Variability Smells in Software Product Lines

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Variability Smells

- A variability smell is an undesirable property of an SPL artifact
- Variability smells may be related to different SPL artifacts
  - Feature models
  - Domain documents
  - Source code, etc.

List of Variability Smells

Obscure Feature Model	**Unused Variability**	**Unused Feature**
Dead Feature	**Dead Feature Code**	Fat Products
Dependency Clusters	**#ifdef Hell**	Runtime Overhead
Binary Chaos	Traceability Mess	Limited Extensibility
Duplicated Code in Alternative Features	Variability-Mechanism Overload	

Obscure Feature Model

- The feature model is overly complex and hard to understand
- This smell results in an unsuitable structure of the feature model
  - Wrong mapping of requirements to features
  - Suboptimal naming and documentation of features

Unused Variability

- A feature is modeled as optional, but it is selected in all current products
- It can be an indicator of variability that is not actually needed
  - Sometimes we may decide to keep the feature optional for future costumers

Unused Feature

- This smell is the opposite of Unused Variability
- A feature is optional, but it is not selected in any current product
- Unused feature can be an indicator of unnecessary code

Dead Feature

- A feature modeled as optional, but not selectable due to existing constraints
  - For instance, a mutually exclusion constraint with a mandatory feature
- Dead feature may be an indicator of an incorrect feature model or feature no longer needed

Dead Feature Code

- A code fragment is dead if it can never be included in any derived product
  - Not even if the feature is selected
- Dead Feature Code is common in annotative implementation strategies
  - For instance, Parameters and Conditional Compilation
- It may be an indicator of wrong mapping from the feature model to code

Duplicated Code in Features

- It occurs when two (alternative) features replicate code in their implementations
  - This variability smell is similar to the traditional smell Duplicated Code
- A solution is extracting the common code into a shared implementation unit
  - Both feature can use this shared unit

#ifdef Hell

- The implementation of a feature is scattered across the code base
  - This variability smell is similar to the traditional smell Shotgun Surgery
- This smell can be an indicator of poor separation of concerns
  - A solution could be redesign the SPL change the implementation strategy

Bibliography

- S. Apel, D. Batory, C. Kastner, G. Saake. **Feature-Oriented Software Product Lines: Concepts and Implementation**. Springer; 2013.
  - Section 8.2