Quality Assurance by Means of Feature Models

David Gollasch
Motivation

Modern business applications are getting increasingly distributed over the Internet as multi-tenant software as a service (SaaS). This leads to new challenges in terms of quality assurance when developing or maintaining such applications, because all customers are directly affected very often.

Which effects do software changes have?

How are those changes proactively determinable?

How useful are feature models to face this?
1. Fundamentals

SaaS

Quality Assurance
Multi-Tenant Software as a Service Applications

...can be seen as a special kind of SPL
Why to focus on quality assurance?

**Laws of software evolution** (development of software in time)

1. Law of continuous change
2. Law of increasing complexity
3. Law of decreasing quality
4. Law of diminishing productivity
5. Law of restricted growth

---

*Sneed 2012, EVOL-21*
How to assure quality in general?

- Take a **quality goal** and try to reach it due to further development.

- Prioritize development process: Remove unnecessary features to **avoid wasting time and money**.

- **X** exclusive
- **G** gradual
- **Q** quantized

- Analyse product structure
2. Quality Assurance by Means of Feature Models

- Quality as Attributes in Feature Models
- Structural Analysis of Feature Models
Qualities as Attributes in Feature Models

Steps to analyze quality goals:

1. Classification as X, G or Q goal
2. Find metric or heuristics
3. Consolidation for configuration

<table>
<thead>
<tr>
<th>Type</th>
<th>Quality Goal</th>
<th>Question/Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>X goal</td>
<td>Capabilities</td>
<td>Are the requirements fulfilled?</td>
</tr>
<tr>
<td>G goal</td>
<td>Efficiency</td>
<td>How efficient is the feature or configuration?</td>
</tr>
<tr>
<td>Q goal</td>
<td>Resource usage</td>
<td>How much memory needs the calculation?</td>
</tr>
</tbody>
</table>
Example: Comparing Configurations (Summation as Consolidation)

Consolidation

Configuration 1

- Customer Relationship Management: 300
- Sales: 200
- Marketing: 100

Configuration 2

- Customer Relationship Management: 500
- Sales: 200
- Service: 300

"Measurement" on feature

300 < 500
Consolidation Methods

- simple approach for X goals: binary (achieved/not achieved)
  ex: If there is one sub feature which does not achieve the X goal, the whole configuration does not achieve the goal.

- simple arithmetic operations for Q/G goals: e.g. summation

- complex consolidation method with dependencies, because not every feature set allows a simple summation of the quality values, e.g. in terms of memory consumption.
  If $F_A$ and $F_B$ → multiply sum with 0.5.
Further Measurement Approaches

- **Not every quality goal can be measured easily, e.g. safety properties!**

- Quality measurement at a **concrete software instance** respectively configuration
  - usage of Benchmarks
  - usage of model and code

- Quality determination by means of a **business approach**
  - Assigning „costs“ to each feature
  - Negotiation of „total costs“ for a configuration according to economical principles (discounts, price increase, ...)
Structural Similarity of Configurations

- **When does comparing configurations make sense?**
- Comparison of similar configurations as a lead for further investigations

Configurations are similar, if there are commonalities that can be identified. This does not necessarily refer to the selection of equal features. Similarity also involves **structural commonalities**. (derived definition)
Structural Similarity of Configurations

- **When does comparing configurations make sense?**
- Comparison of similar configurations as a lead for further investigations

Configurations are similar, if there are commonalities that can be identified. This does not necessarily refer to the selection of equal features. Similarity also involves structural commonalities. *(derived definition)*

- **same parent feature**
- **same group**

![Diagram showing structural similarity of configurations]
Structural Similarity of Configurations

- **When does comparing configurations make sense?**
- **Comparison of similar configurations as a lead for further investigations**

Configurations **are similar, if there are commonalities** that can be identified. This does not necessarily refer to the selection of equal features. Similarity also involves **structural commonalities**. *(derived definition)*

- **same parent feature**
  - **same group**
    - **same features of a group**

![Diagram showing structural similarity of configurations](image)
Structural Similarity of Configurations

- **When does comparing configurations make sense?**
- Comparison of similar configurations as a lead for further investigations

Configurations are similar, if there are commonalities that can be identified. This does not necessarily refer to the selection of equal features. Similarity also involves structural commonalities. *(derived definition)*

- same parent feature
  - same group
    - same features of a group
    - same features of a parent feature

---

**FOSD Meeting, 07.05.2014**

**Quality Assurance by Means of Feature Models**

13 of 21
Structural Similarity of Configurations

- **When does comparing configurations make sense?**
- Comparison of similar configurations as a lead for further investigations

Configurations are similar, if there are commonalities that can be identified. This does not necessarily refer to the selection of equal features. Similarity also involves structural commonalities. *(derived definition)*

- same parent feature
  - same group
    - same features of a group
    - same features of a parent feature
  - same attributes
Draw Conclusions (1)

Quality change due to **software evolution**

- Configuration before evolution step
  - Measure quality
  - Feature set comparison
  - Output: difference

- Configuration after evolution step
  - Measure quality
Draw Conclusions (2)

Quality difference of similar configurations

...and its change due to **software evolution**

Quality difference of similar configurations
Structural Analysis of Feature Models

- Comparisons only based on a feature model and existing derived configurations.
- Complete automation possible
- Conceptual implementation in Java as Eclipse plugin within the extFM-Tooling project (https://github.com/extFM/extFM-Tooling/)

Which features are selected in none of the given configurations?
In which and how many configurations is the feature selected?
Which information is relevant for the selection state of the feature?
3. Discussion

- Possible starting points for further research
  
  - Analysis of the shown methods without the SaaS context
  - Further research in attributed feature models
  - Extension of analysis tools
  - Integration of the prototypical implementation in a practically usable tool
  - Combination of attributed and structural analysis
  - Empirical investigation in practicability in real projects
Thank you for your attention!

1. Fundamentals
   - SaaS
   - Quality Assurance

2. Quality Assurance by Means of Feature Models
   - Quality as Attributes in Feature Models
   - Structural Analysis of Feature Models

3. Discussion

(References on next slides)
References (1)

- Illustrationen von www.freedigitalphotos.net


References (2)


References (3)


