Software Reliability — An Integrated Approach

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

Contents

- Software Measurement and Analysis
- Software Testing and Reliability
- An Integrated Approach
- Conclusions and Perspectives

Overview: Project Background

- Background information:
  - ▷ S/w measurement at U. Maryland & NASA/SEL (1989+)
  - ▷ S/w reliability engineering at IBM (1992+)
  - ▷ Leading to current research at SMU (1995+)

- Ongoing Project at SMU:
  - ▷ Reliability: early measurement/improvement
    - – an integrated lifecycle approach
    - – NSF/CAREER award CCR-9733588
  - ▷ Telecom. S/w Testing & Reliability
    - – THECB/ATP award 003613-0030-1999
  - ▷ Collaboration with IBM and Nortel

An Integrated Approach

- 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

- Integration:
  - ▷ Extended development phases
  - ▷ Early feedback and improvement
  - ▷ Usage of multiple info sources

Software Measurement and Analysis

- Traditional measurements:
  - ▷ Direct measurement:
    - – quality and defects
    - – 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
  - ▷ Other measurements: PPP, env., etc.

- Traditional analyses:
  - ▷ Correlation analysis.
  - ▷ Multiple regression.

Software Measurement and Analysis

- 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

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

TBDMs: Why?

- Risk identification:
  - ▷ Assumption in traditional techniques:
    - linear relation
    - uniformly valid result
  - ▷ Reality of defect distribution:
    - isolated pocket
    - different types of metrics
    - correlation/dependency in metrics
    - qualitative differences

- Risk characterization:
  - ▷ Identified, then what?
  - ▷ Result interpretation.
  - ▷ Remedial/corrective actions.
  - ▷ Extrapolation to new product/release.

- TBDMs (tree-based defect models) for both risk identification and characterization.

TBDMs: Technique

- TBDMs: tree-based defect models.

- Technique: tree-based modeling
  - ▷ Tree: nodes=data-set, edges=decision.
  - ▷ Data attributes:
    - – 1 response & n predictor variables.
  - ▷ Construction: recursive partitioning.
  - ▷ Usage: relating response to predictors
    - – $Y = Tree(X_1, \ldots, X_n)$
    - – understanding vs. predicting
    - – identification and characterization
  - ▷ Works for mixed-types of data.
  - ▷ Tree growing and pruning.

- Many unique advantages:
  See Tian paper in SQP 2(2), 3/2000

TBDM Results

- IBM and Nortel products:
  - ▷ `DF`: Defect fixes per module
  - ▷ IBM-LS: a legacy system
  - ▷ IBM-NS: a new system
  - ▷ Nortel Networks: NT-X
  - ▷ All large software systems
- Results for IBM LS and NS:
  - ▷ LS: change, size, data complexity
  - ▷ NS: design and control complexity
  - ▷ Problem-prone modules identified
- Comparison: NT-X similar to IBM LS
  - ▷ Common traits of legacy systems
  - ▷ Implications: similar initiatives

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 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)

Existing Reliability Models

- Modeling reliability-fault relations through statistical techniques/models

- Time domain approach
  - ▷ SRGMs: S/w reliability growth models.
  - ▷ Reliability growth over time due to testing and defect removal.
  - ▷ Time-between-failure (TBF) SRGMs
    – r.v.: failure interval
  - ▷ Period-failure-count (PFC) SRGMs
    – r.v.: failure count for given interval

- Input domain approach
  - ▷ IDRMs: Input domain rel. models.
  - ▷ Reliability input (sub-)domain relations.

Existing Techniques: SRGMs

- Common assumptions:
  - ▷ Failure arrivals: stochastic process:
  - ▷ Environment: operational profiles.
  - ▷ Randomized testing.
  - ▷ Equivalence of time.
  - ▷ Homogeneous distribution of faults.

- Model output and usage:
  - ▷ *Assessment:* MTBF and failure rate.
  - ▷ *Prediction:* future reliability.
  - ▷ *Control:* exit criteria.

- Experience: use activity-based measurement

Input Domain Reliability Analysis

- A typical IDRM: Nelson model
  - ▷ Running for a sample of n inputs.
  - ▷ Randomly selected from set E:

$$E = \{E_i : i = 1, 2, \ldots, N\}$$

  - ▷ Sampling probability vector:

$$\{P_i : i = 1, 2, \ldots, N\}$$

  - ▷ $\{P_i\}$: Operational profile.
  - ▷ Number of failures: n_e.
  - ▷ Estimated reliability:

$$R = 1 - \frac{n_e}{n}$$

- Use of IDRMs:
  - ▷ Repeated sampling without fixing.
  - ▷ Restrict to specific input states ⇒ possibility for risk identification

An Assessment of Approaches

- Time domain reliability analysis:
  - ▷ Customer perspective.
  - ▷ Overall assessment and prediction.
  - ▷ Ability to track reliability change.
  - ▷ Issues: assumption validity.
  - ▷ Problem: How to improve reliability?

- Input domain reliability analysis:
  - ▷ Explicit operational profile.
  - ▷ Better input state definition.
  - ▷ Hard to handle change/evolution.
  - ▷ Issues: sampling and practicality.
  - ▷ Problem: Realistic reliability assessment?

An Integrated Approach

- Combine strengths of the two ⇒ tree-based rel. models (TBRMs)

- Using TBRM for individual modeling:
  - ▷ Input state: categorical information.
  - ▷ Each run as a data point.
  - ▷ Time cutoff for partitions.
  - ▷ Data sensitive partitioning ⇒ Nelson models for subsets.

- Integrated reliability analyses:
  - ▷ TBRM: partitioned subset reliability.
  - ▷ Use both input and timing information.
  - ▷ Monitoring changes in trees.
  - ▷ Enhanced exit criteria.
  - ▷ SRGM: overall reliability near exit.
  - ▷ Integrate into the testing process.

Using TBRMs

- Interpretation of trees:
  - ▷ Predicted response: success rate. (Nelson reliability estimate.)
  - ▷ Time predictor: reliability change.
  - ▷ State predictor: risk identification.

- Monitoring reliability change:
  - ▷ 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.

TBRMs: Cross Validation

- Consistency with macro models:
  $\Rightarrow$ Effects on cost, schedule, quality.

- Validate with reliability growth models:
  - ▷ Trend of reliability growth.
  - ▷ Stability of failure arrivals.
  - ▷ Estimated reliability.
  - ▷ Product purity level at exit:
    – 90% with TBRMs vs. other 30~70%.

- Process changes & improvements:
  - ▷ Failure detection and fault removal.
  - ▷ Long term effect on development.

- Ultimate test: in-field problems.

Generalized Technique: TBQMs

- Measurements and TBMs
  - ▷ Metrics selection
  - ▷ Direct application of TBRMs 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

Technique: Component Data

- Component data in early phases
- Root approach:
  - ▷ Each component/action as a data point
  - ▷ Refined granularity
  - ▷ Direct measurement and analysis
  - ▷ Issue: mapping to reliability
- Usage approach:
  - ▷ Usage sensitive measurement results
  - ▷ More front end computation
  - ▷ Product/customer view of quality
  - ▷ Issue: usage-component mapping

Lifecycle Connections

- Analysis technique
  - ▷ Binary to general TBMs
  - ▷ Other analysis techniques
  - ▷ Use existing partial results
    - – e.g. from Nortel/EMERALD
- Followup actions
  - ▷ TBQM-guided causal analysis
  - ▷ Preventive measures
- Other quality assurance activities
  - ▷ TBQMs as overall guide
  - ▷ TBQMs as info. consumer/producer

Tools

- Data capturing tools:
  - ▷ Logs: pre-existing
  - ▷ Analyzers: existing metrics tools and new special purpose analyzers
- Analysis tools:
  - ▷ S-PLUS
  - ▷ Extended programs based on S-PLUS
  - ▷ Utility programs
- Presentation/report generation:
  - ▷ S-PLUS
  - ▷ Other GUI tools

Summary and Perspectives

- Availability vs. needs
  - ▷ Need reliability measurement/improvement
  - ▷ Previously available:
    - – software reliability engineering
    - – software measurement & analysis
  - ▷ Available through this research:
    - – integrated approach
    - – effective and wide applicability
    - – techniques/tools (partially) developed
    - – deployment and validation

- Future work:
  - ▷ Deployment and technology transfer
  - ▷ Technique/tool refinement in response to practical problems