Perspectives on Software Assurance

Allen P. Nikora
Autonomy and Control Section
Jet Propulsion Laboratory
California Institute of Technology

OSMA Software Assurance Symposium
Sept 4-6, 2002

The work described in this presentation was carried out at the Jet Propulsion Laboratory, California Institute of Technology. This work is sponsored by the National Aeronautics and Space Administration’s Office of Safety and Mission Assurance under the NASA Software Program led by the NASA Software IV&V Facility. This activity is managed locally at JPL through the Assurance Technology Program Office (ATPO).
Agenda

• Why Software Assurance
• Current Practice
• What Else Can Be Done Right Now?  
  − .. And Why Isn’t It Being Done?
• How Does Research Relate to Practice?
• What Needs to Be Done?
Why Software Assurance?

- A second (and third, and fourth) pair of skilled eyes will find problems the first pair misses.
- Developers tend to be blind to their own mistakes.
- Separate management chain reduces tendency to hide problems
  - ..but management chains converge somewhere
Current Practice

• Generally viewed as separate organization
• SA engineers generally don’t perform development tasks
• Selective evaluation of technical products
  – Limited application of advanced analysis techniques
  – Some collaboration with research tasks
• May be more emphasis on process
  – Maximize use of limited resources
• Separate reporting channel
• Varying levels of interaction with developers
• Varying levels of commitment
What Else Can Be Done Right Now (and Why Is It Not Being Done)?

- Retaining Use of Best Practices
  - An organization may start using an SA method, demonstrate benefits, yet not retain its use over the long term
- Measurement
  - More measurement activity seems to be concerned with estimating/forecasting cost and schedule
  - Quality measurements reflect current status
    - Cumulative number of problems/issues arising during reviews
    - Cumulative failures observed during test vs. time
  - Often not tied to predictive quality models
    - SRE
    - Fault models
What Else Can Be Done Right Now (and Why Is It Not Being Done)?

- Formal Specification/Analytical Verification
  - "Semi-formal" methods currently in use
    - Statecharts
    - UML diagrams
    - Test case generators
  - "Industrial strength" tools currently available (e.g., SPIN, SCR, PVS, ...), not as frequently used
    - Require significant training effort
    - Often not covered in undergraduate CS/Software Engineering curriculum
    - On-the-job training takes away from development time
How Does Research Relate to Practice?

- Practical problems drive research
  - Perhaps more so than in other fields
- Development efforts often supply data needed by researchers
  - Demonstrate mutual benefit
  - Non-intrusive data collection
  - Sensitivity of failure/fault data – perhaps even more so than cost/schedule data
- Infusion rate of research results into practice is low
  - Requires more commitment from practitioners, researchers
  - Practitioners reluctant to try techniques not proven in previous efforts
  - Cost pressure in projects to avoid experimentation
How Does Research Relate to Practice (cont’d)?

- Research results are validated to varying degrees
  - Varying availability of data
  - Development techniques change quickly enough that results may no longer apply
What Needs to Be Done?

- **Education**
  - Extend SW Engineering curriculum at undergraduate level
  - Introduce SA as element in SW Engineering curriculum at undergraduate levels
    - Take advantage of practitioner’s experience
  - Encourage practitioners to take additional training in SA
    - Tuition reimbursement programs, bonuses/rewards
    - Rotating assignment of personnel to SA organization for specified period of time
      - Internal apprenticeship
      - Promotion path
What Needs to Be Done (cont’d)?

• Measurement
  – Establish institutional set of measurement goals
  – Establish standardized measurement mechanisms at institutional level
  – Establish and maintain baseline set of failure/fault data
  – On-going analysis of baseline failure/fault data
    • Project application
    • Exploratory