Software Reliability and Safety

CSE 8317 — Spring 2005

Prof. Jeff Tian, tian@engr.smu.edu
CSE, SMU, Dallas, TX 75275
(214) 768-2861; Fax: (214) 768-3085
www.engr.smu.edu/~tian/class/8317.05s

SRE.2: Reliability Models

- Reliability functions and definitions

- Software Reliability Growth Models

- Combinatorial and Other Models

- Model Assumptions/Limitations/Usage

Reliability Models

- Reliability modeling

  ▷ Reliability-fault relations

  ▷ Exposure assumptions

  ▷ Lyu book: Chapter 3; Tian/AIC paper

- Time domain SRGMs

  ▷ Reliability-fault relation over time

  ▷ Stochastic process for failure arrivals

  ▷ Reliability growth due to fault removal

- Combinatorial & other models

  ▷ Reliability-fault relation over input

  ▷ Fault seeding (FS) models

  ▷ Input domain (ID) models

  ▷ Cleanroom and coverage-based models

Develop and Use Models

1. Preparation:

   ▷ study failure data and environment
   ▷ choose reliability model(s)
   　(reliability expressed as math functions)
   ▷ influence of past experience

2. Modeling (function with parameters):

   ▷ estimate model parameters
   ▷ obtain fitted model
   ▷ goodness-of-fit test
   ▷ obtain performance measures

3. Followup and decision making:
   (assessment/prediction/control aspects)

Environment and Choice of Models

- Environment and data

  ▷ Modeling goals under the environment
  ▷ Environmental constraints
  ▷ Development and testing process
  ▷ Data availability
  ▷ Preliminary choice of models

- Model choice: goal driven

  ▷ Goal: assessment/prediction/control?
  ▷ Proper definition of reliability
    – time/input/stage/coverage?
  ▷ Current or future reliability?
  ▷ Reliability goals as exit criteria
  ▷ Resource allocation (time/distribution)
  ▷ Environmental/data constraints

Choice of Models

- Choice based on experience

  ▷ Previous choices and experience
    − models fitted obs. well?
    − other results: positive/negative?
    − overall feedback from development?
  ▷ Both local and non-local experience
  ▷ Baseline for comparison
  ▷ Adaptation and refinement for now

- Other factors

  ▷ Match model assumptions with reality
    − implications/limitations later
  ▷ Tools and software support
    − SMERFS, CASRE, etc. (Lyu Book)
    − integration with other tools?

Basic Functions and Definitions

- Some basic functions/definitions:

    ▷ $F(t)$: cdf for failure over time
    ▷ $f(t)$: pdf, $f(t) = F'(t)$
    ▷ Reliability function $R(t) = 1 - F(t)$

    $$R(t) = P(T \geq t) = P(\text{no failure by } t)$$

    ▷ Hazard function/rate/intensity

    $$z(t)\Delta t = P\{t < T < t + \Delta t | T > t\}$$

    ▷ Mean function $m(t)$ in NHPP
    ▷ Failure rate/intensity, $\lambda(t) = m'(t)$
    ▷ Time domain definition:

    $$R = \frac{s}{n} = \frac{n - f}{n} = 1 - \frac{f}{n} = 1 - r$$

    ▷ MTBF, MTTF, etc.

- Details/relations: Tian/SQE book Ch.22.

SRGM Classification

- Data used:

  ▷ Time-between-failure (TBF) models
    − r.v.: failure interval
  ▷ Failure-count (FC) models
    − r.v.: failure count for given interval
  ▷ Most widely used (in this class)
  ▷ Some models can use both TBF and FC
    data

- Other classifications possible

  ▷ Time measurement:
    − calendar/wall-clock/execution/etc. time
  ▷ Distribution/f-arrival function:
    − Poisson/binomial/etc.
  ▷ Finite vs infinite failures
  ▷ Musa Book, Chapter 9, Section 9.4.

TBF Models

- Model characteristics

    ▷ Failure intervals as r.v.
    - T_i: r.v. for the time between
      $(i-1)$st and ith failures
    ▷ Distribution $F_i(t)$ or $f_i(t)$
    ▷ Directly define $z_i(t)$
    ▷ Relate $z_i(t)$ to failures/faults

- Defining TBF models

    ▷ Sequence of $z_i(t)$ over i
    ▷ Initial value?
    ▷ Physical interpretation
    ▷ Cumulative or rate data plotting

TBF1: Jelinski-Moranda

- One of the earliest model using TBF (time-between-failure) measurement

- Failure rate (z_i or λ_i):

  ▷ Proportional to defects remaining
  ▷ Step function: $z_i = \phi(N - (i - 1))$
  ▷ z_i: failure rate for the i-th failure
  ▷ Two model parameters:
    $- \phi$ constant for failure exposure
    $- N$ constant for total defects

- Relation to later models

  ▷ Similar assumptions
  ▷ Other failure rate: geometric etc.
  ▷ Continuous version: Goel-Okumoto etc.

TBF2-3: Schick-Wolverton

- Variations of TBF1 model

- Schick-Wolverton linear model (TBF2):

    ▷ Proportional to defects remaining
    ▷ But it is time dependent
    ▷ Slope function with renewal
    ▷ $\lambda_i = \phi(N - (i - 1))t$
    ▷ Assumptions/parameters similar to TBF1

- Schick-Wolverton parabolic model (TBF2.1):

    ▷ Similar to TBF2
    ▷ 2nd order function with renewal
    ▷ $\lambda_i = \phi(N - (i - 1))(at^2 + bt + c)$
    ▷ Assumptions/parameters similar to TBF1

TBF4: Geometric Models

- Similar to Jelinski-Moranda

- Failure rate

  ▷ Step function but geometric step sizes
  ▷ $\lambda_i = \lambda_0 \phi^{i-1}$
  ▷ λ_i: failure rate for the i-th failure
  ▷ Two model parameters:
    − ϕ: step reduction/curvature
    − λ_0: initial failure rate

- Relation to later models

  ▷ Close relation to Musa-Okumoto model (logarithmic Poisson)
  ▷ Models defect discovery situations
  ▷ Hybrid geometric Poisson

$$\lambda_i = \lambda_0 \phi^{i-1} + c$$

TBF5: Imperfect Debugging

- Similar to Jelinski-Moranda

- Failure rate

  ▷ Goel-Okumoto
  ▷ Step function
  ▷ Allow for imperfect debugging
  ▷ $\lambda_i = \phi(N - p(i - 1))$
  ▷ p: prob(imperfect debugging)
  ▷ Other parameters same

- Relation to later models

  ▷ Close relation to Goel-Okumoto NHPP model
  ▷ Models defect removal process

TBF6: Littlewood-Verrall

- Bayesian model

  ▷ t_i: i-th inter-failure interval
  ▷ Distribution (pdf) for t_i:

  $$f(t_i|\lambda_i) = \lambda_i e^{-\lambda_i t_i}$$

  ▷ λ_i: failure rate parameter
  ▷ Distribution (pdf) for λ_i:

  $$f(\lambda_i|\alpha, \psi(i)) = \frac{[\psi(i)]^\alpha \lambda_i^{\alpha-1} e^{-\psi(i)\lambda_i}}{\Gamma(\alpha)}$$

  ▷ $\psi(i)$: increasing function of i
  ▷ α: constant

- In SMERFS, LV model with $\psi(i)$:

  ▷ $\psi(i) = \beta_0 + \beta_1 i$, or
  ▷ $\psi(i) = \beta_0 + \beta_1 i^2$

FC Models

- Model characteristics

  ▷ Failure count N_i as r.v.
  ▷ Time interval: predefined
    − equal: Schneidewind model
    − different: other models
  ▷ Distribution: failure arrival process
  ▷ Directly define process parameters
  ▷ NHPP most common

- Defining FC models

  ▷ Time intervals
  ▷ Underlying stochastic processes
  ▷ Physical interpretation
  ▷ Cumulative or rate data plotting

FC1: Goel-Okumoto

- Process assumption: NHPP
  (Non-homogeneous Poisson Process)

- Model definition:

  ▷ Probability of n failures in $[0, t]$:

  $$P(N(t) = n) = \frac{m(t)^n}{n!} e^{-m(t)}$$

  ▷ $m(t)$: mean function

  $$m(t) = N(1 - e^{-bt})$$

  ▷ $\lambda(t) = m'(t)$: failure rate

  $$\lambda(t) = Nbe^{-bt}$$

  ▷ N is the total estimated failures
  ▷ b captures failure exposure

- Data: period failure count (PFC model)
  ($N(t)$ is the random variable)

FC Models: Other NHPP

- Similar to Goel-Okumoto model

$$P(N(t) = n) = \frac{m(t)^n}{n!} e^{-m(t)}$$

- S-shaped SRGM (2 variations)

  ▷ $m(t) = N(1 - (1 + bt)e^{-bt})$
  ▷ $m(t) = N(1 - e^{-bt})(1 + ce^{-bt})$
  ▷ Allow for slow start

- Modified Goel-Okumoto

  ▷ $m(t) = N(1 - e^{-bt^c})$
  ▷ Similar to modified Jelinski-Moranda

- Logarithmic Poisson (Musa-Okumoto)

$$m(\tau) = \frac{1}{\theta} \log(\lambda_0 \theta \tau + 1)$$

FC Models: Generalized Poisson

- Differences with previous NHPP:

  ▷ Segmented rather that global NHPP
  ▷ Each segment has own parameters
  ▷ Sequence follows some function

- Schneidewind & Generalized Poisson:

  ▷ NHPP overall
  ▷ Each segment a Poisson process:

$$d_i(t) = \lambda_i(t) = \alpha e^{-\beta i}$$

  ▷ Generalized Poisson

$$m_i(t) = \phi(N - M_{i-1})g_i(x_1, x_2, \ldots, x_i)$$

  Can treat many models as special cases
  of this model

FC Models: Brooks–Motley

- Process assumption:

    ▷ Poisson variation
    ▷ Binomial variation

- Model definition:

    ▷ Each period a binomial/Poisson process
    ▷ Progression:
      − n_{ij} failures for ith session, jth module
      − with length K_{ij} or t_{ij}
      − binomial $q_{ij} = 1 - (1 - q)^{K_{ij}}$

$$P(X = n_{ij}) = \binom{N_{ij}}{n_{ij}} q_{ij}^{n_{ij}} (1 - q_{ij})^{N_{ij} - n_{ij}}$$

      − Poisson $\phi_{ij} = 1 - (1 - \phi)^{t_{ij}}$

$$P(X = n_{ij}) = \frac{\left(N_{ij}\phi_{ij}\right)^{n_{ij}} e^{-N_{ij}\phi_{ij}}}{n_{ij}!}$$

      − q_{ij}, ϕ_{ij}: binomial/Poisson constant

FC Models: Musa

- Variations of Musa models

  ▷ Prescriptive: derived from
    product/process characteristics
  ▷ Descriptive: fitted, similar to prev. SRGMs
  ▷ Execution time: used in modeling
  ▷ Calendar time: used in management
  ▷ Conversion between the two times

- Musa models (descriptive):

  ▷ Basic Musa: resembles Jelinski-Moranda
  ▷ (Musa-Okumoto) logarithmic Poisson
    (a variation of NHPP model)

  $$m(\tau) = \frac{1}{\theta} \log(\lambda_0 \theta \tau + 1)$$

  ▷ Execution time used in both above

FC Models: Musa

- Practicality of Musa models

  ▷ Software usage: operational profile and execution time

  ▷ Predictions (prescriptive) based on process and product characteristics

  ▷ Practical issues dealt in Musa book

  ▷ Practicality vs. theoretical focus

- Applications of Musa models

  ▷ AT&T projects: 10-20%

  ▷ Best practice at AT&T

  ▷ Adoption in other environments

  ▷ Tool and other support:
    - AT&T's SRE ToolKit
    - training and benchmarking

  ▷ Most publicized success stories

Choice of SRGMs

- Issues discussed before:

  - ▷ Goal/environment
  - ▷ Tool availability
  - ▷ Experience issues
  - ▷ Data: TBF vs. FC type

- Other model choice issues:

  - ▷ Time measurement and model fit.
  - ▷ Single vs. multiple models.
  - ▷ Existing vs. new models.
  - ▷ Assumptions/limitations/applicability.
  - ▷ (to be examined further next...)

Choice of SRGMs

- Time measurement and model fit:

  ▷ experience at AT&T (exec. time!)

  ▷ IBM experience

  ▷ bad fit $\Rightarrow$ time appropriate?

  ▷ (compare: bad fit $\Rightarrow$ other model)

- Single vs. multiple models:

  ▷ best fitted vs. optimistic (fast rel. growth) vs. pessimistic (slow ..)

  ▷ band/range instead of single estimate

  ▷ related: synthesized models

- Existing vs. new models:

  ▷ simplicity of existing models

  ▷ validation of new models

  ▷ caution against ad-hoc new models

Alternatives to SRGMs

- *Reliability:* Prob(failure-free operation)

  ▷ Time: how to measure $\Rightarrow$ SRGMs
  ▷ Input: characterize/classify
  ▷ Assumptions: failure/OP/time/distr
  ▷ Applicability and limitations

- Alternatives to SRGMs:

  ▷ Input domain/combinatorial
    − also fault seeding
  ▷ Hybrid models: Cleanroom model
  ▷ Coverage-based and predictive
  ▷ TBRMs: tree-based reliability models
    − both time/input info. (SRE.3)

Mills Fault Seeding Model

- Assumptions (BIG!)

    ▷ Random seeding, same distribution
    ▷ Same probability for detection
    ▷ Hyper-geometric distribution

- Seeding/tagging to estimate population

    ▷ n_s seeded, x_s captured
    ▷ n_o original, x_o captured
    ▷ Prob(finding exactly x_s and x_o):

$$P = \frac{\binom{n_o}{x_o}\binom{n_s}{x_s}}{\binom{n_o + n_s}{x_o + x_s}}$$

    ▷ ML estimate of n_o given by $\widehat{n_0}$

$$\widehat{n_0} = \frac{n_s x_o}{x_s}$$

Nelson's Input Domain Model

- 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: f.
    ▷ Estimated reliability = success rate:

    $$R = \frac{n - f}{n} = 1 - \frac{f}{n} = 1 - r$$

    ▷ r: failure rate.

- Repeated sampling without fixing.

Other Input Domain Models

- Brown-Lipow model:

  ▷ Explicit input state distribution.
  ▷ Known probability for sub-domains E_i
  ▷ f_i failures for n_i runs from subdomain E_i

$$R = 1 - \sum_{i=1}^{N} \frac{f_i}{n_i} P(E_i)$$

- Ramamoorthy-Bastani:

  ▷ Safety critical systems, $\hat{R} = 1$
  ▷ Confidence level for $\hat{R}$
  ▷ x_i specific set of inputs
  ▷ P(program correct | correct for x_i's)

$$P = e^{-\lambda V} \prod_{i=1}^{n-1} \frac{2}{1 + e^{-\lambda x_i}}$$

  ▷ λ source code complexity
  ▷ Recent development by Woit-Parnas

Ho's Input Domain Model

- Step 1: Symbolic execution tree

    ▷ Execution tree generation
    ▷ Path identification T_i
    ▷ Path frequency assignment p_i

- Step 2: Path reliability R_i

    ▷ Estimate vs. bound
    ▷ Use Nelson models
    ▷ Ramamoorthy-Bastani model

- Step 3: System reliability for m paths with probability p_i and reliability R_i

$$R = \sum_{i=1}^{m} p_i R_i$$

Cleanroom Reliability Model

- Hybrid model

  ▷ Reliability growth over stages
  ▷ Random sampling within stage

- Factors affecting reliability

  ▷ Increment testing: reliability change
  ▷ Mixture of untested and tested codes

- Certifying statistical quality

  ▷ MTTF $= MR^c$
  ▷ M: Initial MTTF
  ▷ R: Effective ratio for change
  ▷ c: software changes

Coverage and Coverage-Based Models

- Alternative: coverage analysis

  ▷ Defect fixing effect

  ▷ Infeasibility of exhaustive testing

  ▷ Pure coverage vs. cov-based models

- Focus on input/internal state coverage:

  ▷ Function/data/statement coverage.

  ▷ Path and dependency coverage.

  ▷ Assumption: coverage$\uparrow$ $\Rightarrow$ reliability $\uparrow$
  (qualitative relation, not quantified)

- Coverage-based modeling:

  ▷ Analytical: Weyuker etc.

  ▷ Empirical: Mathur etc.

  ▷ Mixed: Chen/Lyu/Wong.

General Assumptions and Implications

- Times between failures are independent

  ▷ Implies randomized testing
  ▷ Practical scenarios:
    − defect fixing effect
    − structure/progression in testing

- Immediate defect removal

  ▷ Duplicate defect counting
  ▷ Related but not duplicate?
  ▷ Infeasible for in-field defects

- No new fault injected

  ▷ Reliability growth assured
  ▷ Practical: injection $<$ removal
  ▷ Related: Decreasing failure rate

Assumptions and Implications

- Relating failure rate to number of faults

  ▷ Variations to the assumption
    − proportionality between the two
    − functional relation between the two
    − time dependent relation
  ▷ Implications of failure detection and detection sequences

- Operational profile

  ▷ Ensures reasonable/meaningful reliability assessments and predictions
  ▷ Limits applicability

- Time as a basis for failure rate

  ▷ Equivalent time units
  ▷ Requires proper time measurement

Assumptions and Applicability

- General considerations

  ▷ Assumptions for different model types
  ▷ Tian/AIC paper
  ▷ Match them to application environment
    − models necessarily simple
    − impossible perfect match

- Applicability to different processes

  ▷ Waterfall generally assumed
  ▷ Testing phases
  ▷ Black-box testing: SRGMs and ID
  ▷ White-box testing: coverage
  ▷ Incremental development: cleanroom
  ▷ Spiral model: iterations
  ▷ Operational phases
    − difference in defect removal
    − data availability

Applicability to Different Phases

- Requirement and specification

  ▷ Reliability goal from customer expecta-
    tion and feasibility (also affordable?)
  ▷ Operational profile construction
  ▷ Prepare for random testing

- Design and coding

  ▷ Fault detection and removal (QA)
  ▷ Musa's prescriptive model
  ▷ Other existing models not applicable
  ▷ Alternative models may be needed:
    – fault and error based models
    – constructive information (white box)
    – predictive models relating to reliability

Applicability to Different Phases

- Unit testing

  ▷ White-box deterministic testing
  ▷ Tester = developer
  ▷ Applicable: fault seeding, coverage-based, (Musa's prescriptive?)
  ▷ Other models not applicable

- Integration and system testing

  ▷ FVT, SVT, regression, integration
  ▷ Focus: customer oriented operations
  ▷ Less emphasis on coverage
  ▷ Main phase for SRGMs
  ▷ FC models more robust
  ▷ Random testing conformance?
  ▷ Use of other models

Applicability to Different Phases

- Acceptance testing

  - ▷ Gate: accept/release or not
    (also plan for product support)
  - ▷ Basis: snapshot(s) or random sampling
  - ▷ Cleanroom-like model usage
  - ▷ Input domain model appropriate
  - ▷ Others, maybe?

- Operational phase

  - ▷ Actual operational (post-release)
  - ▷ Beta or ECI programs (pre-release)
  - ▷ Difference in operational environments
  - ▷ Data availability and treatment
  - ▷ Defect fix and product refreshing
    (availability/expectation management)
  - ▷ Business decisions

Applications and Examples

- Overall procedure

  ▷ A lot of preparation

  ▷ Generic: preparation/modeling/followup

  ▷ Routine procedure once started

  ▷ Often periodic activities

  ▷ Evaluation/feedback/improvement

- Application examples

  ▷ Data: telecommunications (Musa)

  ▷ Wide applications of Goel-Okumoto, Musa, and other models

  ▷ Shuttle: Schneidewind and Keller

  ▷ Examples in IBM