Some Successful Approaches to Software Reliability Modeling in Industry

Daniel R. Jeske and Xuemei Zhang
Bell Laboratories, Lucent Technologies
Holmdel, NJ

1. Introduction and Context
2. Software Reliability Growth Models
3. Architecture-Based Software Reliability Models
4. Case Studies
5. Summary

ASA Conference on Quality and Productivity
May 21, 2003
When are Software Reliability Models Typically Applied?

**Architecture-Based Reliability Models**

.....decisions are being made as to how to design reliability into the system

**Software Reliability Growth Models**

......reliability is quantified and influences the release decision

......reliability predictions are verified

......parameters useful for modeling reliability of next release are estimated
Software Reliability Growth Models (SRGMs)
SRGMs: Questions to Answer

• What would the user-perceived failure rate of the software be if the software was to be released now?

• How much more testing is needed to achieve the reliability targets?

• How many faults exist in the code at the end of system test?
**A Well-Known SRGM**

**Goel-Okumoto Model**

\[ M(t) = c(1 - e^{-d t}) \]

**Cumulative Failures**

**Cumulative Test Time (days)**

\[ \lambda(T) = M'(T) = cde^{-dT} \]

- \( c \) = initial number of faults
- \( d \) = average per-fault failure rate
- \( t \) = cumulative test time, aggregated across all installations
- \( T \) = current value of \( t \)
The expected number of faults detected in a time interval \((t, t + \Delta t)\) is proportional to the number of faults remaining in the software at time \(t\).

Thus,

\[ M(t + \Delta t) - M(t) = d[c - M(t)]\Delta t \]

or equivalently,

\[ M'(t) + dM(t) = cd \]

implying,

\[ M(t) = c(1 - e^{-dt}) \]

Goel and Okumoto further assume that the number of faults observed in \((0, t)\) is Poisson\{\(M(t)\}\).
S-shaped Model

Cumulative Failures

Cumulative Test Time (days)

\[
\begin{align*}
d(t) &= \frac{d}{1 + \beta e^{-\beta t}} \\
M(t) &= \frac{1}{(1 + \beta e^{-\beta t})} \times c(1 - e^{-\beta t})
\end{align*}
\]
Challenges With Applications

Traditional SRGMs are applied to system test data with the hope of obtaining an estimate of software failure rate that will be observed in the field. The following issues have to be taken into consideration:

- The usage profile in the field is typically very different than the testing profile
- Fault removal in field environments is not instantaneous
- Quality of software reliability data
Non-Instantaneous Fault Removal Times In Field Environments

G-O Model

\[ M(t) = c[1 - e^{-dt}] \]
\[ \lambda(t) = cde^{-dt} \]

where:
- \( c \) initial number of faults at time of field deployment
- \( d \) average per fault failure rate in a field environment
- \( p \) probability that a detected fault is successfully removed

G-O Model with Imperfect Debugging

\[ M(t) = \frac{c}{p}[1 - e^{-dpt}] \]
\[ \lambda(t) = cde^{-dpt} \]
• Perfect debugging assumption is an acceptable assumption (based on thorough regression tests)

• We relate non-instantaneous fault removal to imperfect debugging

Under the imperfect debugging model:

\[ E(\text{# of occurrences of each fault}) = \frac{1}{p} \]

Under the situation of non-instantaneous fault removals, if \( \mu \) denotes the mean time to remove a fault and there are \( n \) systems in the field

\[ E(\text{# of occurrences of each fault}) = 1 + n\mu d \]
Non-Instantaneous Fault Removal Times
In Field Environments

\[ M(t) = c(1 + n\mu d)[1 - e^{-\frac{d}{1+n\mu d} \times t}] \]

\[ \lambda(t) = cde^{-\frac{d}{1+n\mu d} \times t} \]

where:
- \( c \) initial number of faults at time of field deployment
- \( d \) average per fault failure rate in the field environment
- \( \mu \) average time to remove a fault (\( \mu = 0 \) gives back the G-O model)
- \( n \) number of systems in the field
- \( t \) cumulative exposure time, aggregated across all installations
Field Failure Rate Prediction if Testing Profile Matches the Field Usage Profile

The number of initial faults at time of field deployment time is the same as the number of residual faults after testing has completed.

The average per fault failure rate in the field environment is identical to the average per fault failure rate in the test environment.

For the field failure rate model, we can use the G-O model replacing $c$ with $ce^{-dT}$ and adjusting for non-instantaneous removal times.

$$\lambda(t) = (ce^{-dT})de^{-\frac{d}{1+n\mu_d}t}$$
Field Failure Rate Prediction When Testing Profile Does Not Match the Field Usage Profile

1) Testing and field usage profiles drive the average per fault failure rate. Usually the failure rate of faults is smaller in field environments than in test environments.

2) Define $K = d / d^*$, where $d^*$ is the average per fault failure rate in the field environment.

3) $K$ is the “per fault failure rate” calibration factor.

4) Estimate $K$ from previous releases of the software, or from related projects.

$$\lambda_{adj}(t) = (ce^{-dT}) (d / K) e^{-d / K} \times t$$
Case Study: GSM Wireless System

BCF Software
- setup calls
- tear down calls
- negotiate cell handoffs
- dynamically control transmit power levels
- alarm handling
- overload control
- equipment provisioning

BCF - Base Station Controller Frame
BTS - Base Transceiver Station
MSC - Mobile Switching Center
PSTN - Public Switched Telephone Network
Objective

Goals
Estimate the field failure rate of R3 BCF software which is currently in system test

Data Available
1) R3 Test Data in a non-operational profile environment
2) Field failure data for R1 and R2
### Field Failure Data for R1

<table>
<thead>
<tr>
<th>Month</th>
<th>Days</th>
<th>Cumulative</th>
<th>Month</th>
<th>Failures</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,249</td>
<td>1,249</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3,472</td>
<td>4,721</td>
<td>6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>4,065</td>
<td>8,786</td>
<td>4</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>4,883</td>
<td>13,669</td>
<td>3</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>5,425</td>
<td>19,094</td>
<td>6</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>5,656</td>
<td>24,750</td>
<td>1</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>7,549</td>
<td>32,299</td>
<td>2</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>8</td>
<td>8,295</td>
<td>40,594</td>
<td>4</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>8,882</td>
<td>49,476</td>
<td>1</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td>6,120</td>
<td>55,596</td>
<td>0</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>2,465</td>
<td>58,061</td>
<td>1</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>527</td>
<td>58,588</td>
<td>1</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>45</td>
<td>58,633</td>
<td>0</td>
<td>33</td>
<td>33</td>
</tr>
</tbody>
</table>
Field Failure Data Analysis

R1 Field

\( T_1 = 58,633 \) system-days (13 months, 167 systems)
\( \mu = 45 \) days
\( \hat{c}_1 = 22.3 \) faults
\( \hat{d}_1 = 0.0000746 \) failures/day/fault
   (StdError: \( 3.07 \times 10^{-5} \))

R2 Field

\( T_2 = 167,900 \) system-days (13 months, 370 systems)
\( \mu = 45 \) days
\( \hat{c}_2 = 114.01 \) faults
\( \hat{d}_2 = 0.0000128 \) failures/day/fault
   (StdError: \( 3.07 \times 10^{-6} \))
# R3 Test Data

As many as 11 frames were being used in parallel during system test.

‘Failure’ is defined as a severity 1, 2 or 3 MR.

<table>
<thead>
<tr>
<th>Calendar Week</th>
<th>Number of Days of Each Software Installation</th>
<th>Cumulative System-Days of Testing</th>
<th>Cumulative Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>2</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>3</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>4</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>5</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>6</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>7</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>8</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>9</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
<tr>
<td>10</td>
<td>5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</td>
<td>5 9 13 18 28 33 43 63 88 123</td>
<td>5 6 13 13 22 24 29 34 40 46</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>33</th>
<th>34</th>
<th>35</th>
<th>36</th>
<th>37</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>955</td>
<td>977</td>
<td>999</td>
<td>1001</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>170</td>
<td>176</td>
<td>180</td>
<td>181</td>
<td>181</td>
</tr>
</tbody>
</table>
G-O Model Fit to the R3 Test Data

\[ M(t) = \hat{c}(1 - e^{-\hat{d}t}) \]

\[
\begin{align*}
\left(\begin{array}{c}
\hat{c} \\
\hat{d}
\end{array}\right) &= \left(\begin{array}{c}
238.3 \text{ faults} \\
0.00142 \text{ fails/day/fault}
\end{array}\right) \\
K &= \frac{\hat{d}}{(\hat{d}_1 + \hat{d}_2)/2} = 31 \\
V\text{âr} \left(\begin{array}{c}
\hat{c} \\
\hat{d}
\end{array}\right) &= \left[\begin{array}{cc}
27.02^2 & -0.00552 \\
-0.00552 & 0.00027^2
\end{array}\right]
\end{align*}
\]
R3 Field Failure Rate Prediction

\[ \hat{\lambda}_{\text{adj}}(t) = 58 \times (365 \times 0.00142 / 31) \times e^{- \frac{0.00142/31}{1 + 345 \times 45 \times 0.00142/31} \times 365 \times 345 \times t} \]

(fails/year, after \( t \) calendar-years of field exposure)
Architecture-Based Software Reliability Models
Questions to Answer

• What type of software redundancy, if any, is needed in order to achieve the reliability target?

• How fast do fault recovery times need to be?

• Should software processes try to restart before a failover is attempted?

• How thorough do system fault diagnostics need to be?

• What is the required software failure rate?
Case Study - Continued

**BTS Controller Software**
- allocate radio resources
- negotiate hand-offs
- manage connections between BTS and BCF
- provision and maintenance

**Alternative Architectures**
- Simplex, no process restart capability
- Simplex, with restart capability
- Active-Standby, with fail-over capability

---

**Definitions**

- BCF - Base Station Controller Frame
- BTS - Base Transceiver Station
- MSC - Mobile Switching Center
- PSTN - Public Switched Telephone Network
General Active-Standby Software Availability Model
Special Case: Cold Standby

Diagram:
- Node 0: Parallel Reboot
  - Undetected SW failure: \((1-c)(\lambda_1 + \lambda_2)\)
  - Detected non-restartable SW failure: \(c\lambda_2\)
  - Detected restartable SW failure: \(c\lambda_1\)

- Node 1: Restart Fails, System Detects
  - (1 - \(r\))\(q\mu\)

- Node 2: Restart Fails, System Does not Detect
  - (1 - \(r\))(1 - \(q\))\(\mu\)

- Node 3: Parallel Reboot
  - Manual Detection

- Node 4: AutoFailover Succeeds
  - \(p\mu_f\)

- Node 5: AutoFailover Fails, System Detects
  - \((1 - p)q_f\mu_f\)

- Node 6: AutoFailover Fails, System Does not Detect
  - \((1 - p)(1 - q_f)\mu\)

- Node 7: Manual Detection
  - \(\mu_{mf}\)
  - Manual Failover Fails

- Node 8: Manual Failover Success
  - \(P_{mf}\mu_{mf}\)

- Node 11: Manual Detection
  - \(\mu_{d} + \lambda_3\)
Simplex Systems as Special Cases

Simplex System With No Restart Capability

\[ \lambda_1 = 0 \text{ (no restartable software)} \]
\[ \lambda_{1s} = \lambda_{2s} = \lambda_{3s} = 0 \text{ (no standby side)} \]
\[ r = 0 \text{ and } \mu_s = \infty \text{ (no restart attempts)} \]
\[ p = 0 \text{ and } \mu_f = \infty \text{ (no automatic failover attempts)} \]
\[ p_m = 0 \text{ and } \mu_{mf} = \infty \text{ (no manual failover attempts)} \]
\[ q_m = q_f = 1 \text{ (no restart or failover attempts)} \]
\[ r_m = 1 \text{ and } \mu_m = \text{ manual recovery duration of} \]

Simplex System With Restart Capability

\[ \lambda_{1s} = \lambda_{2s} = \lambda_{3s} = 0 \text{ (no standby)} \]
\[ p = 0 \text{ and } \mu_f = \infty \text{ (no automatic failover attempts)} \]
\[ p_m = 0 \text{ and } \mu_{mf} = \infty \text{ (no manual failover attempts)} \]
\[ q_f = 1 \text{ (no automatic failover attempts)} \]
Network Traffic Profile

% of Busy Hour Traffic

Hour of Day
|
|---|
| **Reward Model** |

\[ C = \% \text{ Capacity Remaining After the Failure} \]
\[ C_i = \text{Capacity Required for Traffic Volume Offered in Hour-i} \]
\[ S_i = \% \text{ Offered Traffic Served if Failure Occurs in Hour-i} \]
\[ = \min(C_i, C)/C_i \]
\[ \bar{S} = \text{Expected \%Traffic Served, Given a Failure That Leaves Capacity at C} \]
Feasibility Regions for Critical Parameters

Level Curves for 99.999% Availability
(all other parameters are fixed)

\( \lambda_1 + \lambda_2 \)

\( C \)
Identifying Influential Parameters

Challenges

• Availability formula is straightforward to obtain, but is complicated in that it is highly non-linear and it depends on numerous (23) parameters.

• Analysis of derivatives, while also straightforward is not particularly insightful, since they themselves depend on the same set of parameters.

• “Tornado Graphs” depends too much on the fixed values for the other parameters.

Approach

• Identify likely ranges for each parameter to define the domain of the availability function.

• Uniformly sample N times from the domain to obtain $A_1, A_2, \ldots, A_N$.

• Use an efficient second-order polynomial to approximate the availability formula.

• Use the t-statistics as the associated measure of influence.
### Case Study – Continued
(Likely Ranges for Parameter Values in Active-Standby Architecture)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>Fails/yr</td>
<td>[0.1, 1.0]</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>Fails/yr</td>
<td>[0.1, 1.0]</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>Fails/yr</td>
<td>[0.1, 0.5]</td>
</tr>
<tr>
<td>$\lambda_{1s}$</td>
<td>Fails/yr</td>
<td>0</td>
</tr>
<tr>
<td>$\lambda_{2s}$</td>
<td>Fails/yr</td>
<td>[0.1, 0.5]</td>
</tr>
<tr>
<td>$\lambda_{3s}$</td>
<td>Fails/yr</td>
<td>[0.1, 0.5]</td>
</tr>
<tr>
<td>$\mu_s$</td>
<td>Restarts/yr</td>
<td>31,536,000</td>
</tr>
<tr>
<td>$\mu_f$</td>
<td>Failovers/yr</td>
<td>3,153,600</td>
</tr>
<tr>
<td>$\mu_r$</td>
<td>Reboots/yr</td>
<td>26,280</td>
</tr>
<tr>
<td>$\mu_d$</td>
<td>Detections/yr</td>
<td>[1095, 2190]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{mf}$</td>
<td>Failovers/yr</td>
<td>26,280</td>
</tr>
<tr>
<td>$\mu_m$</td>
<td>Recoveries/yr</td>
<td>17,520</td>
</tr>
<tr>
<td>$r$</td>
<td></td>
<td>[0.9, 0.99]</td>
</tr>
<tr>
<td>$r_m$</td>
<td></td>
<td>[0.9, 0.99]</td>
</tr>
<tr>
<td>$q_r$</td>
<td></td>
<td>[0.9, 0.99]</td>
</tr>
<tr>
<td>$q_f$</td>
<td></td>
<td>[0.9, 0.99]</td>
</tr>
<tr>
<td>$p$</td>
<td></td>
<td>[0.9, 0.99]</td>
</tr>
<tr>
<td>$p_m$</td>
<td></td>
<td>[0.9, 0.99]</td>
</tr>
<tr>
<td>$c$</td>
<td></td>
<td>[0.9, 0.99]</td>
</tr>
<tr>
<td>$w$</td>
<td></td>
<td>[0.9, 0.99]</td>
</tr>
</tbody>
</table>
Case Study – Continued
(Efficient Second-Order Regression Model)

\[
D(\text{min/yr}) = 497 + 379\lambda_1 + 378\lambda_2 - 358\lambda_1 c - 356\lambda_2 c + 284(\mu_d /1000)c - 268(\mu_d /1000) - 461c
\]

-12.2\lambda_1 (\mu_d /1000) + 319\lambda_3 - 12\lambda_2 (\mu_d /1000) - 298\lambda_3 c

- 22p - 11.6\lambda_3 (\mu_d /1000) - 11.7r - 9.6q_f - 9q_r - 5.8r_m - 5w - 2.1p_m

**Adequacy of Fit**

**Interpretation**

**Influential Parameters**
- Software failure rates on active side
- Silent failure detection time
- Coverage factor
- Cross-product of coverage factor with software failure rates and silent failure detection time

**Weakly Influential Parameters**
- All of the success probabilities
Needed Research

- SRGM fitting tools
- Small sample inference procedures
- Failure rate template
- Improved methods for linking SRGMs to architecture-based models