

# **Software Quality Assurance**

## **— Meeting New Challenges**

Jeff Tian (tian@engr.smu.edu)  
Southern Methodist University  
Dallas, Texas, USA

### **Contents**

- Software Quality: Why & What?
- Concerted QA Activities/Effort
- Testing and Reliability Engineering
- Measurement and Risk Management

---

## Software Quality: Why?

---

- Software in modern society:
  - ▷ widespread/ubiquitous applications
  - ▷ society/people's reliance on it
  - ▷ need software quality assurance (QA)
  
- New challenges:
  - ▷ diverse environment and functionality
  - ▷ large size and complexity
  - ▷ massive user population
  
- How to meet the new challenges:
  - ▷ multiple QA alternatives
  - ▷ selective/usage-based testing and reliability engineering
  - ▷ measurement and risk management

---

## Software Quality: What?

---

- Goal: Deliver software system that...
  - ▷ does what it is supposed to do.
  - ▷ does the things correctly.
  - ▷ show/demonstrate/prove it ("does").
- Quality views and attributes:

View	Attribute	
	Correctness	Effectiveness
Customer (external)	Failures: reliability safety	Usability Maintainability Portability Performance Installability Readability
Developer (internal)	Faults: count distr class	Design Size Change Complexity

---

## Defect and Quality

---

- Defect/bug definition
  - ▷ Needed to defining quality.
  - ▷ Failure: external behavior
    - deviation from expected behavior
  - ▷ Fault: internal characteristics
    - cause for failures
  - ▷ Error: incorrect/missing human action
    - conceptual mistakes
  - ▷ Relations (not necessarily 1-1)
    - errors  $\Rightarrow$  faults  $\Rightarrow$  failures
  
- Defect measurement & analyses
  - ▷ data: where/when/type/etc.
  - ▷ goal: assessment/prediction/improvement
  - ▷ defect analysis and quality models:
    - causal/risk/reliability analyses

---

## QA1: Combined Activities

---

- Preventing fault injection
  - ▷ Causal/statistical/etc. analyses based
  - ▷ Preventive measures:
    - education, technology, process, tools
  - ▷ Formal verification (faults absent)
  
- Removal of faults
  - ▷ Inspection: faults discovered
  - ▷ Testing: failures  $\Rightarrow$  faults
  
- Tolerance of faults
  - ▷ Local failure  $\neq \Rightarrow$  global failure
  - ▷ Dynamic measures to tolerant faults

---

## QA2: Selective Testing

---

- Testing as part of QA:
  - ▷ Activities focus on testing phase
  - ▷ Related act. throughout dev. process
  - ▷ One of the most important part of QA
  - ▷ What: black vs. white box
  - ▷ When to stop: coverage vs. usage-based
  
- Usage-based random/statistical testing
  - ▷ Meet the software QA challenges
  - ▷ Actual usage and scenarios/information
  - ▷ Captured in operational profiles (OPs)
  - ▷ Simulated in testing environment
  - ▷ Termination criteria: reliability goals

---

## QA2: Testing and Reliability

---

- Usage based testing to ensure reliability.
  
- *Reliability*: Probability of failure-free operation for a specific time period or a given set of input under a specific environment
  - ▷ Reliability: customer view of quality
  - ▷ Probability: statistical modeling
  - ▷ Time/input/environment: OP
  
- OP: Operational Profile
  - ▷ Quantitative characterization of the way a (software) system will be used.
  - ▷ Generate/select/execute test cases (for usage-based statistical testing)
  - ▷ Realistic assessment/predictions
  - ▷ Development decisions/priorities

---

## QA2: OP or Usage Models

---

- Types of OP:
  - ▷ Musa 1: flat (tabular) OP
  - ▷ Musa 2: multi-stage/attribute
  - ▷ Markov chains or Markov OP
  
- Markov OP: Basic ideas
  - ▷ Markov chain for usage information
  - ▷ State: operations/functions
  - ▷ Transition: probabilistic
    - reflects usage sequence/frequency
    - history independent (Markovian)
    - but reflects local usage info.

---

## QA2: Case Study in Web Testing

---

- Web testing factors:
  - ▷ Web size, complexity, user focus
  - ▷ Focus on source failures
  - ▷ Statistical web testing
    - modeling, testing, result analysis
  
- Measurement and analysis:
  - ▷ Model construction: access-log
  - ▷ Model usage in testing:
    - usage frequency/probability
    - coverage and criticality
    - hierarchy and dynamic expansion
  - ▷ Reliability analysis: error-log

---

## QA2: SRE

---

- SRE: software reliability engineering.
  
- Reliability: Prob(failure-free operations)
  - ▷ *Time domain*: for a specific period.  
⇒ Reliability growth models.
  - ▷ *Input domain*: for a specific input set.  
⇒ Repeated sampling models.
  - ▷ Statistical testing based on customer operational profiles (OP)
  
- A new integrated approach:
  - ▷ Tree-based reliability models (TBRMs).
  - ▷ Both input/time domain information.
  - ▷ Risk focusing and remedial actions.
  - ▷ Main info. source: (Tian 1998)

## QA2: TBRM

---

- Combine time/input domain models  
⇒ tree-based rel. models (TBRMs)
  
- Reliability assessment/prediction:
  - ▷ Change in predicted response.
  - ▷ Through tree structural change.
  
- Risk identification and remedies:
  - ▷ Identify high risk input state.
  - ▷ Additional analysis.
  - ▷ Enhanced test cases.
  - ▷ Remedies for components.
  
- Cross validation: IBM data/results.

## QA3: S/w Measurement & Analysis

---

- Traditional measurements:
  - ▷ Direct measurement:
    - quality and defects
    - also cost, schedule, effort etc.
    - objective to be controlled/optimized
  - ▷ Indirect product measurement:
    - size and volume
    - complexity: control/data/presentation
    - change, dependency, etc.
    - at product/module/component levels
    - used to affect outcome above
  
- Traditional analyses:
  - ▷ Correlation analysis.
  - ▷ Multiple regression.

---

## QA3: Major Issues

---

- Issue: Measure evaluation/selection
  - ▷ Weyuker's desirable properties
  - ▷ Tian-Zelkowitz axioms & classification
  - ▷ Selection as constrained optimization
  - ▷ Use with CTA for effort prediction
  - ▷ Application to NASA/SEL
  
- Issue: Risk identification
  - ▷ Risk: (high) probability of undesirable situations or consequences
  - ▷ 80:20 rule: 80% of problems traceable to 20% of components
  - ▷ Need to identify high risk modules
  - ▷ Characterization of these modules
  - ▷ Lead to corrective/remedial actions

---

## QA3: Risk Identification, How?

---

- Traditional techniques:
  - ▷ Correlation analysis.
  - ▷ Multiple regression.
  - ▷ Examples and problems.
  
- New techniques:
  - ▷ Principal component analysis (PCA) and discriminant analysis
  - ▷ Neural networks
  - ▷ Learning algorithms
  - ▷ Tree-based modeling
  - ▷ Pattern matching approaches
  - ▷ Examples and comparison
  
- Survey paper: Tian, SQP 2(2), 3/2000

---

## QA3: TBDMs

---

- Risk identification:
  - ▷ Assumption in traditional techniques:
    - linear relation
    - uniformly valid result
  - ▷ Reality of defect distribution:
    - isolated pocket
    - metric types & qualitative differences
    - correlation/dependency in metrics
  
- Risk characterization:
  - ▷ Identified, then what?
  - ▷ Result interpretation & extrapolation
  - ▷ Remedial/corrective actions
  
- TBDMs (tree-based defect models) for both risk identification and characterization:
  - Application in IBM and Nortel Networks

---

## QA: An Integrated Approach

---

- QA: Concerted effort with focus on selective, usage-based, statistical testing
  
- Measurement and analysis:
  - ▷ Software measurement and analysis
    - early: coding and some design
    - but internal focus
  - ▷ Software reliability engineering
    - customer quality perspective
    - but late in testing
    - although my TBRMs push it forward
  
- Integrated/expanded analysis:
  - ▷ Extended development phases
  - ▷ Early feedback and improvement
  - ▷ Usage of multiple info sources

## Generalized Technique: TBQMs

---

- Measurements and TBMs
  - ▷ Metrics selection
  - ▷ Direct application of TBQMs in testing
  - ▷ Generalized TBQMs  
(tree-based quality models)
  
- TBQMs using all information:
  - ▷ Defects from testing/inspection/etc.
  - ▷ Inspection/testing details
  - ▷ Analyzer for design/code/etc.
  - ▷ System monitoring devices
  - ▷ Root/usage approach to gather data

---

## Summary and Perspectives

---

- Availability vs. needs
  - ▷ Need software quality assurance
    - new challenges
  - ▷ Previously available:
    - individual QA techniques
    - software reliability engineering
    - software measurement & analysis
  - ▷ Available through this research:
    - integrated approach (3 components)
    - effective and wide applicability
  
- Future work:
  - ▷ New/better techniques
  - ▷ Deployment and technology transfer
  - ▷ Technique/tool refinement in response to practical problems