Lecture 01.2: Introduction to Software Engineering

Dietmar Pfahl
email: dietmar.pfahl@ut.ee

Fall 2017
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: Crowdsourced Testing

Week 11: Continuous Development and Integration
Week 12: Agile/Lean Methods
Week 13: Software Craftsmanship
Week 14: Course wrap-up, review and exam preparation
Week 15: no lecture
Software Engineering

What?
Why?
Software Development – Three Ps

- Software Development

Project or Iteration

P ?

P ?

P ?
Software Development – Three Ps

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

Who has practiced SE?
Who knows what SE is?
Software Development – Three Ps

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

**Products**

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

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

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
- ...

Skills:
- Must match roles

Training:
- Must fill skill-gaps

Education:
- Curricula (ACM/IEEE)

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

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)
Processes in SW Development

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

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

**Process Types:**
- Heavy-weight (rich)
- Light-weight
  - 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

Functionality

Time

Requirements
Design
Implementation
Testing

Waterfall
Incremental, e.g. RUP
Agile - XP

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

• 26 Countries

• 500+ Responses

• URL: https://helenastudy.wordpress.com

RUP = Rational Unified Process
XP = Extreme Programming
Process Frameworks Used in Estonia

Frameworks/Methods
Which of the following frameworks and methods do you use?

- Scrum: 58%
- Iterative Development: 20%
- Kanban: 8%
- DevOps: 8%
- Classic Waterfall Process: 8%
- eXtreme Programming (XP): 8%
- Lean Software Development: 8%
- Domain-Driven Design: 8%
- ScrumBan: 8%
- Feature Driven Development (FDD): 8%
- V-shaped Process (V-Model): 8%
- Phase / Stage-gate model: 8%
- Model-Driven Architecture (MDA): 8%
- Scaled Agile Framework (SAFe): 8%
- Team Software Process: 8%
- Personal Software Process: 8%
- Nexus: 8%
- Large-Scale Scrum (LESS): 8%
- SSADM: 8%
- Spiral Model: 8%
- Dynamic Systems Development Method: 8%
- Crystal Family: 8%
- PRINCE2: 8%
- Rational Unified Process: 8%

Legend:
- Do not know the framework
- Do not know if we use it
- We rarely use it
- We sometimes use it
- We often use it
- We always use the framework
Process Frameworks Used in Sweden

Frameworks/Methods
Which of the following frameworks and methods do you use?

- Scrum: 40% use it, 38% sometimes use it, 8% rarely use it, 8% never use it
- Iterative Development: 46% use it, 15% sometimes use it, 8% rarely use it, 8% never use it
- Kanban: 38% use it, 15% sometimes use it, 15% rarely use it, 8% never use it
- DevOps: 31% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Classic Waterfall Process: 23% use it, 15% sometimes use it, 15% rarely use it, 8% never use it
- eXtreme Programming (XP): 23% use it, 15% sometimes use it, 15% rarely use it, 8% never use it
- Lean Software Development: 15% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Domain-Driven Design: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- ScrumBan: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Feature Driven Development (FDD): 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- V-shaped Process (V-Model): 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Phase / Stage-gate model: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Model-Driven Architecture (MDA): 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Scaled Agile Framework (SAFe): 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Team Software Process: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Personal Software Process: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Nexus: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Large-Scale Scrum (LESS): 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- SSADM: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Spiral Model: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Dynamic Systems Development Method: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Crystal Family: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- PRINCE2: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
- Rational Unified Process: 23% use it, 38% sometimes use it, 15% rarely use it, 8% never use it
Process Frameworks Used in 26 Countries

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

- Scrum
- Iterative Development
- Kanban
- DevOps
- Classic Waterfall Process
- eXtreme Programming (XP)
- Lean Software Development
- Domain-Driven Design
- ScrumBan
- Feature Driven Development (FDD)
- V-shaped Process (V-Model)
- Phase / Stage-gate model
- Model-Driven Architecture (MDA)
- Scaled Agile Framework (SAFe)
- Team Software Process
- Personal Software Process
- Nexus
- Large-Scale Scrum (LESS)
- SSADM
- Spiral Model
- Dynamic Systems Development Method
- Crystal Family
- PRINCE2
- Rational Unified Process

- Do not know the framework
- We never use it
- We rarely use it
- We sometimes use it
- We often use it
- We always use the framework

UNIVERSITY OF TARTU
INSTITUTE OF COMPUTER SCIENCE
Dev. Practices Used in Estonia
Dev. Practices Used in Sweden
Dev. Practices Used in 26 Countries

<table>
<thead>
<tr>
<th>Practices</th>
<th>Countries Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding standards</td>
<td>26</td>
</tr>
<tr>
<td>Code review</td>
<td>26</td>
</tr>
<tr>
<td>Prototyping</td>
<td>26</td>
</tr>
<tr>
<td>Refactoring</td>
<td>26</td>
</tr>
<tr>
<td>Release planning</td>
<td>26</td>
</tr>
<tr>
<td>Automated Unit Testing</td>
<td>26</td>
</tr>
<tr>
<td>Expert/Team based estimation</td>
<td>26</td>
</tr>
<tr>
<td>Design Reviews</td>
<td>26</td>
</tr>
<tr>
<td>Backlog Management</td>
<td>26</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>26</td>
</tr>
<tr>
<td>User Stories</td>
<td>26</td>
</tr>
<tr>
<td>Architecture Specifications</td>
<td>26</td>
</tr>
<tr>
<td>Iteration/Sprint Reviews</td>
<td>26</td>
</tr>
<tr>
<td>Limit Work-in-Progress</td>
<td>26</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>26</td>
</tr>
<tr>
<td>Daily Standup</td>
<td>26</td>
</tr>
<tr>
<td>Continuous deployment</td>
<td>26</td>
</tr>
<tr>
<td>Detailed Designs</td>
<td>26</td>
</tr>
<tr>
<td>Definition of done / ready</td>
<td>26</td>
</tr>
<tr>
<td>Formal estimation</td>
<td>26</td>
</tr>
<tr>
<td>Security Testing</td>
<td>26</td>
</tr>
<tr>
<td>Burn-Down Charts</td>
<td>26</td>
</tr>
<tr>
<td>Pair Programming</td>
<td>26</td>
</tr>
<tr>
<td>End-to-End (System) Testing</td>
<td>26</td>
</tr>
<tr>
<td>Collective code ownership</td>
<td>26</td>
</tr>
<tr>
<td>Test-driven Development (TDD)</td>
<td>26</td>
</tr>
<tr>
<td>Use Case Modeling</td>
<td>26</td>
</tr>
<tr>
<td>Automated Code Generation</td>
<td>26</td>
</tr>
<tr>
<td>On-Site Customer</td>
<td>26</td>
</tr>
<tr>
<td>Velocity-based planning</td>
<td>26</td>
</tr>
<tr>
<td>Iteration Planning</td>
<td>26</td>
</tr>
<tr>
<td>Destructive Testing</td>
<td>26</td>
</tr>
<tr>
<td>Scrumb-of-Scrums</td>
<td>26</td>
</tr>
<tr>
<td>Model Checking</td>
<td>26</td>
</tr>
<tr>
<td>Formal Specification</td>
<td>26</td>
</tr>
<tr>
<td>Automated Theorem Proving</td>
<td>26</td>
</tr>
</tbody>
</table>

| Do not know it                                  |        |
| Do not know if we use it                        |        |
| We never use it                                |        |
| We rarely use it                               |        |
| We sometimes use it                            |        |
| We often use it                                |        |
| We always use it                               |        |
Company Sizes of Respondents: Estonia vs. Sweden vs. 26 Countries

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

All the countries

Estonia

Sweden

- Not Answered
- Small (11 - 50 employees)
- Large (251 - 2499 employees)
- Micro (<10 employees)
- Medium (51 - 250 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, 15-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!