

Requirements Engineering II

Practice and Limitations of Formal Methods

Stefan Hutter & Sebastian Müller

14.12.2009

Agenda

- Thoughts about Formal Methods by Daniel M. Berry
 - Definitions of FMs
 - Economics of FMs
 - Problems using FMs
 - Benefits from using FMs
- Formal Methods in Practice – Successful Technology Transfer at NASA
 - Problems of Technology Transfer
 - The Flight Code Generator
 - How became the FCG Project a Success?

Definitions of FMs

- „An FM is any attempt to use mathematics in the development of a SWICBS, in order to improve the quality of the resulting SWICBS.“ (Berry)

[SWICBS = Software-Intensive Computer-Based Systems]

Goal:

- Verification
- Mathematical study of key problems
- Refutation

Economics of FMs

- Explains, why FMs are not used very often in practice:
 - Doing FMs is very **cost-intensive**: FMs drive the cost up from 2 to 5 fold
 - One can do the software twice (which extremely increases the quality of the second solution) at the cost of about 100%
- ➔ For most software it is **just not worth**

Problems using FMs

- Principle of **preservation of difficulty** by Sudarsky
 - Difficulties caused by **lack of understanding the real world domain** can't be eliminated by using FMs
 - FMs make the problem **even worse**: the misunderstanding gets formalized into the specification

Why using FMs can be successful

- **Second time phenomenon** by Wing and Berry
 - „If you write a formal specification and then you write code, you have done the problem formally two times.“
 - The **code will be better** than if you had not done the formal specification
- **Hawthorne effect**
 - If the use of FM is experimental and the participants observed they will try harder even if he or she is not the FMs expert

Problems of Technology Transfer

theoreticians

- Solve intellectually interesting problems without regard to actual applications
- Build tools in isolation without consideration of the work process of software production
- Don't maintain and support their tools

vs.

practitioners

- Don't like the obscure notation, given toy examples of FM which don't scale
- Are concerned about schedule risk
- Are skeptical and want to inspect and understand the generated code

Flight Code Generator



Automatic code generation experiment for a new satellite program at NASA in 1995

Task: generate a key module from specification to achieve a different functionality.

Problems: None of the lab's researchers had any experience with the satellite domain. Solved by getting an aerospace engineer allocated to the project.

Results: Experiment was successful but was rejected as the costs didn't fit the within the parameters considered reasonable by NASA.

How does FCG works?

Input: Specifications of the control laws in MatLab.

Manually add necessary information as comments in the MatLab code so they can still be processed.

FCG: Batch tool written in Lisp

Output: Database records that describe telemetry and command information, Ada package that conforms to interfaces and conventions of the reusable architecture, test environment that allow interactive or scripted unit testing, documentation.

Why is FCG a Success?

- Specialized to a narrow domain
- Generates Code that fits into an existing architecture
- Fits into an established development process
- It's always possible to revert to the old ways
- Human-readable code and a shallow learning curve

