

Formal vs. Informal Requirement Specifications



Nate Kuehner
Toronto Design Centre



© Copyright 2003 Motorola Inc. All rights reserved.

Why don't we use formal specs?

- Motorola uses natural language requirement specs (almost 100%)
- Why? Economics. How much saved for how much spent?
(Why are most course programming assignments given in natural language?)
- How much value does a second formalization add?
 - It's not a question of *whether* to formalize, but *once or twice*.
 - Value of second formalization depends on many factors (next slide)
- Usually our biggest problem is *missed* requirements.
 - Many formal spec examples are in mature domains where requirements are well-known and the goal is simply to replace hardware with software.
- Other economical ways to cope with ambiguity, inconsistency.
 - Maintain a shared interpretative context (domain expertise)
 - Write the program



© Copyright 2003 Motorola Inc. All rights reserved.

Which parts of which specification methods are applicable to which parts of which projects?

Project Characteristics

- Criticality – consequences of errors
- Cost of detecting, correcting errors
- Application Domain, software type
- Scale, Complexity
- Product Lifespan, Time to Market
- Domain expertise of software team

Method Characteristics

- Degree of Formality
- Features, Capabilities
 - Proof, Simulation, Refinement, Test Case Generation, etc.
- Expressiveness (domain fit, degree of abstraction power)
- Maturity, Industry Acceptance
- Tool Support
- Usability



© Copyright 2003 Motorola Inc. All rights reserved.

But we're eager for good news!

- We don't enjoy the pain of finding and fixing software errors under pressure.
- We are very eager to understand the answers to the previous question, and to explore new possibilities...
- **Help us understand which methods are economical for which problems!**



© Copyright 2003 Motorola Inc. All rights reserved.