Introduction and Overview

Lecture 1

Kristian Sandahl
Department of Computer and Information Science
Linköping University, Sweden
Kristian.Sandahl@ida.liu.se

Software Engineering
CUGS course
Spring 2011

Who is Kristian?

Gotland
Småland
Linköping

Naval HQ
Protein purification
Ericsson

Part I
Definition and role of SE

Part II
A life cycle model
Part I
Definition and role of Software Engineering

Part II – A life cycle model

Kristian.Sandahl
@liu.se
A typical Software Project

How the customer explained it

Part I
Definition and role of SE

Part II
A life cycle model

How the project leader understood it

Part I
Definition and role of SE

Part II
A life cycle model
Part I
Definition and role of SE

Part II
A life cycle model

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

Part I
Definition and role of SE

Part II
A life cycle model

How the project was documented
Part I
Definition and role of SE

Part II
A life cycle model

A typical Software Project

What operations installed

A typical Software Project

How the customer was billed
A typical Software Project

How it was supported

Part I
Definition and role of SE

Part II
A life cycle model

Finally, what the customer really needed
Why do we need SE?

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*)
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

Part I
Definition and role of SE

Part II
A life cycle model

Studying SE theory

Your job
- Find central concepts and methods over the entire field
- Study their applicability, pros and cons
- Practice analysing and designing small examples
- Participate in examination
- Ask us questions
- Give us suggestions for further improvement

Our job
- Account for the most central concepts in lectures
- Set things in context
- Discuss industrial relevance
- Arrange examination
- Answer your questions
- Listen to feedback:

Kristian.Sandahl@ida.liu.se  0706-68 19 57  3B:470
Subject: DF14900
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, perhaps already obtained
- You will have a curious, but critical, attitude towards existing and new methods

Part II

Contents of a Software Life Cycle
**Part I**
Definition and role of SE

**Part II**
A life cycle model

---

**Scope of a software life-cycle model**

- **Idea**
- **Software Product**

- **Usage**
- **Operation & maintenance**
- **Replacement**

---

**Model of a life-cycle**

- **Time**
- **Abstraction**

- **Carol**
  the customer

- **Diana**
  the developer

---

**Part I**
Definition and role of SE

**Part II**
A life cycle model
Part I
Definition and role of SE

Part II
A life cycle model

Requirements

Elicitation
Collect user requirement

Analyze
Understand

Specify
Document

Validate
Check that it matches user requirements

Functional requirements
A required behavior or function
E.g. “delete material items”

Non-Functional requirements
Quality requirements
E.g. performance, scalability, usability maintainability, etc.
Part I
Definition and role of SE

Part II
A life cycle model

Requirements and Acceptance Test

Validate Requirements, Verify Specification

Requirements

Acceptance Test
(Release testing)

System Design

Validate Requirements, Verify Specification

Requirements

Acceptance Test
(Release testing)

System Design
(Architecture, High-level Design)

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

Module 1

Module 2

Module 3

Architecture styles, e.g.
- Client-server
- Layered Models
- Pipes and Filters

Part I
Definition and role of SE

Part II
A life cycle model
**System and Module Design**

- **Requirements**
- **System Design** (Architecture, High-level Design)
- **Module Design** (Program Design, Detailed Design)

---

**Validate Requirements, Verify Specification**

**Acceptance Test** (Release testing)

**Unified Modeling Language (UML)**

Use a standardized way to model system graphically

**Design Patterns**

Reuse design solution that has worked before

---

**Testing**

- **Requirements**
- **System Design** (Architecture, High-level Design)
- **Module Design** (Program Design, Detailed Design)

---

**Validate Requirements, VerifySpecification**

**Acceptance Test** (Release testing)

**System Testing** (Integration testing of modules)

**Module Testing** (Integration testing of units)

**Verify Module Design**

**Verify Implementation**

**Implementation of Units (classes, procedures, functions)**

**Unit testing**

---

**Part I**
Definition and role of SE

**Part II**
A life cycle model
Part I
Definition and role of SE

Part II
A life cycle model

Testing

- Validate Requirements, Verify Specification
- Acceptance Test (Release testing)

Integration testing
- Dependencies between modules

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

Supporting Processes

- Validate Requirements, Verify Specification
- Acceptance Test (Release testing)

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

Part I
Definition and role of SE

Part II
A life cycle model
Part I
Definition and role of SE

Part II
A life cycle model


Software Life Cycles and Configuration Management

V-model
Waterfall model

Iterative models

Configuration Management (CM)

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

Inspection

Formal Inspection
- Reviewing with a defined process
- Aim at finding defects
- Works in practice
Part I
Definition and role of SE

Part II
A life cycle model

A Software Life-cycle

Requirements
Validate Requirements, Verify Specification

System Design
(Architecture, High-level Design)
Verify System Design
System Testing
(Integration testing of modules)

Module Design
(Program Design, Detailed Design)
Verify Module Design
Module Testing
(Integration testing of units)

Implementation
of Units (classes, procedures, functions)
Unit testing

Acceptance Test
(Release testing)

Project Management, Software Quality Assurance (SQA), Supporting Tools, Education

Summary

Requirements
Validate Requirements, Verify Specification

System Design
(Architecture, High-level Design)
Verify System Design
System Testing
(Integration testing of modules)

Module Design
(Program Design, Detailed Design)
Verify Module Design
Module Testing
(Integration testing of units)

Implementation
of Units (classes, procedures, functions)
Unit testing

Acceptance Test
(Release testing)

Part I
Definition and role of SE
Part II
A life cycle model