Course introduction
Software Engineering Theory

Kristian Sandahl
Department of Computer and Information Science
2014-09-02
Who is Kristian?
Who is Kristian?

• Professor in Software Engineering since 2001
• I love studying (large) software development organizations and evaluate practices of Software Engineering
• I love teaching project courses, but I’m not a faithful constructivist
• My mission: To bring some critical thinking and common sense to the world
• Background: Civ ing (D) -83; PhD -92
  Epitec: spin-off company 86-87
  Ericsson research: 95-01
Who are you?

- SVP
- I, li
- Master
- Erasmus
- Other
- C/D/U

Programs:
- TDDC88
- 725G64
- TDDC93
- Laboratory work
- Project
- TDDD69
- ISE
Why should I take this course?

Well, I have to, it's compulsory...

It is not a hard mathematical course

This is a rather pragmatic course, concerning real problem in the industry

Are you going to work in the software industry?
If yes, avoid Larger disasters


• Frequent failures:
  • Space shuttle
  • Nike
  • Denver airport
  • LA airport


• A commercial blog:
  http://www.wayferry.com/failures
Software Engineering

- Application of systematic, disciplined, quantifiable approach to software development, operation and maintenance of software. (IEEE-Std.)
Course Goals

Give sound theoretical foundation of Software Engineering (TDDC88 and TDDC93)

The intended learning outcomes are that the student at the end of the course can:

• explain and exemplify basic concepts in the area of large-scale software engineering.

• explain how to specify, model, implement and test a software system.

• explain how to execute a software development project.
Course Goals

To gain basic experience of being part of a larger software engineering project (TDDC88 only)

The intended learning outcomes are that the student at the end of the course can:

- specify, model, implement, and test a smaller software system
- define, plan and execute a development project in a group of 25-30 students, where several groups cooperate to produce a common product.
- elicit, analyze and document experience from the own development project
- use basic functions from a selection of tools used in software industry
Ambition level

• You will know enough to communicate easily with professional software engineers
• You will have the basic knowledge to start generating your own experience already in student projects
• You will have a curious, but critical, attitude towards existing and new methods
"There is no silver bullet"

- A good Software Engineer knows several methods, tools, and techniques:
  - How does it work?
  - What are the benefits?
  - What are the drawbacks?
  - Is it good for my situation?

"No Silver Bullet — Essence and Accidents of Software Engineering"
Fred Brooks, 1986
Studying SE theory

- Reading about concepts and methods derived from experience
- Almost everything have their pros and cons
- Quite little is based on mathematical theory
- Requires disciplined plan for reading different sources
Scope of a software life-cycle model

Idea → Software Product → Usage → Operation & maintenance → Replacement
Model of a life-cycle

Abstraction level

Carol
the customer

Diana
the developer

Time

LiU EXPANDING REALITY
Model of a life-cycle

Knowledge areas – lecture week overview

- Week 36 – Requirements
- Week 37 – Planning and Processes
- Week 38 – Design and Architecture
- Week 39/40 – Testing and SCM
- Week 41 – Software Quality
KA #1: Requirements

Requirements

Lecture - Requirements

- **Elicitation**: Collect user requirement
- **Analyze**: Understand
- **Specify**: Document
- **Validate**: Check that it matches user requirements

- **Ulla**: the user
  - No user input until too late
- **Sam**: the seller
  - No written specification
Supporting Processes

Requirements
KA #2: Planning and Processes

Lecture - Project Management

- Time plan
  - Resources
  - Calendar-time
  - Milestones
  - Time-budget
  - Activities

Risk Management
- Identify
- Analyze
- Plan
  - probability - consequence
- Resource management - Harry
- Time-buffer, toll-gates
- Internal milestones - keep track

Harry the hacker
- Minimize
- Transfer
- Accept
KA #2: Planning and Processes

Lecture - Software Life Cycles and Processes
Agile Methodologies

- V-model
- Waterfall model
- Iterative models
  - Processes e.g. Scrum, XP, RUP
KA #3: Design and Architecture

Lecture - System Design and Architecture

Decompose into sub-systems or modules
- Well-defined interfaces
- High level of abstraction

Architecture styles, e.g.
- Client-server
- Layered Models
- Pipes and Filters
- SOA
KA #3: Design and Architecture

Requirements

System Design
(Architecture, High-level Design)

Module Design
(Program Design, Detailed Design)

Lecture - Module Design and UML

Unified Modeling Language (UML)

Use a standardized way to model system graphically

Design Patterns

Reuse design solution that has worked before

Conny
the consultant

His "own" notations
KA #4: Testing and SCM

Validate Requirements, Verify Specification

Acceptance Test
(Release testing)

Lecture - Testing

Integration testing
- Dependencies between modules

Approaches
- Top-down (need stubs)
- Bottom-up (need drivers)
- Sandwich
- Big-bang

Lack of unit, integration and system testing
KA#4: Testing and SCM

Verify System Design

Validate Requirements, Verify Specification

Acceptance Test
(Release testing)

System Testing
(Integration testing of modules)

Verify Module Design

Module Design
(Program Design, Detailed Design)

System Design
(Architecture, High-level Design)

Requirements

Implementation of Units (classes, procedures, functions)

Unit testing

Verify Implementation

Keep track of versions.
Used the wrong code-base.

Lecture - Software Configuration Management

Configuration Management (CM)

Carol the customer

Crash!
KA #5: Software Quality

**Lecture - Software Quality Management**
**Lecture - Software Reviews**
**Lecture - Software Metrics**

Different quality factors
- Quality of product
- Quality of process
- Quality in business
- Finding defects
- Learning organization
Course ending (Theory part)

- Validate Requirements, Verify Specification
- Verify System Design
- Acceptance Test (Release testing)
- System Testing (Integration testing of modules)
- Unit testing

- Requirements
- System Design (Architecture, High-level Design)
- Module Design (Program Design, Detailed Design)
- Implementation of Units (classes, procedures, functions)
- System Testing (Integration testing of modules)
- Acceptance Test (Release testing)
- Validate Requirements, Verify Specification

- Lecture – Course Summary
- Lecture – Exam rehearsal
- Guest lectures

Project Management, Software Quality Assurance (SQA), Supporting Tools, Education
What is the content of this course?

**General**
- Literature
- Personnel
- Feedback

**Theory**
- Examination
- Lectures & Exercises

**Project (only TDDC88 TDDD69 725G64)**

**Labs (only TDDC88 725G64)**

www.ida.liu.se/~TDDC88
(same for all)
Literature (recommended)

Pfleeger, S. L. and Atlee, J. M. Software Engineering Theory and Practice

or

Pfleeger, S. L. and Atlee, J. M. Software Engineering Theory and Practice

+ links to online articles and websites
Personnel

Examiner and Course Leader
Kristian Sandahl, kristian.sandahl@liu.se

Course Administrator
Åsa Kärrman, asa.karrman@liu.se

Lecture Exercise Assistant
Alexander Widerberg, alexander.widerberg@liu.se

Laboratory Exercise Assistants
Erik Hansson, erik.hansson@liu.se
Jonas Wallgren, jonas.wallgren@liu.se
Johannes Schmidt, johannes.schmidt@liu.se

Project Supervisors and Examiners
CEO for the Company Rita Kovordányi, rita.kovordanyi@liu.se
Supervisor for Requirements, Testing, and Quality
Maria Vasilevskaya, maria.vasilevskaya@liu.se
Supervisor for the Architecture, Design, and Configuration Management
Rikard Nordin, rikard.nordin@liu.se

Main customer contact Zenterio AB

Director of studies
Tommy Färnqvist, tommy.farnqvist@liu.se
Feedback

Muddy Cards, Wednesday week 38

All feedback is welcome
- Via mail kristian.sandahl@liu.se
- Via phone 0706-681957
- Via coffee. Drop an e-mail, so we can schedule a time

KURT (webbased system)

Student representative feed-back
Main improvements this year due to student feedback

- TDDC88 theory/TDDC93: New assignments
- TDDC88 theory/TDDC93: Seminars skipped
- TDDC88 theory/TDDC93: Clearer course definitions
- TDDC88 project: Course code and Gender used in lottery of groups
- TDDC88 project: Unscheduled 4 hour blocks vt2
- TDDC88 project: Student run role assignment
- TDDC88 labs: Replaced lab on JDBC with Software Metrics
Changes since last year

Main improvements this year due to research

- TDDC88 project: Peer assessment
- TDDC88 project: Run experience build up twice
**Feedback**

**General**
- Literature
- Personnel

**Theory**
- Examination
- Lectures & Exercises
- QWA

**Project (only TDDC88)**

**Labs (only TDDC88)**

---

**Hi!**

Can we really use the term non-functional? Isn't that just an abstraction of functional requirements?

Pär Pärsson, D3
cellular +46 (0)/06 68 19 57
Examination - when?

<table>
<thead>
<tr>
<th>Course part</th>
<th>Examination</th>
<th>Credits</th>
<th>Applicable to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory</td>
<td>Written Exam</td>
<td>4hp, 4 ECTS</td>
<td>TDDC88 and TDDC93</td>
</tr>
<tr>
<td>Project</td>
<td>Project Tasks</td>
<td>6hp, 6 ECTS</td>
<td>TDDC88 and TDDD69</td>
</tr>
<tr>
<td>Laboratory exercises</td>
<td>Oral Exam and Written Exercises</td>
<td>2hp, 2 ECTS</td>
<td>TDDC88</td>
</tr>
</tbody>
</table>

Written Exam (both TDDC88 and TDDC93)

- Primary Exam 14:00-18:00, October 31, 2014
- Retake Exam 14:00-18:00, January 10, 2013
Examination - content? to pass?

**Part I: Fundamentals**
- Requirements
- Planning and Processes
- Design and Architecture
- Testing and SCM
- Software Quality

10 credits per area. Max 50 credits.

**Part II: Advanced**
50 credits, distributed over 2-5 questions.
-argue, compare, and analyze different concepts and techniques.
-construct and/or design solutions to larger problem.
-explain more advanced and specific topics.

---

**General**
- Literature
- Personnel
- Feedback

**Theory**
- Examination
- Lectures & Exercises
- QWA

**Project (only TDDC88)**

**Labs (only TDDC88)**

**To pass the exam (alternatives)**
1. a) at least 4 credits in all areas in fundamentals **and**
   b) at least 50 credits in total
2. a) at least 4 credits in at least 4 areas **and**
   b) at least 60 credits in total
Examination - grades?

<table>
<thead>
<tr>
<th>Total credits</th>
<th>Grades in Swedish system</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 83</td>
<td>5</td>
</tr>
<tr>
<td>83-67</td>
<td>4</td>
</tr>
<tr>
<td>66-50</td>
<td>3</td>
</tr>
<tr>
<td>49-0</td>
<td>UK</td>
</tr>
</tbody>
</table>

- General
  - Literature
  - Personnel
  - Feedback

- Theory
  - Examination
  - Lectures & Exercises
  - QWA

- Project (only TDDC88)

- Labs (only TDDC88)
Lectures & Exercises

**Week 36**
- Lecture: Course Introduction and Overview
- Lecture: Software Engineering Roles (45min)
- Lecture: Requirements (45+ 45 min)
  - Readings and Exercise 1 (Requirements)

**Week 37**
- Deadline Exercise 1
- Lecture: Project Management (45+45 min)
- Lecture: Software Life Cycles and Processes (45min)
- Lecture: Agile methodologies (45min)
  - Readings and Exercise 2 (Planning and Processes)

**Week 38**
- Deadline Exercise 2
- Lecture: System Design and Architecture (45min)
  etc…
Lecture Exercises

- **Lecture Exercises**
  - Optional
  - 5 exercises, one for each knowledge area
  - Solve exercises in groups of 2 students
  - 0-4 extra credits on the exam for each exercise (max 20)
  - Possibility: Higher grades and easier to pass the exam

*Note! The credits are only valid on the exams in October 2014, January 2015, and August 2015*

For details, see course page under “Lecture Exercises”
Lectures - Project (only TDDC88)

Read the course page!!!

IMPORTANT!
Project information lecture
Thursday 4 September 13-15 in A1

Even more IMPORTANT
Role assignment
Thursday 4 September 15-17
Friday 5 September 15-17 (reserve time)
Lectures - Laboratory Exercises (only TDDC88)

General
- Literature
- Personnel
- Feedback

Theory
- Examination
- Lectures
- QWA

Project (only TDDC88)

Laboratory Exercises
- Lab 1 - Project Planning
- Lab 2 - Unified Modeling Language (UML)
- Lab 3 - Software Configuration Management (SCM)
- Lab 4 - Testing
- Lab 5 - SW metrics New lab 2014.

Labs (only TDDC88)
- One group = 2 students
- Oral and written examination
- Register in webreg now! (see homepage) Sign-up deadline September 4
- Deadline for all labs October 13

Problems? Contact Erik Hansson
A typical Software Project

How the customer explained it
A typical Software Project

How the project leader understood it
A typical Software Project

How the analyst designed it
A typical Software Project

How the programmer wrote it
A typical Software Project

How the business consultant described it
A typical Software Project

How the project was documented
A typical Software Project

What operations installed
How the customer was billed
A typical Software Project

How it was supported
A typical Software Project

Welcome to
the Software Engineering course 2014!

Finally, what the customer really needed