The Future of Software Engineering

Mats Heimdahl
Professor, Computer Science and Engineering
Director, University of Minnesota Software Engineering Center

What Do YOU Think?

• Present challenges?
• Major breakthroughs the last 5-10 years?
• Future challenges?

Present

• Not much new
  – All familiar problems still there
• Some “new stuff”
  – The “Cloud”
  – Service oriented
  – Outsourcing
  – “Agile” movement
The (bleak) Future

- Ignorance will still be rampant
- Security/Privacy will be key
  - But will be unachievable
- Replace people with automation
  (not cheaper people)
  - Static Analysis

Widespread Ignorance

“We are preparing to submit our new [critical device with various real-time constraints] for certification. Our software has been developed using Visual Basic under Windows, do you have any advice on how to best prepare our approval package?”

More…
High Profile Failures

How Can We **NOT** Know?

1. Software development is difficult, time consuming, and costly
   And, operation and maintenance is costly

2. If you do not plan for 1, you will be behind schedule, over budget, and possibly fail

Engineering Successes?

No More Heroes

We Need More Craftspeople

Courtesy John Knight, University of Virginia
Achieving Safety/Security/Quality

- Apply well known systems engineering techniques
- Identify the software requirements
  - In particular the safety/security/quality requirements
- Rigorously apply well known software engineering techniques to ensure software safety/security/quality requirements met

- The End
Development Driven By Models

Platform Independent Model (PIM)
Platform Specific Model (PSM)

Requirements
Implementation

Static Analysis Techniques

Even Small Systems Have Trillions (of Trillions) of Possible Tests!

Testing Checks Only the Values We Select

Model Checker Tries Every Possible Value!

Discussion
Ignorance, An Example: Process

- CMM and CMMI
  - Interpreted as heavy handed processes
  - Often are
- “Agile”
  - Interpreted as “anything goes”
  - Excuse and cover for *&amp;&amp;#$#@ practices

**Constant over-correction**