Organization
The Lecture’s Goal
The goal is to enable you to systematically carry out small(er) commercial or open-source software projects.
Basic Goals

• To get a brief overview of “all” areas of software engineering
• To understand agile software development processes
• To be able to perform object-oriented analysis and design
• To be able to read and create basic UML diagrams
• To perform basic Software Quality Assurance
• To get first hands-on experience and to learn to use basic software development tools
• To be able to use basic design patterns
Basic programming skills are required.

- Basic knowledge of object-oriented programming concepts is necessary i.e., you should readily understand the following terms:
  - class, interface
  - object
  - inheritance
  - polymorphism
- Working knowledge of the Java programming language
- We are going to use Java 8
The Lecture’s Structure
What is Software Engineering?

Historical Background

Requirements Engineering

Software Engineering Processes

Waterfall Model

Agile Methods

Software Project Management

Risk Management

Properties of Software Projects

Testing

Unit Testing

“Code Coverage”

Software Engineering Tools

Revision Control Systems

GIT

Software Engineering

Dynamic Behavior

Static Structure

Class Diagrams

Sequence Diagrams

Communication Diagrams

Object Diagrams

Domain Modeling

IDEs

Revision Control Systems

Unit Testing

“Code Coverage”

Software Engineering

Agile Methods

Waterfall Model

Requirements Engineering
Software Engineering

What is Software Engineering?

Software Project Management

Testing

Software Engineering Tools

Object-Oriented Analysis and Design

UML Modeling

Historical Background

Properties of Software Projects

Requirements Engineering

Software Engineering Processes

Risk Management

Agile Methods

Waterfall Model

Dynamic Behavior

Static Structure

Class Diagrams

Sequence Diagrams

Communication Diagrams

Object Diagrams

Domain Modeling

Design Patterns

Idioms

General Design Goals

Low Coupling

High Cohesion

Singleton

Template Method

...
The Team
The Team

Dr. Michael Eichberg
Leonid Glanz
Sven Amann

Contact
WWW
http://www.stg.tu-darmstadt.de/teaching/teaching_iverview.en.jsp

Forum
http://www.d120.de/forum/
→kanonische Einführungsveranstaltungen
→Einführung in Software Engineering
Organization
Lecture

• Thursdays 13:30-15:00 in S2 02 | C205 and C110

• The slides are in English
  (Key terms will be translated into German.)

• The slides will generally be available after the lecture
  (I will try hard to make a preliminary version available the day before the lecture.)

• The slides can be found at
  http://stg-tud.github.io/eise/
Exercises

• Every week, we will have an exercise, starting today
• Exercises are expected to be solved in teams of 3 students
• The content of the exercise is relevant for the exam
• Sign-up as a team; if you don’t have a team, we will assign you to a team
• The content of the exercise is relevant for the exam
Exercises - Bonus

• You can get a bonus by successfully completing the exercise.

• The maximum bonus is equivalent to a full grade (e.g., 2,0 => 1,0).

• You need to achieve at least 25% of all exercise points to get some bonus.

• The bonus cannot be used to pass the exam.
Written Exam

• The exam will be on February, 25th 2016 - 14:30 - 16:00
  (The rooms will be announced in the forum. The exam will take 90min.)

• You need to register for the exam in TUCaN
  (There are no further prerequisites; “everyone” can attend the exam.)

• The exam will be an open book exam

• The very best students are expected to be able to solve the entire exam
Related Bibliography
Essential Bibliography

• Design Patterns - Elements of Reusable Object-Oriented Software; Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides; Addison-Wesley, 1995

• Applying UML and Patterns - An Introduction to Object-oriented Analysis and Design; Craig Larman; Prentice Hall
A Recommended / Very Useful Podcast

Software Engineering Radio
The Podcast for Professional Software Developers

Recent Episodes

Episode 211: Continuous Delivery on Windows with Rachel Laycock and Max Lincoln
Filed in Episodes by SE-Radio on September 30, 2014 • 0 Comments

Johannes talks with Rachel Laycock and Max Lincoln from ThoughtWorks about continuous delivery on Windows. The outline includes: introduction to continuous delivery; continuous integration; DevOps and ChatOps; decisions to be taken when implementing continuous delivery on windows; build tools on windows; packaging and deploy on windows; infrastructure automation and infrastructure as code with chef, puppet […]

Continue Reading »

Episode 210: Stefan Tilkov on Architecture and
External Talks / Events

• **MSG**
  “Requirements Engineering”
  December, 2015

• **Capgemini**
  Excursion / One-day Workshop
  December, 2015
Software Engineering in der industriellen Praxis
Software Engineering - Projektmanagement
Software Engineering Design and Construction
Konzepte der Programmiersprachen
...