Course Introduction and Overview

Software Engineering

Lecture 1

Software Engineering
TDDC88/TDDC93
autumn 2013

Kristian Sandahl, Aseel Berglund (slides by David Broman)
Department of Computer and Information Science
Linköping University, Sweden
kristian.sandahl@liu.se
Who is Kristian?

Gotland

Småland

Linköping

Naval HQ

Protein purification

Ericsson

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information
Who is Aseel?

University lecturer

Industrial expertise in:
- Linköpings University
- SAAB
- COMBITECH
- MOTOROLA
- NOKIA
- SIEMENS

Different types of products

Building right products in an effective way

Irakiskmat.blogspot.com

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information

kristian.sandahl@liu.se
aseel.berglund@liu.se
Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information
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?
Software is the soul of Swedish Industry

- 80% of Ericsson's investment is Software (SEK 20,000,000,000/year)
- 25-35% of the value of a car is Software
- 70% of innovation in a truck is Software
- 75,000 people work directly with Software in Sweden

(source: www.swedsoft.se)
Agenda - What will you learn today?

Part I
What is Software Engineering?

Part II
A software life-cycle

Part III
Course Information
Part I
What is Software Engineering?
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

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information

kristian.sandahl@liu.se
aseel.berglund@liu.se
Finally, what the customer really needed
Larger disasters

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


Software Engineering

- Application of systematic, disciplined, quantifiable approach to software development, operation and maintenance of software. *(IEEE-Std.)*
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
Part II
Contents of a Software Life Cycle
Scope of a software life-cycle model

Idea → Software Product

Usage → Operation & maintenance → Replacement

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information

kristian.sandahl@liu.se
aseel.berglund@liu.se
Model of a life-cycle

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information
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 – Testing and SCM
- Week 40 – Software Quality
KA #1: Requirements (Week 36)

Lecture - Requirements

Collect user requirement

- Elicitation
- Analyze
- Specify

Understand

Document/build

Check that it matches user/customer requirements

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information

kristian.sandahl@liu.se
aseel.berglund@liu.se
KA #2: Planning and Processes (Week 37)

Lecture - Project Management

Resources  Calendar-time
Milestones  Time-budget  Activities

Plan

- Resource management - Harry
- Time-buffer, 3 months deliver
- Internal milestones - keep track

Risk Management

Identify  Analyze  Plan

probability - consequence

- Minimize
- Transfer
- Accept

Harry
the hacker

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information
Lecture - Software Life Cycles and Processes

Agile Methodologies

V-model
Waterfall model

Iterative models

Processes e.g. Scrum, Kanban, Lean, XP, RUP
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

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information

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

Requirements

System Design (Architecture, High-level Design)

Module Design (Program Design, Detailed Design)

Conny
the consultant

His "own" notations
KA #4: Testing and SCM (Week 39)

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information

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

Validation process:
- Validate Requirements, Verify Specification
- Verify System Design
- Verify Module Design
- Verify Implementation
KA #4: Testing and SCM

Validate Requirements, Verify Specification

Requirements

Acceptance Test (Release testing)

Lecture - Testing

Module 1

Module 2

Module 3

Integration testing
• Dependencies between modules

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

Lack of unit, integration and system testing

System Design
(Architecture, High-level Design)

Module Design
(Program Design, Detailed Design)

System Testing
(Integration testing of modules)

Implementation of Units (classes, procedures, functions)

Unit testing

Verify Implementation

Module Testing
(Integration testing of units)

Verify Module Design

Integration testing
• Dependencies between modules

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

Lack of unit, integration and system testing

Requirements

System Testing
(Architecture, High-level Design)

Module Design
(Program Design, Detailed Design)

System Testing
(Integration testing of modules)

Implementation of Units (classes, procedures, functions)

Unit testing

Verify Implementation

Module Testing
(Integration testing of units)

Verify Module Design

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

Part I: What is Software Engineering?

Part II: Contents of a Software Life Cycle

Part III: Course Information

kristian.sandahl@liu.se
aseel.berglund@liu.se

Verify System Design

Validate Requirements, Verify Specification

Acceptance Test (Release testing)

Verify System Design

Lecture - Software Configuration Management

Configuration Management (CM)

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

Carol the customer

Crash!
KA #5: Software Quality (Week 40)

Validate Requirements, Verify Specification

Acceptance Test (Release testing)

System Testing (Integration testing of modules)

Verify System Design

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

Different qualities
- Quality of product
- Quality of process
- Quality in a business environment
- Finding defects
- Learning organization

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information
Part III
Course Information
Course Goals

Give sound theoretical foundation of Software Engineering (TDDC88, 725G64, 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, 725G64, TDDD69)

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
What is the content of this course?

General
- Literature
- Personnel
- Feedback

Theory (TDDC88, TDDC93, 725G64)
- Examination
- Lectures & Exercises
- QWA

Project (TDDC88, TDDD69, 725G64)

Labs (TDDC88, 725G64)

www.ida.liu.se/~TDDC88
(same for TDDC93)
Literature (recommended)

General
- Literature
- Personnel
- Feedback

Theory
- Examination
- Lectures & Exercises
- QWA

Project

Labs

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

General
- Literature
- Personnel
- Feedback

Theory
- Examination
- Lectures & Exercises
- QWA

Project

Labs

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

Course Administrator
Liselotte Lundberg, liselotte.lundberg@liu.se

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

Laboratory Exercise Assistants
Erik Hansson, erik.hansson@liu.se
Kristian Stavåker, kristian.stavaker@liu.se
Jonas Wallgren, jonas.wallgren@liu.se

Project Supervisors and Examiners
CEO for the Company Lena Buffoni, lena.buffoni@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 Mohamed Abu Baker, Zenterio AB

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

General
- Literature
- Personnel
- Feedback

Theory
- Examination
- Lectures & Exercises
- QWA

Project

Labs

Muddy Cards, Tuesday 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
Changes since last year

Course evaluation feedback
- KURT (webbased system)
- Muddy cards
- Personal feedback – talk to us!
- Student representative feedback

Main improvements this year due to student feedback
- theory: Added lean-startup, rework software quality
- project: Support self reflection
- labs: One assistant more, debugging lab not mandatory
Feedback

General
- Literature
- Personnel
- Feedback

Theory
- Examination
- Lectures & Exercises
- QWA

Project

Labs

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information

kristian.sandahl@liu.se
aseel.berglund@liu.se
General
- Literature
- Personnel
- Feedback

Theory
- Examination
- Lectures & Exercises
- QWA

Project

Labs

<table>
<thead>
<tr>
<th>Course part</th>
<th>Examination</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory</td>
<td>Written Exam</td>
<td>2.5p, 4hp, 4 ECTS</td>
</tr>
<tr>
<td>Project</td>
<td>Project Tasks</td>
<td>4p, 6hp, 6 ECTS</td>
</tr>
<tr>
<td>Laboratory exercises</td>
<td>Oral Exam and Written Exercises</td>
<td>1.5p, 2hp, 2 ECTS</td>
</tr>
</tbody>
</table>

Written Exam
- Primary Exam 14:00-18:00, November 01, 2013
- Retake Exam 08:00-12:00, January 10, 2014
- Retake Exam August, 2014

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information
Examination - content? to pass?

**General**
- Literature
- Personnel
- Feedback

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

**Project**

**Labs**

**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.

**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?

### General
- Literature
- Personnel
- Feedback

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

### Project

### Labs

<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>
## Lectures & Exercises

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

### Week 37
- Seminar: Requirements
- Lecture: Project Management
- Lecture: Software Life Cycles and Processes
- Lecture: Agile methodologies
- Readings and Exercise 2 (Planning and Processes)

### Week 38
- Seminar: Planning and Processes
- Lecture: System Design and Architecture 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 2013, January 2014, and August 2014

For details, see course page under “Lecture Exercises”
Questions Without Answers (QWA)

- Hints on what to focus on
- Only fundamental part

Lectures - QWA

General
- Literature
- Personnel
- Feedback

Theory
- Examination
- Lectures & Exercises
- QWA

Project (only TDDC88)

Labs (only TDDC88)
Lectures - Project

Part I
What is Software Engineering?

Part II
Contents of a Software Life Cycle

Part III
Course Information

Read the course page!!!

IMPORTANT!
Project information lecture TODAY
(Monday 15-17, Visionen, B-house entrance 27)
### Laboratory Exercises

- Lab 1 – Basic User Application Programming
- Lab 2 – Project Planning
- Lab 3 – Unified Modeling Language (UML)
- Lab 4 – Software Configuration Management (SCM)
- Lab 5 – Testing

- One group = 2 students
- Oral and written examination
- Register in webreg now! (see homepage) Sign-up deadline **September 09**
- Deadline for all labs **October 18**
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
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