Karlsruher Institut für Technologie

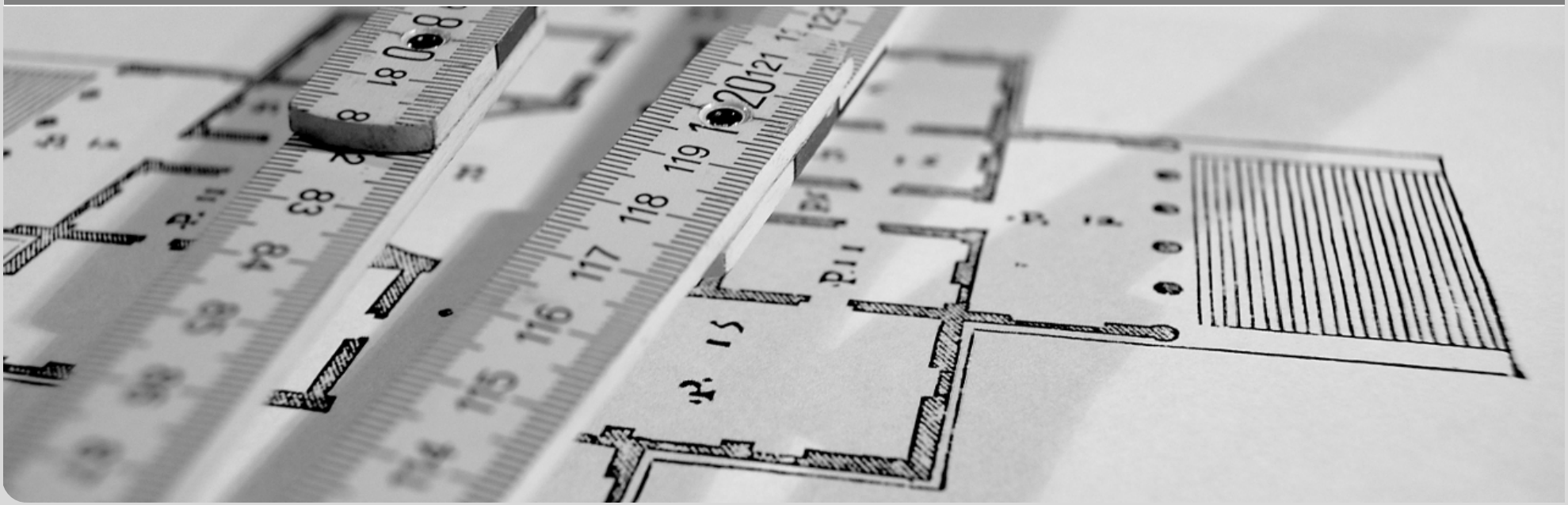
SDQ Software Design and Quality

SWT 1 - Kennenlerntutorium

Tutorium Nr. 17

Kay Schmitteckert | 24.04.2015

INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



1. Organisatorisches
   - Vorstellung
   - Übungsbetrieb & Klausur
2. Tools
   - Git
   - Maven
   - Checkstyle
3. Übungsblatt
   - JUnit Modultests
   - Codebeispiel

Über mich

- Kay Schmitteckert
- Informatik
- 4. Semester
- E-Mail: uheat@student.kit.edu

Ihr

- Name
- Studiengang und Semester
- Programmiersprachen
- Erfahrungen mit Softwareentwicklung
- Erwartungen und Wünsche ans Tutorium

Verhalten im Tutorium

- lockere Atmosphäre
- Rücksichtnahme
  - Höflichkeit
  - Pünktlichkeit
- aktive Teilnahme
  - **Mitdenken**
  - **Fragen stellen**
- Folien im Ilias

Übungsbetrieb

- Tutorium alle 14 Tage **(nächstes Tut am 08.05.2015)**
- 6 Übungsblätter
- Abgabe in der Regel Mittwoch 12 Uhr (auch alle 14 Tage)
- insgesamt 150 Punkte
- Übungsschein bestanden mit 50% der Punkte
  - Anmeldung im *Studienportal* notwendig!
- Abgabe:
  - Theorieteil: Holzkiste 3. Stock vor Raum 368 im Infobau
  - Praxisteil: Lösungseinzugszentrale: *http://lez.ipd.kit.edu*

Klausur

- i.d.R 60 Minute Klausur
  - Hauptklausur: 10.08.2015, 14:00 Uhr
  - Nachklausur: 06.10.2015, 11:00 Uhr
- Anmeldung auch übers *Studienportal*

Git

Wichtige Befehle

`git init`	initialisiert ein Git-Depot im aktuellen Ordner
`git add` *Dateiname*	fügt eine Datei zur Versionsverwaltung hinzu
`git cp` *Dateiname*	kopiert eine Datei im Verzeichnis
`git log`	gibt alle Änderungen in chronologischer Reihenfolge aus
`git rm` *Dateiname*	löscht eine Datei im Verzeichnis
`git commit -m` *Beschreibung*	bucht Änderungen in das Depot ein

Git

Wichtig!

- kurze aber aussagekräftige Kommentare
- **So kurz wie möglich, so lang wie nötig**
- Einbuchen in sinnvollen Teilschritten - *"neue Klasse hinzugefügt"* kein guter Eintrag!
- Entscheidet euch für eine Sprache und behaltet sie bei!

Maven

Wichtige Befehle

`mvn compile`	kompiliert die verwendeten Quelltexte zu .class-Dateien
`mvn test`	kompiliert die Tests und führt sie aus
`mvn package`	erstellt ein ausführbares .jar-Paket des Projekts, das mit `java -jar projekt.jar` gestartet werden kann
`mvn clean`	löscht den Ordner mit den Kompilaten

Checkstyle

Kennt ihr aus Programmieren

JUnit Einführung

- Testframework für Java
- bereits in Eclipse integriert
- Anlegen einer Testklasse
  - mindestens ein eigenes Package für Tests
    src/test/java/"PackageName"
  - mindestens eine Testklasse pro Komponente

Kennzeichnung der Methoden durch Annotationen

`@Before`	wird vor jedem Testfall ausgeführt
`@After`	wird nach jedem Testfall ausgeführt
`@Test`	kennzeichnet einen Testfall
`@BeforeClass`	Ausführung vor Instanziierung der Testklasse
`@AfterClass`	Ausführung nach allen anderen Methoden
`@Ignore`	noch kein Inhalt, Test wird ignoriert

Codebeispiel