Evaluating The Use of Domain-Specific Modeling in Practice

Juha Kärnä, Polar Electro

Juha-Pekka Tolvanen, MetaCase

Steven Kelly, MetaCase

DSM'09
25 October 2009

Contents

- Motivation and research question

- Introduction to the domain

- Research method

- Results of the evaluation

- Summary

Motivation and research question

- How to measure that the created DSM solution provides the desired outcomes?

- Research method challenge

  - Many proper research methods are costly

  - Companies are not looking generalization or theory building – but have more modest objectives

- Companies need data for decision making, e.g. to finalize the DSM solution or to start new projects with DSM

Domain: Sport computer apps

Features include:
- Heart rate measuring, analysis and visualization
- Calorie calculation, like current, cumulative, expenditure rate, active time
- Speed: current, average, maximum
- Distance, based on interval, trip, recovery
- Altimeter, vertical speed, altitude alarms, slope counter, graphical trend
- Cycling information like pedaling rate and cycling power
- Barometer, pressure drop alarm, graphical trend
- Exercise diaries
- Sensor connectivity (heart rate, speed, cadence, power, GPS)
- Compass, Temperature, Odometer, Logbooks, etc.

About the product line

- Software development is constrained by limited resources:
  - memory, processor speed and battery life
- Polar focused on UI application development
  - Single largest piece of software
    - Takes 40-50% of the development time
  - Typically always vary among products
  - Results of domain analysis showed
    - 70% of the UI application code easy to automate
    - 25% could be probably be handled as well
    - 5% of the UI special cases not easy to tackle with DSM

Requirements for the DSM solution

- Polar created the needed languages and generators internally
- Key requirements for the DSM solution were:
  1. Improve the productivity of UI app development
  2. Improve the quality and maintainability of the code
  3. Reduce the manual work needed to copy data from specifications into code
  4. Make the introduction of new developers easier
  5. Be usable for experienced and novice developers
- Evaluation focused on analyzing how these requirements are satisfied

Sample design of UI application
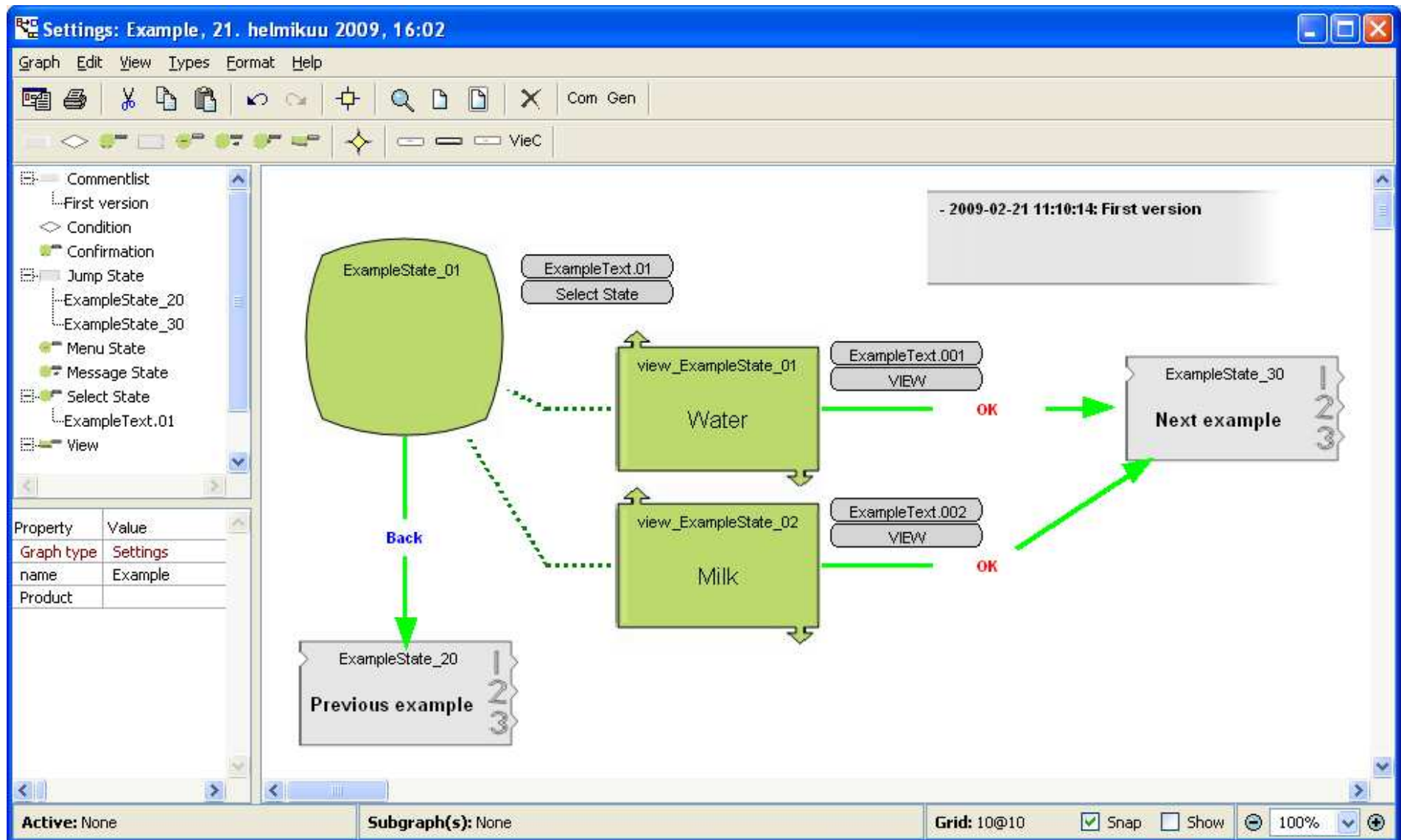
Evaluation methods

- Seeking balance between evaluation costs and getting creditable and repeatable results

- Two research methods
  - Laboratory study
    - Implement a small, typical feature
    - 6 current developers, 6 implementations
  - Pilot project
    - Implement large portion (64%) of a whole product
    - 1 person

- Compare the use of DSM and the current practice

Phases of laboratory study

1. Training
   - Introduced language and tool (1h)
2. Conducting laboratory experiment
   - Started with a common UI specification
   - Implemented between 75 to 125 minutes (mean 105 minutes)
3. Evaluating the correctness of the results
   - Answering possible questions during laboratory study
   - Ensuring that feature was implemented as expected
4. Reporting experiences
   - Collecting developers opinions on the DSM solution and on the current practice

Measuring productivity

- Comparing the use of DSM and currently used approach
    - Developing the same type of applications
    - Asking opinions and measuring development time
- Results:
    - All developers agree similarly on productivity
    - Calculated productivity improvement: at least 750%

Perceived productivity
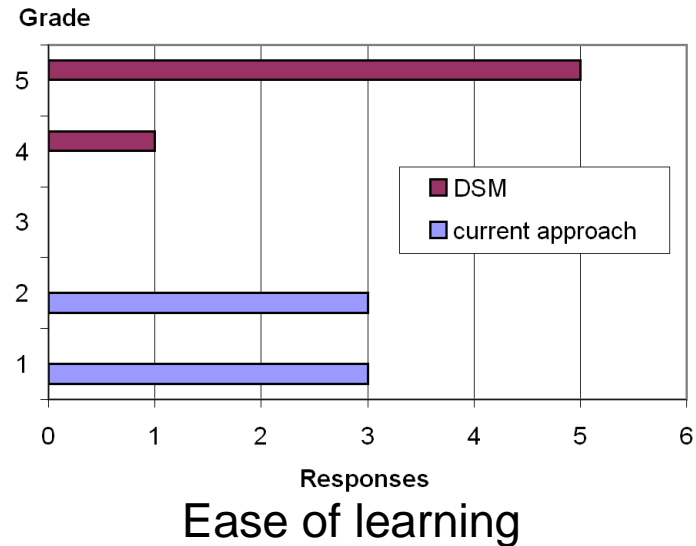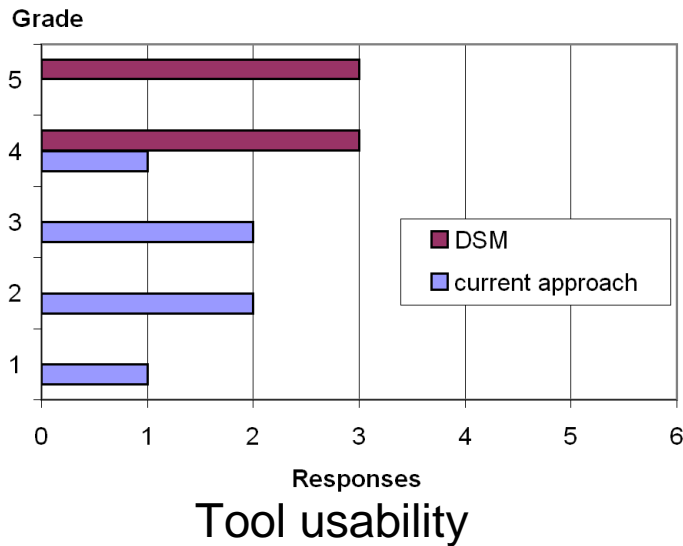(scale 1-5, 5=best productivity)
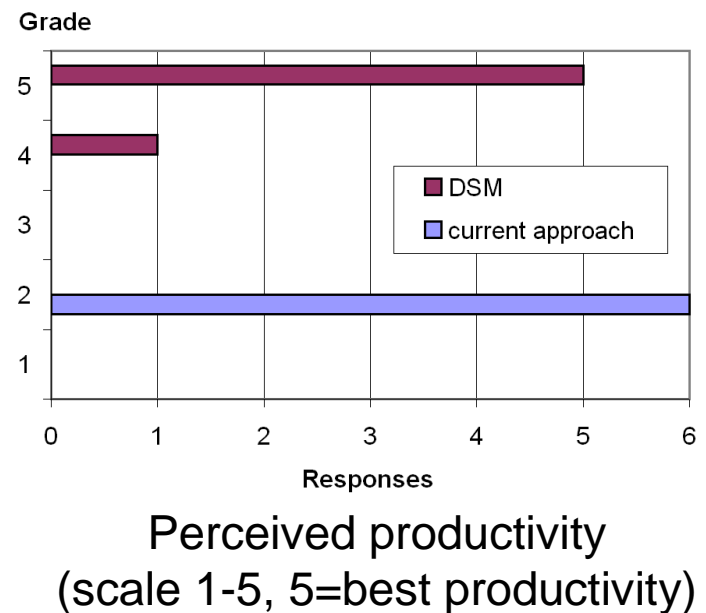
Evaluating quality of product and its development process

- Process quality
  - How well the development approach prevents errors
- Quality of the product
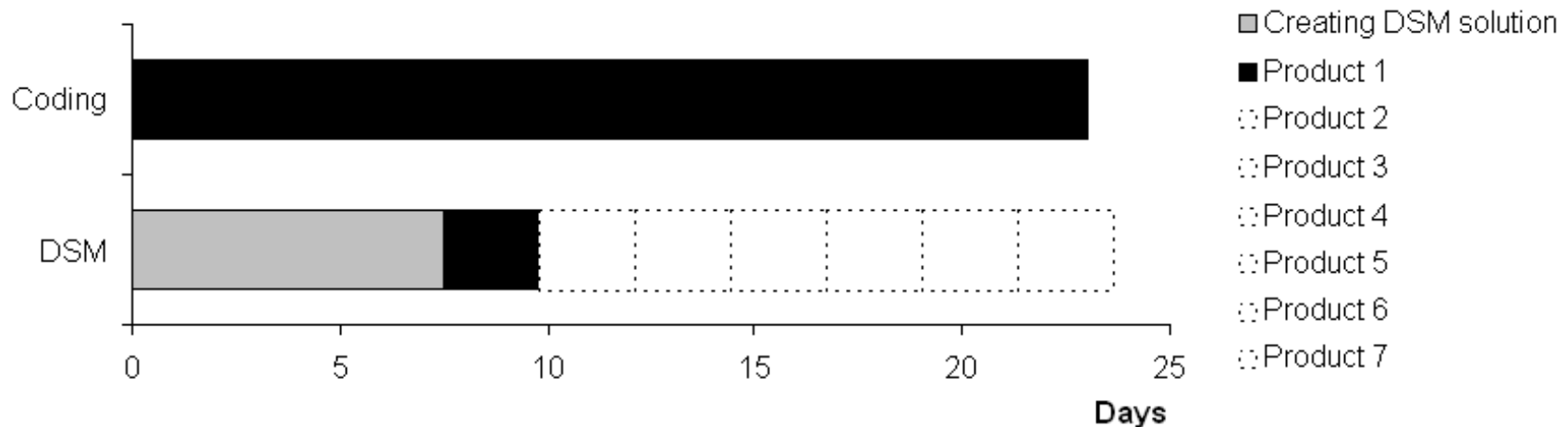 - Code quality
- Results:

Error prevention

Code quality
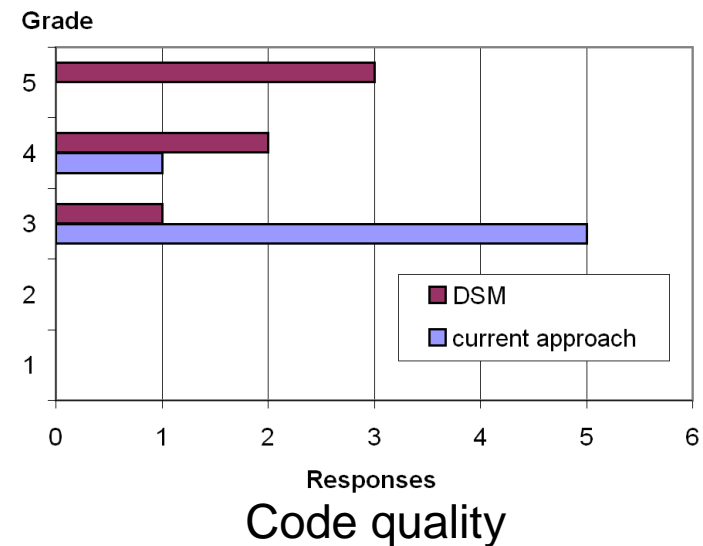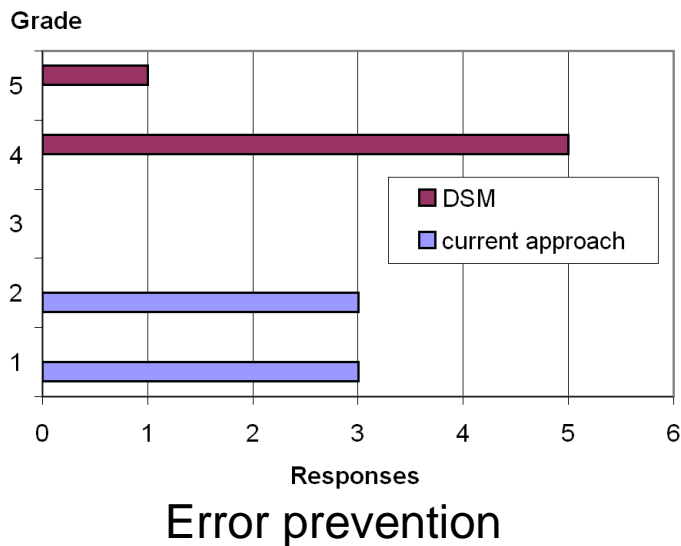
Usability and learning

- Usability
  - How usable are the tools
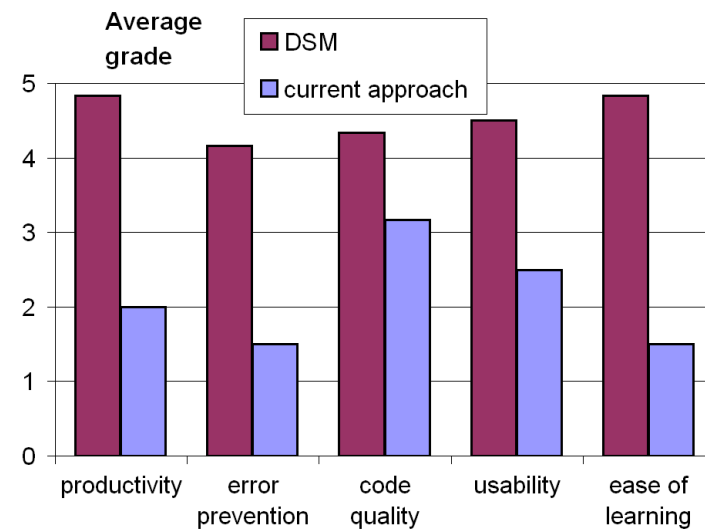- Learning
  - How easy it is to learn and use
- Results:

Tool usability

Ease of learning

Return on investment

- Theoretical model (=DSM is not changing)

Legend:
- Creating DSM solution
- Product 1
- Product 2
- Product 3
- Product 4
- Product 5
- Product 6
- Product 7

- DSM solution developed in 60 hours
- Product development with:
  - the current approach: 23 days
  - DSM: 2,3 days

Summary

- A practical approach to evaluate DSM proposed

- The approach used to study a particular DSM solution with controlled test and pilot project.

- Results show:
  - Better productivity
  - Improved quality
  - Development easier to learn and use

- Can be extended to cover other development phases, like specification, testing, localization

Thank you!

- Questions, please?