Visualization of Feature Survival in Platform-Based Embedded System Development for Improved Understanding of Scope Dynamics

Krzysztof Wnuk, Björn Regnell – Lund University
Krzysztof.Wnuk@cs.lth.se, Bjorn.Regnell@cs.lth.se

Lena Karlsson – DNV ITGS Sweden
Lena.Karlsson@dnv.com
Case Study Motivation

• Our previous results revealed the problem of setting a good balanced scope of the platform projects

• Why?
  – decisions about new functionalities are made *a priori*
  – often changed by key customers or management

• Solution - Visualization support for improved understanding of scoping decisions
Case Study Introduction

• Empirical data from an industrial project at a large company (5000 employees) using a product line approach
• Develops embedded systems for a global market
• The company uses a stage-gate model for requirements projects
• Feature as a decision unit
Requirements Management Process

- Requirements Teams (RTs) and Design Teams (DTs)

**Roadmap**

1. **MS 1**
   - **High-level features (estimates)**

2. **MS 2**
   - Features are refined to req.

3. **MS 3**
   - Effort estimates refined. Design started

4. **MS 4**
   - Final scope ready requirements and design finished.

wnuk@cs.lth.se
Scoping Process

• A feature as a basic entity for scoping process
• The market-value and effort estimates are obtained using a cost-value approach
• The scope is decided based on the ROI in relation to the available development resources within the DTs
• Feature List (FL) document contains the scope
• The FL is updated and baselined each week after decision in the Change Control Board (CCB)
Methodology

STEP 1: extraction the scoping information
- value of scope parameter was exported for each baseline of Feature List document (result: 81 baselines between 300 and 600 features)

STEP 2: processing the data
- sampling policy as a result of validation of gathered data with requirements experts
- as a result 39 data points were considered as relevant

STEP 3: transformation of the data into a mathematical representation
- coding scheme: rows and columns
- value of the Scope attribute is mapped to integer value
- coloring scheme: green is in, red is out and gray is not yet decided
Feature Survival Chart
Feature Growth Chart

Feature Growth Chart

- In Primary or Secondary Flow
- Out-scoped
- Not decided
- 1 week on the timeline

Project Timeline

- MS 1 (20.08.07)
- MS 2 (30.11.07)
- MS 3 (21.02.08)
- MS 4 (15.05.08)

wnuk@cs.lth.se
Graphs per RTs
Validation

Three steps of validation

• **STEP 1:** Interviews with RTs to understand the challenges of setting a realistic scope early in the project (before generating the graphs).

• **STEP 2:** Performing visualization keeping in mind the issues and input received from the first step.

• **STEP 3:** Presenting the results to RTs and project management

Limitations

– A static two-dimensional figure
– Limited end-user’s configurability
– Size of features in terms of number of sub-requirements, their criticality or implementation cost
– Tightly coupled with the specific requirements and the requirements engineering practices of this particular case
Conclusions

• Increases awareness of balancing between setting limited scope early and setting a too large scope
• Helps to identify what features and what time frames to analyze in order to find scoping problems
• May be useful in visualizing instability of the scope setting process
• Empirically evaluated in a large industrial environment
• Can be reused to visualize other attributes
Next steps

- Improve interaction with the user
- Improve configurability
- Y axis scaling – show underlying system requirements.
- Introduce additional attributes, such as criticality and implementation cost
- Introduce grouping the features based on feature dependencies for visualizing simultaneous scoping or de-scoping of related features.
- Add statistical measures such as average time to de-scoping of a feature and the total effort spend on non-survivors.
- More empirical results from other companies.
Questions?