

Formal Methods in Practice

Requirements Engineering II



Economics of Formal Methods

- Main Formal Method Groups
 - Verification
 - Intensive study of key problems
 - Refutation
- Errors and Requirements Specifications
- Second time phenomenon
- Qualities of people



Verification

- Code proved consistent with a formal specification
- (Semi-)Automated verification of code
- Completeness as measure of amount of verification
 - **Verification Levels**
 1. Formal specification of requirements
 2. Verification of requirements specification
 3. Formal specification of design
 4. Verification of design specification
 5. Formal specification of code
 6. Verification of consistency of design and code specifications
 7. Code
 8. Verification of 6 + code



Verification (2)

- Problem of FM for all code
 - Increase in development time
 - 200% over normal systematic methods for basic evaluation
 - 1000% over normal systematic methods for only verification of requirements and design, then coding (levels 1 – 4, 7)



Intensive Study of Key Problems

- Verification not feasible in most situations
- Only focus on difficult / problematic parts of the system
- Perform verification there
- Problems
 - Danger of overlooking parts of the system
 - Proving that no parts were overlooked negates benefits
- ... but with experience it's possible to counter problems



Refutation

- Proving that a system works can be done in 2 ways
 - Show that it works by testing everything
 - Trying to show that it doesn't work
- Refutation only requires finding one mistake to prove system does not work
- 2 types of refutation
 - Computable properties of specification
 - Execution of code based on specification and test data
- Not guaranteed to reveal errors



... actually there is a 4th type of FM

- Programming itself
- Sometimes more economic to write code twice than to validate
- Problem
 - Errors become more expensive with progress
- Cost / Benefit must be better



Errors and requirement specifications

- Most errors arise early in project lifetime
 - 65% - 85% in discovery, specification and documentation stage
 - Coding stage only 15% - 35% of errors
- Coding errors are the easiest to fix
- Emphasis is being made on code verification
- Economically more feasible to focus formal methods on requirements
- BUT: Using FM does not fix gaps in understanding



Why FM work:

The Second Time Phenomenon

- When doing something the 2nd time, we do it better
- First time: writing formal method
 - Not detailed enough to write informal specification
- Second time: writing source code
 - Implementation errors will remain
 - Logical errors can be found better 2nd time around



Qualities of People

- FM require absence of assumptions
- Formal Methods can turn out more accurate if author knows nothing of the problem domain
- Lack of assumptions can
 - Force to build up assumptions and discover the problem
 - Regard slight errors more rigorously
 - Foster asking questions to rest of the team
- Good results achieved when using one person who knows nothing and one domain expert in team



Conclusion

- FM do work, but require careful consideration
- High cost of using FM is the reason it is often not used in practice
- Coding as a 4th type of FM often abused
 - Short term cost savings
 - Lazy developers
 - Potential cost hazard!