

Praxis der Softwareentwicklung – WS 2016/17

Prof. Dr. Gregor Snelting

LEHRSTUHL PROGRAMMIERPARADIGMEN



Fakultät für **Informatik**

- Ziel: Entwicklung eines mittelgroßen Systems im Team mit objektorientierter Softwaretechnik

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

- Ziel: Entwicklung eines mittelgroßen Systems im Team mit objektorientierter Softwaretechnik
- Zielsystem: max 10kLOC
objektorientierter Entwurf (UML), Implementierung (Java/C++/C#), Qualitätssicherung (z. B. Jcov, Junit)
- Teilnehmer: 3. oder 4. Sem BA Informatik
Voraussetzung: Grundbegriffe der Informatik, Programmieren, Softwaretechnik I, Klausur Lineare Algebra 1
- **Empfehlung:** PSE erst, wenn alle Module aus 1./2. Semester bestanden sind
PSE wird jedes Semester angeboten

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

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

- Neue Prüfungsordnung (SPO 2015) 9 LP für PSE
Arbeitsumfang bleibt aber gleich
Im WS 2016/17 betrifft das alle Studenten aus dem 3. Fachsemester

- Pflichtveranstaltung im Rahmen der Soft Skills (2 LP) kann nur zusammen mit PSE belegt werden
- soll PSE auf 8/9 LP bringen; explizite Lernziele Teamfähigkeit, Sprach-/Kommunikationskompetenz, Projektplanung/-management
- Note: nach Möglichkeit selbe wie „Kern-PSE“, nach Möglichkeit einheitlich für Team

- Zeitplan: Anfang November 2016 – März 2017; 17 Wochen
Praktikumsbetrieb nach Absprache
vorlesungsfreie Zeit muss zur Entzerrung genutzt werden, da sonst leicht Überlastung möglich
- 24 verschiedene Aufgabenstellungen von 12 Lehrstühlen
- objektorientiertes Phasenmodell verbindlich (vgl. Modulhandbuch)
- max. 37 Teams à 5-6 Studenten
Wünsche zu Teamzusammensetzung / Aufgabe werden nach Möglichkeit berücksichtigt

- Zeitplan: Anfang November 2016 – März 2017; 17 Wochen
Praktikumsbetrieb nach Absprache
vorlesungsfreie Zeit muss zur Entzerrung genutzt werden, da sonst leicht Überlastung möglich
- 24 verschiedene Aufgabenstellungen von 12 Lehrstühlen
- objektorientiertes Phasenmodell verbindlich (vgl. Modulhandbuch)
- max. 37 Teams à 5-6 Studenten
Wünsche zu Teamzusammensetzung / Aufgabe werden nach Möglichkeit berücksichtigt
- Eventuell stehen nicht genügend Teilnehmerplätze zur Verfügung ⇒
Warteliste für nächstes Semester
- Bitte Webseite beachten:
<http://pp.ipd.kit.edu/lehre/WS201617/pse/>

Es gibt folgendes Zulassungsverfahren:

1. Teilnehmer melden sich in WebInScribe an.
2. Gegenprüfung der formalen Voraussetzungen am IPD Snelting soweit möglich.
3. Rückmeldung an Betreuer, falls Gegenprüfung *nicht* erfolgreich.
4. Betreuer prüfen Notenspiegel der fraglichen Fälle.
5. Umverteilung der Teams in 2. PSE-Woche, falls Teams auseinander fallen.

- moderne Softwaretechnik ist wichtig für alle BA-Absolventen!
- vollständige Entwicklung eines größeren Systems
- Phasenmodell:
 1. Pflichtenheft
 2. Entwurf
 3. Implementierung
 4. Qualitätssicherung
 5. Abschlusspräsentation
- Phasenverantwortliche
- Teamarbeit (Teams à 5-6 Teilnehmer)
- durchgehend Objektorientierung
- Toolunterstützung, z. B. Eclipse, Rational Architect, JUnit, Jcov, ...

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

objektorientiert (UML)

Phasenziel

- Festlegung der Klassenstruktur
- Schnittstellendefinition der Klassen
- Beziehungen zw. Klassen (Vererbung, Assoziationen)
- Klassendiagramm, ausgewählte Sequenzdiagramme, evtl. Zustandsdiagramm
- Einsatz von Design Patterns, MVC

objektorientiert (UML)

Phasenziel

- Festlegung der Klassenstruktur
- Schnittstellendefinition der Klassen
- Beziehungen zw. Klassen (Vererbung, Assoziationen)
- Klassendiagramm, ausgewählte Sequenzdiagramme, evtl. Zustandsdiagramm
- Einsatz von Design Patterns, MVC

Grundprinzipien

- Geheimnisprinzip
- schwache Kopplung
- hohe Kohäsion
- Lokalitätsprinzip
- Wiederverwendbarkeit von Klassen/Subsystemen

objektorientiert (UML)

Phasenziel

- Festlegung der Klassenstruktur
- Schnittstellendefinition der Klassen
- Beziehungen zw. Klassen (Vererbung, Assoziationen)
- Klassendiagramm, ausgewählte Sequenzdiagramme, evtl. Zustandsdiagramm
- Einsatz von Design Patterns, MVC

Grundprinzipien

- Geheimnisprinzip
- schwache Kopplung
- hohe Kohäsion
- Lokalitätsprinzip
- Wiederverwendbarkeit von Klassen/Subsystemen
- OO: Vererbung/dynamische Bindung statt Fallunterscheidung

Vorgehen

- Kombination von Top-Down und Bottom-Up Design
- Identifikation von Klassen, Vererbung, Assoziationen
- Festlegung der Schnittstellen aller Klassen
- informale Beschreibung aller Klassen
- evtl. Einsatz von Entwurfsmetriken (JMetrics)

Vorgehen

- Kombination von Top-Down und Bottom-Up Design
- Identifikation von Klassen, Vererbung, Assoziationen
- Festlegung der Schnittstellen aller Klassen
- informale Beschreibung aller Klassen
- evtl. Einsatz von Entwurfsmetriken (JMetrics)

verlangt wird

- Abgabe der UML-Diagramme nebst informeller Beschreibung nach 4 Wochen; Verteidigung im zweiten Kolloquium
- *Nachweis der Evolutionsfähigkeit* (z. B. Lokalitätsprinzip)

3. Implementierung

Phasenziel: Programmierung des Systems

Grundprinzipien

- Programmierung in Java (evtl. C#, C++)
- Umsetzung der Architektur

3. Implementierung

Phasenziel: Programmierung des Systems

Grundprinzipien

- Programmierung in Java (evtl. C#, C++)
- Umsetzung der Architektur

Vorgehen

- Implementierungsplan vorher
- Implementierung der Methoden
- funktionaler Komponententest mit `JUnit`, evtl. Überdeckungstests (z. B. `JCov`); verschränkt mit Implementierung
- Realisation der Szenarien aus Pflichtenheft

verlangt wird

- Implementierungsplan; Implementierung;
- Implementierungskolloquium (2 Wochen vor Abschluss)

4. Qualitätssicherung

Phasenziel: Test des Systems

Grundprinzipien

- werkzeugunterstützte Qualitätssicherung

4. Qualitätssicherung

Phasenziel: Test des Systems

Grundprinzipien

- werkzeugunterstützte Qualitätssicherung

Vorgehen

- Integrationstest, Robustheitstest
- Prüfen der Szenarien aus Pflichtenheft

verlangt wird

- Testbericht; Systemabnahme (1 Woche vor Abschluss)

5. Abschlusspräsentation

- Abschlusspräsentation (Frühjahr 2017)
- Nach Möglichkeit mehrere Lehrstühle zusammen



image source: <https://www.flickr.com/photos/svenwerk/506579282/>

Allgemeine Hinweise und Tipps

20 Seiten Hinweise und Tipps (im Laufe der letzten Jahre entstanden) für Betreuer und Studierende als PDF gibt es auf der PSE-Webseite zum Download:

<http://pp.ipd.kit.edu/lehre/WS201617/pse/>

Ihre Beiträge dazu nehmen wir gerne hier entgegen:

<https://git.scc.kit.edu/IPDSnelting/pse-tipps/>