MTAT.03.094
Software Engineering

Lecture 01.2:
Introduction to Software Engineering

Dietmar Pfahl
email: dietmar.pfahl@ut.ee

Fall 2016
Schedule of Lectures (Tentative)

**Week 01: Introduction to SE**
- Week 02: Requirements Engineering I
- Week 03: Requirements Engineering II
- Week 04: Analysis

**Week 05: Development Infrastructure I**
- Week 06: Development Infrastructure II
- Week 07: Architecture and Design
- Week 08: Refactoring

**Week 09: Verification and Validation I**
- Week 10: Verification and Validation II

**Week 11: no lecture**
- Week 12: Software Quality Management
- Week 13: Agile/Lean Methods
- Week 14: Software Craftsmanship
- Week 15: Measurement / Course wrap-up, review and exam preparation

**Week 16: no lecture**
Software Engineering

What?

Why?
Software Development – Three Ps

- Software Development

Project or Iteration
Software Development – Three Ps

- Software Development
- Project or Iteration
- Products
- People
- Processes
Software Development – Three Ps

- Software Development

Products

Project or Iteration

People

Processes
Products in Software Development

**Code:**
- Production code:
  - Source code
  - Object code
- Non-production code:
  - Test code

**Non-Code:**
- Requirements
- Specifications
- Architecture/Design docs
- Issue reports
- User manuals
- Plans of all kinds
- ...

**Models**

**Properties of Software:**
- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

**Types of Software:**
- Embedded/real-time
- Information System
- Web application
- System software
- ...

**Elements of Software:**
- Code
- Properties
- Models

**Elements of Software:**
- Code
- Properties
- Models
Properties of Software

The software should deliver the required functionality and performance to the user and should be maintainable, dependable and acceptable.

- **Maintainability**
  Software must evolve to meet changing needs;

- **Dependability (Reliability)**
  Software must be trustworthy;

- **Efficiency**
  Software should not make wasteful use of system resources;

- **Usability**
  Software must be accepted by the users for which it was designed. This means it must be understandable, usable and compatible with other systems.
SW Product Modeling

UML = Unified Modeling Language

Online information: http://www.uml.org
Software Development – Three Ps

- Software Development
- Project or Iteration
- Products
- Processes
- People
People in Software Development

Roles:
- Project Manager
- Product Manager
- Architect/Analyst
- Programmer
- Tester
- ...

Teams:
- Team building
- Geographically distributed (international/global)
- Mechanisms for collaboration/cooperation
- Motivation, Personality, Values, Culture

Skills:
- Must match roles

Training:
- Must fill skill-gaps

Education:
- Curricula (ACM/IEEE)

User models
Software Development – Three Ps

- Software Development
- Project or Iteration
- Products
- People
- Processes
Software Development Process

Coding

Deploying
Software Development Process

Find Requirements
Analysis / Designing
Coding
Testing
Deploying
Software Development Process

(Royce, 1970)
Agile Process
Agile Process

Scrum

eXtreme Programming (XP)
Scrum Elements – Process, Artifacts, Roles

http://www.scrumforteamsystem.com/processguidance/v1/Scrum/Scrum.html
13 XP Practices

**Project Cycle**
- Planning Game (Poker)
- Small Releases
- Whole Team
- Customer Tests

**Development Cycle**
- Simple Design
- Pair Programming
- TDD (Unit Test)
- Refactoring

**Supporting Practices**
- Coding Standard
- Sustainable Pace (40-hour week)
- Metaphor (Common Understanding)
- Continuous Integration
- Collective Ownership
Comparison of Basic Process Types

Time

Functionality

- Requirements
- Design
- Implementation
- Testing

Waterfall

Incremental, e.g. RUP

Agile - XP

RUP = Rational Unified Process
XP = Extreme Programming
Processes in SW Development

Process (Model) Elements:
- Activity
- Input/Output Product(s)
- Roles
- Methods/Techniques/Tools

Process Taxonomy:
- Non-engineering processes
  - Business processes
  - Social processes
- Engineering processes
  - Product-engineering proc.
    - Technical prod.-eng. proc.
    - Managerial prod.-eng. proc.
  - Process-engineering proc.

Process Modeling:
- Descriptive PMs
- Prescriptive PMs
  - Standards
  - Families

Process Types:
- Heavy-weight (rich)
- Light-weight
  - Lean
  - Agile
  - Kanban
Software Engineering

Consistent application of engineering principles and methods to the development of software (intensive) systems

Engineering:
Application of systematic (i.e., predictable, repeatable, scalable) procedures
- with well-defined goals (e.g., quality, functionality/scope, cost, time)
- with well-defined/structured products, processes, and organization
Adherence to existing body of knowledge
Observation of constraints (standards, time/cost/quality requirements, etc.)
Development and use of models

Magic Triangle

Quality / Scope

Product

Effort (Cost)

Time

Trade-offs!
Magic Triangle of SE

- Quality / Scope
- Effort (Cost)
- Time

Trade-offs!

- 6 Quality Characteristics (ISO 9126 / ISO 25000):
  - Functionality
  - Reliability
  - Usability
  - Efficiency
  - Maintainability
  - Portability

- Effort/Cost and Time:
  - Development
  - Maintenance
Software Engineering

A bridge from customer/user needs to software product

Customer, User Needs

Software Engineering

Developer (SW Engineer)

Software Product
Next Lecture

• Date/Time:
  • Friday, 11-Sep, 10:15-12:00

• Topic:
  • Requirements Engineering I ➔ 1st Homework!

• For you to do:
  • Have a look at the course wiki
  • Make sure you know to which lab group you have been enrolled + start forming project teams
  • MOST IMPORTANTLY: Go to the labs next week!