Agile Product Liner DSL

Test-based feature management
Content

- Motivation
- Agile Product Liner DSL
- Future Work
Agile Product Line Engineering

- Integration of two popular practices:
  - Agile Software Development
  - Software Product Lines

- Different approaches and targets:
  - SPL-based organizations that want to be more agile
  - Agile organizations that want to establish a SPL
Acceptance tests as bridge

- In test-driven development in agile teams:
  - Feature development driven by ATs
- ATs are
  - traceable design artifacts
  - core assets that can be reused
- Approach:
  - Feature model popular to describe SPLs
  -> Test-based feature management
Tool support

- tool support for APLE
  - Integrate feature modeling and acceptance tests
    → Test-based feature management
- Prerequisites
  - Visual Studio 2010 extension
  - C#
Many findings:
- Java & Eclipse

Only two tools for Visual Studio:
- Feature Model DSL
- Feature Model Tool

Source code, open source license:
- Only Feature Model DSL
Feature Model DSL

- Based on the DSL Tools
- Graphical feature modeling
  - Features (mandatory, optional)
  - Alternatives (multiplicity)
- Cross-diagram navigation
- Feature model validation
- Configuration tool window
  - Load/Save configurations
  - Custom actions interface
  - Creating HTML reports for configuration
DSL Tools

- DSL → Domain-specific language
- DSL Tools extend the Visual Studio SDK
- Framework offers
  - Graphical domain-specific (modeling) language creation
  - Integration into the Visual Studio environment
    - Designer
    - Tool windows
    - Code generation
Agile Product Liner DSL

- Based on FMD
  - Extend DSL of FMD
    - Tests
    - Relationship between tests and features
    - Exclude constraint between features
    - Collapsing and expand of tests
- Extend configuration tool window of FMD
  - Tests
- Mapping of tests in the model to test files
Extending GreenPepe2010
- Export of functionality via MEF
  - Query of tests in the solution
  - Execution of tests

Consuming in APLD
- Test mapping
- Test execution from model and configuration tool window
- Presenting test results in model and configuration
Scope of work

Agile Product Liner DSL
- Modeling extensions
- Test mapping
- Configuration tool window extensions

Feature Model DSL

GreenPepe 2010
Inter plug-in communication

Export test execution and recognition

GreenPepe 2010 core
Future Work

- Instantiation Process
  - AT execution ➔ code coverage ➔ Build
  - Model and configuration are persisted in XML
    - Process configuration in other programs
      - XML schema optimized for instantiation

- Supporting cross-diagram references
End