CSEPM - A Continuous Software Engineering Process Metamodel
RCoSE 2017

Stephan Krusche and Bernd Bruegge (May 22, 2017)
Introduction

• The term “software engineering” (SE) was proposed in the 1960s to overcome “chaos” (software crisis) using a defined process control model.

• In the 1990s, agile methods changed the emphasis from defined to empirical process control.

**Defined Process**
- Planned
- Follows strict rules
- Avoids deviations

**Empirical Process**
- Not entirely planned
- Inspect and adapt
Motivation

- Jez Humble defined a model for continuous delivery: the goal is to keep software in a state so that changes to it can be released at any time.

- Jan Bosch coined the term continuous software engineering (CSE).

- However, there has not yet been an attempt to model software engineering continuously.

- **Main objective:** create a software process metamodel to describe the continuous and unexpected nature of SE through workflows that can be interrupted by change events.

  ➡ **Continuous Software Engineering Process Metamodel: CSEPM**

  ➡ Improve communication about CSE and formalize tailoring, customization, and extension possibilities.
CSEPM in the Meta Object Facility

<table>
<thead>
<tr>
<th>MOF Layer</th>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M3 - Meta Metamodel</strong></td>
<td>Meta Object Facility</td>
<td><img src="image" alt="Class" /></td>
</tr>
<tr>
<td><strong>M2 - Metamodel</strong></td>
<td>CSEPM</td>
<td><img src="image" alt="Work Queue" /> to <img src="image" alt="Work Item" /></td>
</tr>
<tr>
<td>(Generic concepts)</td>
<td></td>
<td><img src="image" alt="Sprint Backlog" /> to <img src="image" alt="Backlog Item" /></td>
</tr>
<tr>
<td><strong>M1 - Model</strong></td>
<td>Process Model</td>
<td><img src="image" alt="Sprint 1 Backlog" /></td>
</tr>
<tr>
<td>(Way of working)</td>
<td></td>
<td>ID</td>
</tr>
<tr>
<td><strong>M0 - Reality</strong></td>
<td>Concrete Process</td>
<td>1</td>
</tr>
<tr>
<td>(What actually happens)</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>
CSEPM: Static process metamodel (simplified)
CSEPM: Lifecycle for a Workflow

Start → Sleep

Rugby Receive Event → Rugby Send Event
activate → sleep

Active → Finished
stop

Blocked → Resume
interrupt → resume

End
CSEPM: Lifecycle for a Workflow (Example)

Analysis
- New Requirement
- Feature Change

Design
- Design Change
- Architecture Change

Implementation Change

Start
- Rugby Receive Event
  - activate
  - interrupt
  - Rugby Send Event
  - sleep
  - resume

Active
- stop

Blocked
- End

New Requirement

Design Change
CSEPM: Dynamic process metamodel

- **Project Start**
  - Workflow Start
  - Start Workflow
  - Workflow
  - Subscribe to Events
    - Sleep

- **Incoming Event**
  - Activate Workflow
  - Perform Work
    - Work finished?
      - No
      - Yes
        - Outgoing Event
          - Sleep

- **Workflow Customization**
  - Stop Workflow
  - Customize Workflow
  - Workflow Start
    - Sleep

- **Project End**
  - Stop Workflow
  - Project End
CSEPM Instance of a Sequential Model: Waterfall

<table>
<thead>
<tr>
<th>Requirements Elicitation</th>
<th>Analysis</th>
<th>...</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Start</td>
<td>Project Start</td>
<td>...</td>
<td>Project End</td>
</tr>
<tr>
<td>Elicit requirements</td>
<td>Requirements Elicitation finished</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem Statement</td>
<td>Analyze</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements Elicitation finished</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis finished</td>
<td>Testing finished</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sleep</td>
<td>Deploy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deployed Software</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software deployed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project End</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stephan Krusche and Bernd Bruegge — CSEPM - A Continuous Software Engineering Process Metamodel
CSEPM Instance of an Iterative Model: Unified Process

<table>
<thead>
<tr>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Start</td>
<td>Lifecycle Objective</td>
<td>Lifecycle Architecture</td>
<td>Initial Operational Capability</td>
</tr>
<tr>
<td>Iterate</td>
<td>Iterate</td>
<td>Iterate</td>
<td>Iterate</td>
</tr>
<tr>
<td>Iteration needed?</td>
<td>Iteration needed?</td>
<td>Intermediate Release</td>
<td>Iteration needed?</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Candidate Architecture
Executable Architecture
Lifecycle Objective
Lifecycle Architecture
Initial Operational Capability
Final Release
Product Release
Project End
CSEPM Instance of an Agile Model: Scrum

Project Start → Kickoff Meeting → Create Product Backlog → Initial Product Backlog → Product Backlog created

Create Sprint Backlog → Sprint Planning Meeting → Sprint Review Meeting

Sprint N

Development done? No → Yes → Build Product Increment → Product Increment N

Develop Backlog Item → Sprint Backlog N

Sprint Review Meeting → Feedback Report

Time based Release

Project finished? Yes → No → Project End

Yes → Project End

Feedback Report
CSEPM Instance of a Continuous Model: Rugby (simplified)

Requirements Elicitation
- New Backlog Item
- Change Request
- Improvement Request
- Backlog Item ready for development

Development
- Specify
- Prioritize
- Analyze
- Design
- Implement
- Test
- Finished?
- Merge Request
- Need feedback?
- Release Request
- Release Request
- Improvement Request

Review Management
- Merge Request
- Quality ok?
- Merge
- Backlog Item finished

Release Management
- Release Request
- Release
- Use
- Feedback Report

Feedback Management
- Feedback Report
- Analyze Feedback
- Is new requirement?
- New Backlog Item
- Change Request

Stephan Krusche and Bernd Bruegge — CSEPM - A Continuous Software Engineering Process Metamodel
Conclusion

- CSEPM is an empirical process metamodel for continuous software engineering
- It also allows to model instances for sequential, iterative and agile process models
- It treats development activities as parallel running workflows to enable rapid cycles
- It allows tailoring, customization and extension

➡ CSEPM helps to understand the continuous and unexpected nature of software engineering
CSEPM - A Continuous Software Engineering Process Metamodel

RCoSE 2017

Stephan Krusche and Bernd Bruegge (May 22, 2017)