Software Engineering

Lecture 01.2: Introduction to Software Engineering

Dietmar Pfahl
email: dietmar.pfahl@ut.ee

Fall 2021
Schedule of Lectures

**Week 01: Introduction to SE**
**Week 02: Requirements Engineering I**
**Week 03: Requirements Engineering II**
**Week 04: Analysis**
**Week 05: Development Infrastructure**
**Week 06: Continuous Development and Integration**
**Week 07: Project Estimation / Architecture and Design I**
**Week 08: Architecture and Design II**

**Week 09: Verification and Validation I**
**Week 10: Verification and Validation II**
**Week 11: Refactoring (and TDD)**
**Week 12: Agile/Lean Methods**
**Week 13: Industry Guest Lecture**
**Week 14: Course wrap-up, review and exam preparation**

**Week 15: Reserve time slot (no lecture scheduled as of today)**
Software Engineering

What?

Why?
Software Development – Three Ps

- Software Development

Project or Iteration

P ?
P ?
P ?
Software Development – Three Ps

- Software Development
- Products
- People
- Processes

Project or Iteration
Software Development – Three Ps

• Software Development

Project or Iteration

Products

People

Processes
Products in Software Development
Products in Software Development

class ISort
{
    public static void insertionSort(int[] a, String direction)
    {
        int in, out;
        for(out=1; out<a.length; out++)
        {
            int temp = a[out];
            in = out;
            while((in>0 && a[in-1] >= temp && direction.equalsIgnoreCase("A"))
                  || (in>0 && a[in-1] <= temp && direction.equalsIgnoreCase("D")))
            {
                a[in] = a[in-1];
                --in;
            }
            a[in] = temp;
        }
    }
}
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
- ...

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

**Types of Software:**
- Embedded/real-time
- Information System
- Web application
- Mobile application
- Systems of systems
- ...

**Models**
Software in a Car

State-of-practice:
- 40-100 ECUs
- 5-10 Mio Lines of Code
- > 100 MB Software
- > 3 Bus Systems

ECU = Electronic Control Unit

+ Autonomous driving
Secure cloud communication

Source:
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

Products

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

System Requirements → Software Requirements → Preliminary Program Design → Analysis → Program Design → Coding → Testing → Usage → Preliminary Design → Analysis → Program Design → Coding → Testing → Operations

(Royce, 1970)
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)
  - Lean
  - Agile
  - Kanban

Processes
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

Waterfall

Implementation

Design

Requirements

Functionality

Incremental, e.g. RUP

Agile - XP

RUP = Rational Unified Process
XP = Extreme Programming
Survey – Software Development Processes and Practices

• 26 Countries

• 500+ Responses (from Industry!)

• URL: https://helenastudy.wordpress.com
Process Frameworks Used in 26 Countries

<table>
<thead>
<tr>
<th>Frameworks/Methods</th>
<th>Often Used</th>
<th>Always Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum</td>
<td></td>
<td>26%</td>
</tr>
<tr>
<td>Iterative Development</td>
<td>23%</td>
<td></td>
</tr>
<tr>
<td>Kanban</td>
<td>17%</td>
<td></td>
</tr>
<tr>
<td>DevOps</td>
<td>11%</td>
<td></td>
</tr>
<tr>
<td>Classic Waterfall Process</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>eXtreme Programming (XP)</td>
<td>9%</td>
<td></td>
</tr>
<tr>
<td>Lean Software Development</td>
<td>8%</td>
<td></td>
</tr>
<tr>
<td>Domain-Driven Design</td>
<td>6%</td>
<td></td>
</tr>
<tr>
<td>Feature Driven Development (FDD)</td>
<td>5%</td>
<td></td>
</tr>
<tr>
<td>V-shaped Process (V-Model)</td>
<td>4%</td>
<td></td>
</tr>
<tr>
<td>Phase / Stage-gate model</td>
<td>3%</td>
<td></td>
</tr>
<tr>
<td>Model-Driven Architecture (MDA)</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Scaled Agile Framework (SAFe)</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Team Software Process</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Personal Software Process</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Nexus</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Large-Scale Scrum (LESS)</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>SSADM</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Spiral Model</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Dynamic Systems Development Method</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Crystal Family</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>PRINCE2</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Rational Unified Process</td>
<td>2%</td>
<td></td>
</tr>
</tbody>
</table>

1. Scrum
2. Iterative Development
3. Kanban
4. DevOps (CI/CD)
5. Classic Waterfall
Process Frameworks Used in Estonia

1. Scrum
2. DevOps (CI/CD)
3. Iterative Development
4. Kanban
5. XP
### Process Frameworks Used in Sweden

#### Which of the following frameworks and methods do you use?

<table>
<thead>
<tr>
<th>Frameworks/Methods</th>
<th>Often Used</th>
<th>Always Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum</td>
<td>46%</td>
<td>8%</td>
</tr>
<tr>
<td>Iterative Development</td>
<td>23%</td>
<td>15%</td>
</tr>
<tr>
<td>Kanban</td>
<td>23%</td>
<td>15%</td>
</tr>
<tr>
<td>DevOps</td>
<td>38%</td>
<td>8%</td>
</tr>
<tr>
<td>Classic Waterfall Process</td>
<td>46%</td>
<td>15%</td>
</tr>
<tr>
<td>eXtreme Programming (XP)</td>
<td>8%</td>
<td>8%</td>
</tr>
<tr>
<td>Lean Software Development</td>
<td>23%</td>
<td>23%</td>
</tr>
<tr>
<td>Domain-Driven Design</td>
<td>38%</td>
<td>8%</td>
</tr>
<tr>
<td>ScrumBan</td>
<td>46%</td>
<td>8%</td>
</tr>
<tr>
<td>Feature Driven Development (FDD)</td>
<td>23%</td>
<td>15%</td>
</tr>
<tr>
<td>V-shaped Process (V-Model)</td>
<td>31%</td>
<td>8%</td>
</tr>
<tr>
<td>Phase / Stage-gate model</td>
<td>15%</td>
<td>15%</td>
</tr>
<tr>
<td>Model-Driven Architecture (MDA)</td>
<td>15%</td>
<td>15%</td>
</tr>
<tr>
<td>Scaled Agile Framework (SAFe)</td>
<td>15%</td>
<td>8%</td>
</tr>
<tr>
<td>Team Software Process</td>
<td>62%</td>
<td>8%</td>
</tr>
<tr>
<td>Personal Software Process</td>
<td>6%</td>
<td>8%</td>
</tr>
<tr>
<td>Nexus</td>
<td>23%</td>
<td>8%</td>
</tr>
<tr>
<td>Large-Scale Scrum (LESS)</td>
<td>31%</td>
<td>15%</td>
</tr>
<tr>
<td>SSADM</td>
<td>54%</td>
<td>8%</td>
</tr>
<tr>
<td>Spiral Model</td>
<td>23%</td>
<td>23%</td>
</tr>
<tr>
<td>Dynamic Systems Development Method</td>
<td>15%</td>
<td>23%</td>
</tr>
<tr>
<td>Crystal Family</td>
<td>77%</td>
<td>8%</td>
</tr>
<tr>
<td>PRINCE2</td>
<td>54%</td>
<td>8%</td>
</tr>
<tr>
<td>Rational Unified Process</td>
<td>62%</td>
<td>8%</td>
</tr>
</tbody>
</table>

1. Scrum
2. Iterative Development
3. Kanban
4. Classic Waterfall Process
5. XP
Company Sizes of Respondents: Estonia vs. Sweden vs. 26 Countries

Company Size
What is your company’s size in equivalent fulltime employees (FTEs)?

- All the countries
- Estonia
- Sweden

Legend:
- Not Answered
- Small (11 - 50 employees)
- Medium (51 - 250 employees)
- Large (251 - 2499 employees)
- Very Large (>2500 employees)
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 of SE

- 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

Developer (SW Engineer)

Software Product/ System
Next Lecture

• Date/Time:
  - Friday, 10-Sep, 10:15-12:00
• Topic:
  - Requirements Engineering I \( \rightarrow \) 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!