Software Quality Assurance

David Janzen

What is quality?

- Crosby: “Conformance to requirements”
  - Issues:
    - who establishes requirements?
    - implicit requirements
- Juran: “Fitness for intended use”
  - Issues:
    - Who defines fitness? Novice users, experts, engineers?
- IEEE: “The degree to which the software possesses a desired combination of attributes”
  - Possible attributes:
    - usability, features, performance, 0 defects, low cost, elegant code, …

Quality Evolution 1

- Quality Control
  - Measure quality *after* system is built
  - Typical practices:
    - Testing, inspections, metrics at end of construction
    - E.g. # requirements met, # tests passed, coupling
  - Problems:
    - Have we tested enough?
    - Defect fixes inject new defects
    - Result in adversarial relationships

Quality Evolution 2

- Quality Assurance
  - IEEE: “A planned and systematic pattern of all actions necessary to provide adequate confidence that the product conforms to established technical requirements”
  - Typical practices:
    - Inspections, reviews, audits, metrics, communication throughout development process
    - SQA Plan (see examples on web)
  - Problems:
    - QA skills are rare
    - Separate QA team: communication issues, disputes
    - Commitment to QA wanes under schedule pressure

Quality Evolution 3

- Quality Engineering
  - Build quality as part of the SE process
  - Typical practices:
    - Everyone considers quality part of their job
    - Finding defects is good
    - QA team coaches/mentors, not evaluators
    - Fact-based decision-making based on metrics
  - Problems:
    - Process and cultural change

Quality is Free

- Crosby: “Quality is free. But it is not a gift.”
  - Prevent defects rather than remove them
  - “Zero-Defect is the attitude of defect prevention. It means, 'do the job right the first time.‘”