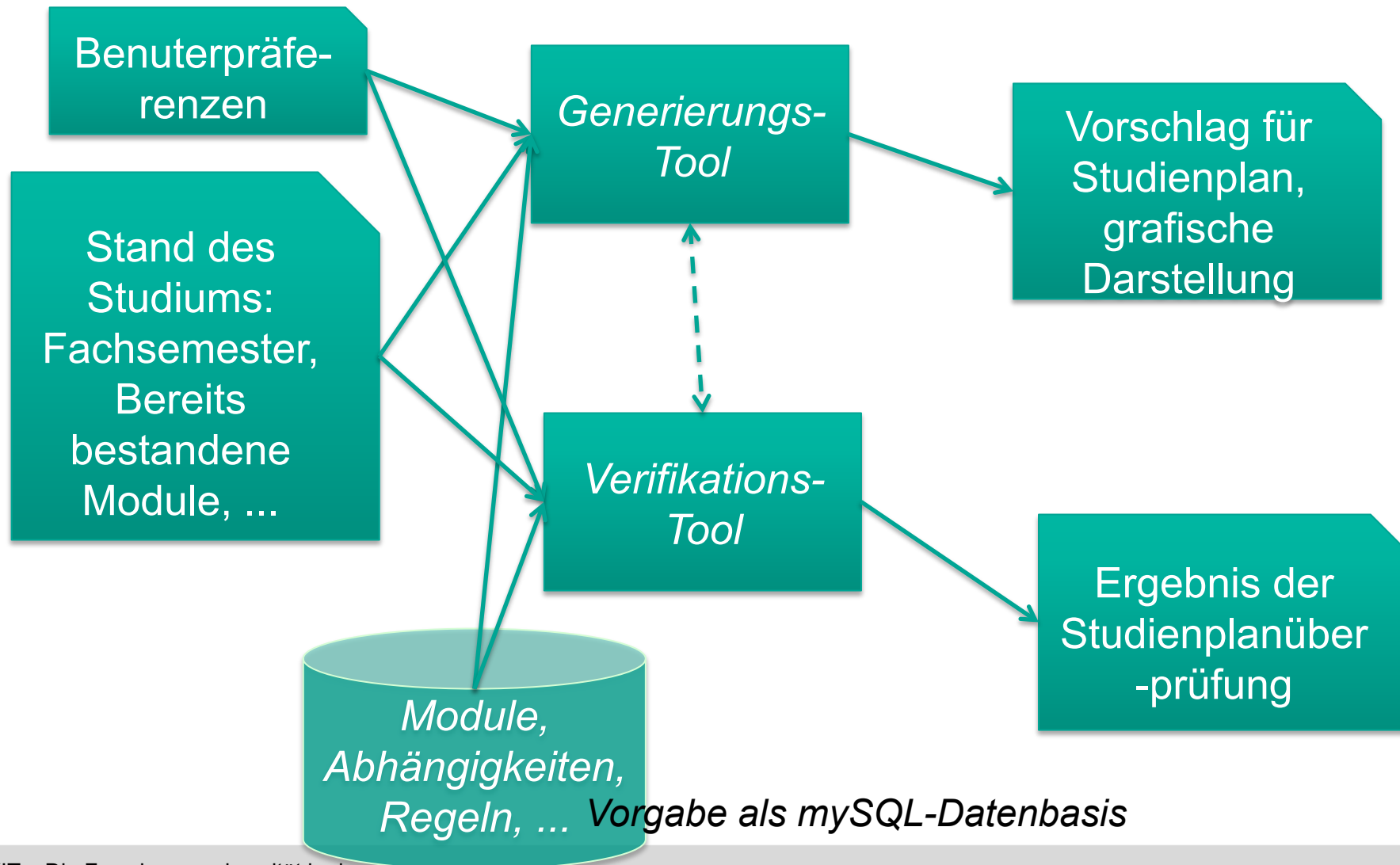
- Module werden mit ihrem Namen, ECTS-Punkte, Angebot im Winter- oder Sommersemester, Art der Veranstaltung (Vorlesung, Übung, Praktikum, Seminar, ...) modelliert.
- Abhängigkeiten zwischen Modulen (z.B. ist-Voraussetzung-für, gehört-zu, überschneiden-sich) müssen beachtet werden
- Benutzerpräferenzen, d. h. Vorgaben und Wünsche des Studierenden sollen berücksichtigt werden:
 - Vorgabe von gewünschten Vertiefungsgebieten
 - Anzahl noch benötigter Semester
 - Vorgabe von konkreten, gewünschten Modulen
 - Weitere gewünschte Eigenschaften (nicht zuviele ,nicht erforderliche‘ ECTS-Punkte, ...)

Detailliertere Aufgabenstellung

Eingaben

Black Boxes

Ergebnisse



- Motivation und Aufgabenstellung
- Vorarbeiten
- Detaillierung der Aufgabenstellung
- **Organisatorisches**
 - Teams
 - Zeitplan und Termine
 - Tools
- Phasen
- Phase 1: Pflichtenheft erstellen
- Was steht an?

Team 1

Chatti, Nada

Jungkind, Daniel

Rheinheimer, Ulrike

Teuber, Samuel

Kuchelmeister, Hannes

Uhl, Tim

Team 2

Westfechtel, Janek
Naseband, Clemens
Huss, Jacques
Bejhad, Ali Sina
Berger, Robin

Zeitplan und Termine

- Zeitplan: Anfang November 2016 – März 2017;
17 Wochen
- wöchentliche Gruppentreffen:
 - Gebäude 50.34, Raum 301
 - 45 Minuten pro Team
 - Team 1: Mittwochs 15:45 – 16:30 Uhr,
 - Team 2: Mittwochs 16:30 – 17:15 Uhr
- Abgabe der Ergebnisse der Phasen:
jeweils Dienstags bis 13:00 Uhr.

Zeitplan und Termine

- **Pflichtenheft:** 3 Wochen KW45 – 47 09.11. – 29.11.
Kolloquium: Mi, 30.11.
- **Entwurf:** 4 Wochen, KW48 – 51 30.11. – 10.01.
Kolloquium Mi, 11.01.
- Vorlesungsfrei: 2 Wochen KW52 – 1
- **Implementierung:** 4 Wochen, KW2 – 5 11.01. – 07.02.
Kolloquium Mi, 08.02.
- Klausurpause (nach Absprache) 2 Wochen KW6 – 7
- **Qualitätssicherung:** 3 Wochen KW8 – 10, 22.02. – 14.03.
Kolloquium Mi, 15.03.
- **Abschlusspräsentation:** Mitte/Ende März (wird noch festgelegt)

- Allgemeine Tipps & Tricks zum Downloaden auf:
<http://pp.ipd.kit.edu/lehre/WS201617/pse/>
- Latex für Schreiben von Dokumenten
- Versionsverwaltung, unser Vorschlag SVN:
 1. Registrieren mit ATIS-Account (einmal Login) unter
<https://svnserver.informatik.kit.edu/i40/login/>
 2. E-Mail an muelle@kit.edu
 3. Zuweisung eines SVN-Projekts (E-Mail)
- Toolunterstützung, z. B. Eclipse, JUnit, JCov, ...

- Motivation und Aufgabenstellung
- Vorarbeiten
- Detaillierung der Aufgabenstellung
- Organisatorisches
 - Teams
 - Zeitplan und Termine
 - Tools
- **Phasen**
- Phase 1: Pflichtenheft erstellen
- Was steht an?

Phasen des Projekts

- Objektorientiertes Phasenmodell ist verbindlich!
- Durchgehend Objektorientierung
- Zu den einzelnen Phasen: siehe auch [tipps.pdf](#) auf PSE-Website (Snelting)

1. Pflichtenheft
2. Entwurf
3. Implementierung
4. Qualitätssicherung
5. Abschlusspräsentation

- Motivation und Aufgabenstellung
- Vorarbeiten
- Detaillierung der Aufgabenstellung
- Organisatorisches
 - Teams
 - Zeitplan und Termine
 - Tools
- Phasen
- **Phase 1: Pflichtenheft erstellen**
- Was steht an?

Phase 1: Pflichtenheft

- **Phasenziel**
 - detaillierte Festlegung der Leistungsmerkmale eines Systems
- Grundprinzipien
 - Präzision
 - Vollständigkeit
 - Konsistenz
- **Vorgehen**
 - Systemmodell (grobe Übersicht), Systemumgebung (Hard/Software)
 - vollständige funktionale Anforderungen
 - GUI-Entwürfe (manuell oder programmiert)
 - ausführliche Testfallszenarien
- **verlangt wird**
 - Abgabe des Pflichtenheftes nach 3 Wochen;
 - Erläuterung im ersten Kolloquium

Phase 1: Pflichtenheft

- Umfang: ca. 20 Seiten
- Einleitung: Vollständige Beschreibung der Aufgabenstellung.
- Zielbestimmungen: Essentiell. Beschreibt die Funktionalität des Systems.
- Musskriterien: Mindestanforderungen, gehen aus Aufgabenstellung hervor.
- Wunschkriterien: Von den Gruppen selbst definierte, zusätzliche Funktionalität.
- Abgrenzungskriterien: Selbsterklärend.
- Produkteinsatz: Beschreibt Einsatzgebiete, Zielgruppe und Betriebsbedingungen.
- Produktumgebung: Notwendige Hard- und Software.
- Produktfunktionen: Essentiell. Detailliertere Beschreibung der Funktionalität, wiederum gegliedert in Grundfunktionen und optionale Funktionen.
- Produktdaten: Anfallende Daten außerhalb des Quellcodes.
- Systemmodell: Essentiell. Grobes Architekturdiagramm plus Beschreibung.

Phase 1: Pflichtenheft

- Produktleistungen: Anforderungen an Laufzeitverhalten oder Speicherplatz. Wichtig z. B. bei Echtzeitsystemen, eingebetteten Systemen.
- Bedienoberfläche: Essentiell. Screenshots der angedachten Bedienoberfläche (z. B. Prototyp oder frei gezeichnet) und Erläuterung der Menüstruktur.
- Qualitätszielbestimmungen: Anforderungen an Stabilität des Systems.
- Testfälle und Testszenarien: Essentiell. Testfälle für die einzelnen Produktfunktionen, die alle abgedeckt sein müssen.
- Testszenarien für typische Anwendungsszenarien.
- Entwicklungsumgebung: Zur Entwicklung verwendete Hard- und Software.
- In der Pflichtenheft-Phase sollen sich die Gruppen in die Tools einarbeiten und sich hier vorläufig festlegen.
- Glossar

- Motivation und Aufgabenstellung
- Vorarbeiten
- Detaillierung der Aufgabenstellung
- Organisatorisches
 - Teams
 - Zeitplan und Termine
 - Tools
- Phasen
- Phase 1: Pflichtenheft erstellen
- **Was steht an?**

Was steht an?

- Teams organisieren
 - Treffen absprechen
 - Phasenverantwortliche festlegen
 - ...
- Phase 1 vorbereiten

Nächstes Treffen in einer Woche:

- 16.11.2016 im SR 301
 - Team 1: 15:45 Uhr
 - Team 2: 16:30 Uhr

Viel Erfolg!