A Case Study on the Evolution of a Component-based Product Line

Wolfgang Heider\textsuperscript{1}, Michael Vierhauser\textsuperscript{2}, Daniela Lettner\textsuperscript{1}, Paul Grünbacher\textsuperscript{1}

\textsuperscript{1} Christian Doppler Lab. for ASE, Johannes Kepler University
\textsuperscript{2} Siemens VAI Metals Technologies
Linz, Austria
August 23, 2012
Context, Motivation, Goal

• Maintenance and evolution of product lines is increasingly complex

• We developed a product line engineering tool suite, the DOPLER tool suite

• We observed the evolution of this industrial product line while it was refactored for industrial use cases by Siemens VAI

→ Analyze developer challenges and tool support requirements
Involved Artifacts

- Dependency
- Component
- Interface
Involved Artifacts

Problem Space
- Product Line Feature
- Product Line Release

Solution Space
- Dependency
- Component
- Interface
Involved Artifacts
Involved Artifacts

Domain Engineering Artifacts
- Product Line Feature
- Product Line Release

Application Engineering Artifacts
- Derived Product

Problem Space

Solution Space
- Dependency
- Component
- Interface

Generated Executable Product
- Components

generate
reuse
Involved Artifacts

- Component Interface
- Product Line
- Feature Dependency
- Derived Product
- Solution Space
- Problem Space
- Dependency
- Component
- Interface
- Generated Executable Product
- Components
- generated
- reuse
- Application Engineering Artifacts
- Domain Engineering Artifacts
- Product Line Feature
- Product Line Release
- Derived Product

Diagram showing the involvement of artifacts in the context of domain engineering and application engineering, with arrows indicating relationships such as "generate" and "reuse."
Research Questions

Goal:  Developer challenges and tool support requirements

RQ1:  Which basic development activities are performed by developers to evolve the product line?

RQ2:  Which types of change impact analyses are performed by developers?

RQ3:  Which tool features are needed to perform these analyses?
The Case

Project to Refactor the Tool Suite

… a project for transitioning the research tool to industrial settings

• Siemens VAI project
• Siemens managers and business users were involved as stakeholders
• 4 hired developers
• Researchers advised and observed development
• 30 person-months of professional development
• 6 derived products that need to be maintained

<table>
<thead>
<tr>
<th></th>
<th>v2.0</th>
<th>v2.10</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoC (measured with cloc 1.53)</td>
<td>192,427</td>
<td>154,924</td>
<td>-20%</td>
</tr>
<tr>
<td>Java Classes</td>
<td>1,902</td>
<td>1,776</td>
<td>-6%</td>
</tr>
<tr>
<td>Eclipse Plugins, Features</td>
<td>61</td>
<td>101</td>
<td>+66%</td>
</tr>
</tbody>
</table>
Research Process

TeamForge Artifacts
(epics, stories, tasks)

Daily SCRUM Meeting Protocols
(action items, open issues)

Changes relevant to PL evolution

categorize by

Development Activities

Impact Analyses

derive

PLE Scenarios

derive

derive

Trace Information Model & Desired Trace Links

select

select

map to

derive

categorize by

categorize by
Research Process

TeamForge Artifacts (epics, stories, tasks)

Daily SCRUM Meeting Protocols (action items, open issues)

select

Changes relevant to PL evolution

select

categorize by

Development Activities

- Add / change / remove component
- Add / change / remove interface
- Add / remove dependency
- Define release

RQ1
Changes relevant to PL evolution

TeamForge Artifacts
(epics, stories, tasks)

Daily SCRUM Meeting Protocols
(action items, open issues)

- Which **components** are affected?
- Which **interfaces** are changed?
- Which **dependencies** are added?
- Which **features** do change?
- Which **product** files need to be adapted?
Research Process

TeamForge Artifacts (epics, stories, tasks)  
Daily SCRUM Meeting Protocols  
(action items, open issues)

Changes relevant to PL evolution

Development Activities

RQ1: classify by
- Add feature
- Remove feature
- Define feature variability
- Maintain architecture
- Maintain assets

RQ2: map to
PLE Scenarios
Changes relevant to PL evolution
TeamForge Artifacts
(epics, stories, tasks)
Daily SCRUM Meeting Protocols
(action items, open issues)

Development Activities
Impact Analyses
Trace Information Model & Desired Trace Links

RQ1
RQ2
RQ3

Changed artifact type → Analyzed artifact type
Research Process

| TeamForge Artifacts (epics, stories, tasks) | Daily SCRUM Meeting Protocols (action items, open issues) |

**Epic:** Refactor Configuration Workflow  
**Story:** Add additional authentication  
**Task:** Define authentication extension point

**Action Item:** Develop authentication interface  
**Open Issues:** How to adapt old authentication?  
Which products need which authentication?

**Impediment:** The fu*beep* build server is offline!
Research Process

TeamForge Artifacts (epics, stories, tasks) select

Daily SCRUM Meeting Protocols (action items, open issues) select

Changes relevant to PL evolution

Add Smartcard Authentication
Research Process

- **Change** configuration UI plugin
- **Add** authentication interface
- **Add** authentication plugins
Research Process

- Which **plugins** are affected by the new authentication?
- Which **product** files need to be adapted to include the new authentication?
Research Process

TeamForge Artifacts (epics, stories, tasks)

select

categorize by

Development Activities

Daily SCRUM Meeting Protocols (action items, open issues)

select

categorize by

Impact Analyses

map to

PLE Scenarios

- Add authentication feature
- Define authentication variability
While defining feature variability, the configuration UI plugin was changed and developers analyzed which product files need to be adapted?
Resulting Trace Information Model

Problem Space
- Product Line Feature
- Product Line Release

Solution Space
- Dependency
- Component
- Interface

Generated Executable Product
- Components
Resulting Trace Information Model

```
Component Interface
Product Line Feature
Product Line Release
Derived Product

Dependency
Generated Executable Product

Solution Space
Problem Space

Domain Engineering Artifacts
Application Engineering Artifacts

Product Line
Feature Dependency
Derived Product
Component Interface
```

August 23, 2012 | Christian Doppler Laboratory for Automated Software Engineering
Resulting Trace Information Model
Developers need to see the effects of changes on features and feature variability.
The tool presents features related to a component under change.
The tool presents affected features and violated feature variability (if dependencies between components are added or removed).
Resulting Trace Information Model

→ Developers need to see the impact of changes on derived products

Domain Engineering Artifacts

Problem Space

Product Line Feature

Product Line Release

Derived Product

Application Engineering Artifacts

Generating Executable Product

Components

Solution Space

Dependency

Component

Interface
The tool presents already derived products or ongoing product derivations based on a previous PL release to analyze the possibility of product updates.
Before we developed approaches and solutions …

• We conducted a case study to find industrial challenges
• We observed the evolution of an industrial product line
• We derived scenarios of impact analysis as requirements

• Our approach to provide impact analysis: Regression testing with variability models [Heider et al. @SPLC2012]