Karlsruher Institut für Technologie

# Praxis der Softwareentwicklung – SS 2015

**Prof. Dr. Gregor Snelting**

LEHRSTUHL PROGRAMMIERPARADIGMEN

Fakultät für **Informatik**

# Praxis der Software-Entwicklung (PSE)

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

# 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#), Qualitätssicherung (z. B. Jcov, Junit)

# Praxis der Software-Entwicklung (PSE)

Karlsruher Institut für Technologie

- 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, *ein* Modul aus der Mathematik (HM oder LA)
- Empfehlung: PSE erst, wenn alle Module aus 1./2. Semester bestanden sind
  PSE wird jedes Semester angeboten

# 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#), Qualitätssicherung (z. B. Jcov, Junit)
- Teilnehmer: 3. oder 4. Sem BA Informatik
  Voraussetzung: Grundbegriffe der Informatik, Programmieren, Softwaretechnik I, *ein* Modul aus der Mathematik (HM oder LA)
- Empfehlung: PSE erst, wenn alle Module aus 1./2. Semester bestanden sind
  PSE wird jedes Semester angeboten
- Umfang: 8 LP, $\approx$ 250 Arbeitsstunden / Teilnehmer,
  $\approx$ 2 Arbeitstage/Woche/Teilnehmer

# Teamarbeit in der Software-Entwicklung (TSE)

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

# PSE / Organisation

- Zeitplan: Ende April 2015 - August/September 2015; 17 Wochen Praktikumsbetrieb nach Absprache
  vorlesungsfreie Zeit muss zur Entzerrung genutzt werden, da sonst leicht Überlastung möglich
- 16 verschiedene Aufgabenstellungen von 12 Lehrstühlen
- objektorientiertes Phasenmodell verbindlich (vgl. Modulhandbuch)
- max. 22 Teams à 5-6 Studenten
  Wünsche zu Teamzusammensetzung / Aufgabe werden nach Möglichkeit berücksichtigt

# PSE / Organisation

- Zeitplan: Ende April 2015 - August/September 2015; 17 Wochen Praktikumsbetrieb nach Absprache
  vorlesungsfreie Zeit muss zur Entzerrung genutzt werden, da sonst leicht Überlastung möglich
- 16 verschiedene Aufgabenstellungen von 12 Lehrstühlen
- objektorientiertes Phasenmodell verbindlich (vgl. Modulhandbuch)
- max. 22 Teams à 5-6 Studenten
  Wünsche zu Teamzusammensetzung / Aufgabe werden nach Möglichkeit berücksichtigt
- Voraussichtlich stehen nicht genügend Teilnehmerplätze zur Verfügung $\Rightarrow$ Warteliste für nächstes Semester
- Bitte Webseite beachten:
  `http://pp.ipd.kit.edu/lehre/SS2015/pse/`

# Zulassungsverfahren

Aufgrund neuer formaler Voraussetzungen gibt es dieses Semester 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. VL-Woche, falls Teams auseinander fallen.

# PSE / Übersicht

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

# 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

# Entwurf & Spezifikation

*objektorientiert (UML)*

## Phasenziel

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

# Entwurf & Spezifikation

*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

# Entwurf & Spezifikation

*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

# Entwurf & Spezifikation / 2

## 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)

# Entwurf & Spezifikation / 2

## 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)

# Implementierung & Validierung

## Phasenziel

Programmierung und Test des Systems

## Grundprinzipien

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

# Implementierung & Validierung

## Phasenziel

Programmierung und Test des Systems

## Grundprinzipien

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

## Vorgehen

- Implementierung der Methoden
- funktionaler Komponententest mit `Junit`, evtl. Überdeckungstests (z. B. JCov); verschränkt mit Implementierung
- Integrationstest, Robustheitstest
- Realisation der Szenarien aus Pflichtenheft

# Implementierung & Validierung / 2

verlangt wird

- Implementierungsplan; Implementierung;
- Implementierungskolloquium (2 Wochen vor Abschluss)
- Testbericht; Systemabnahme (1 Woche vor Abschluss)
- Abschlusspräsentation (Sommer 2015) nach Möglichkeit mehrere Lehrstühle zusammen