Software Failure and Reliability Assessment Tool (SFRAT): An Open Source Application for the Practitioner and Research Community

Vidhyashree Nagaraju\textsuperscript{1}, Thierry Wandji\textsuperscript{2}, and Lance Fiondella\textsuperscript{1*}

\textsuperscript{1}University of Massachusetts, North Dartmouth, MA 02747
\textsuperscript{2}Naval Air Systems Command, Patuxent River, MD 20670
Motivation

• Many government organizations and national labs depend on mission and life critical software
  • To assure national security
  • Safety of human operators and communities in which software enabled systems reside

Reliability key to success of software
Motivation (2)

• Department of Defense (DoD) increasingly depends on software intensive systems
  • Mission and life critical
  • Must preserve high reliability and availability

• Urgency to deploy new technologies and military capabilities may result in
  • Inadequate reliability testing
  • Severe economic damage and loss of life
Background

• Recent National Academies report on Enhancing Defense System Reliability recommends
  – Use of reliability growth models to direct contractor design and test activities

• Several tools to
  – Automatically apply reliability models
  – Automate reliability test and evaluation
Existing Tools

• CASRE (Computer-Aided Software Reliability Estimation Tool)
  – Incorporates SMERFS (Statistical Modeling & Estimation of Reliability Functions for Software)
  – Automatically ranks models according to set of goodness of fit measures

• Caution: Users strongly advised to study underlying mathematics
  – Can better inform model selection process
Shortcomings of existing tools

• Over 20 years old
  – Not updated in over 15 years
  – Not compatible with current operating systems

• Interface
  – Does not impose intuitive workflow
  – Possible to run models on data that does not exhibit reliability improvement
  – User may fail to recognize lack of model fit

• Not open source (not sustainable)
  – Inhibits dialog between researchers and practitioners
Software Failure and Reliability Assessment Tool (SFRAT)

- SFRAT is an open source application
  - Designed for practitioner and research community
  - Programmed in R and provides functionality through a Shiny graphical user interface
- Reduces the need for knowledge of the underlying statistical techniques
  - Can help contractors quantitatively assess software as part of their data collection and reporting process
Software Failure and Reliability Assessment Tool (SFRAT) (2)

- Allows users to answer following questions about a software system during test
  1. Is the software ready to release (has it achieved a specified reliability goal)?
  2. How much more time and test effort will be required to achieve a specified goal?
  3. What will be the consequences to system’s operational reliability if not enough testing resources are available?
SFRAT Output/Deliverables

• Trend tests
• Model rankings

• Visualization
  – Cumulative failure plot
  – Time between failure plot
  – Failure intensity plot
  – Reliability growth plot

• Predictions
  – Time to achieve reliability
  – Expected number of faults for next $t$ time units
  – Expected time to next $k$ failures
SRGM classification

• Based on data formats
  – Failure Rate models
    • Inter-failure times - time between \((i - 1)^{st}\) and \(i^{th}\) failure, defined as \(t_i = (T_i - T_{i-1})\)
    • Failure times – vector of failure times,
      \(T = \langle t_1, t_2, \ldots, t_n \rangle\)
  – Failure Counting models
    • Failure count data - length of the interval and number failures observed within it,
      \(< T, K > = \langle (t_1, k_1), (t_2, k_2), \ldots, (t_n, k_n) \rangle\)
SFRAT - Tab view

- Open, analyze, and subset file
- Apply models, plot results
- Detailed model queries
- Evaluate model performance
Input File Format

• Excel or csv
• First row indicates type of failure data
  – FN – Failure number
  – IF – Inter failure times
  – FT – Failure times
  – FC – Failure count
• Regardless of input format
  – Tool converts data to other two formats
Tab 1
Select, Analyze, and Filter data
Tab 1 – After data upload

Cumulative failure data view
Laplace trend test – SYS1 data

Decreasing trend indicates reliability growth (application of SRGM appropriate)
Laplace trend test – J4 data

Does not exhibit reliability growth (additional testing required)
Increasing trend indicates reliability growth
Tab 2
Set Up and Apply Models
Tab 2 view

Configure and Apply Models
Specify the number of failures for which the models will make predictions
Specify for how many failures into the future the models will predict.
1

Choose one or more models to run, or exclude one or more models:
- Delayed S-Shape
- Geometric
- Goel-Okumoto
- Jelinski-Moranda
- Weibull

Run Selected Models

Display Model Results
Choose one or more sets of model results to display.
No model results to display

Choose the type of plot for model results.
Choose a plot type:
- Cumulative Failures

For how much time should the model results curve extend beyond the last prediction point?
10000

Show data on plot
Show end of data on plot

Draw the plot with data points and lines, points only, or lines only?
- Both
- Points
- Lines

Choose the type of file to save plots. Tables are saved as CSV files.
- JPEG
- PDF
- PNG
- TIFF

Save
Cumulative failures

Plot enables comparison of data and model fits
Time between failures

Interfailure Times vs. Cumulative Test Time for SYS1

Times between failures should increase (indicates reliability growth)
Failure intensity

Failure intensity should decrease (indicates reliability growth)
Reliability growth curve

Can determine time to achieve target reliability
Tab 3: Model predictions

• Allows users to answer the following questions
  1. How much time will be required to observe the next N failures
  2. How many failures will be observed over the next N time units?
  3. How much more test time to achieve a specified reliability?
Tab 3 Options

Make Detailed Predictions From Model Results

Choose one or more sets of model results to display.

- Delayed S-Shape
- Geometric
- Goel-Okumoto
- Jelinski-Moranda
- Weibull

How much time will be required to observe the next N failures

Specify the number of failures that are to be observed.

- 1

How many failures will be observed over the next N time units?

Specify the amount of additional time for which the software will run.

- 4116

How much more test time to achieve a specified reliability?

Specify the desired reliability.

- 0.9

Specify the length of the interval for which reliability will be computed.

- 4116

Save detailed model results as PDF or CSV?

- CSV
- PDF

Save Model Predictions
## Failure Predictions

<table>
<thead>
<tr>
<th>Model</th>
<th>Time to achieve R = 0.9 for mission of length 4116</th>
<th>Expected # of failures for next 4116 time units</th>
<th>Nth failure</th>
<th>Expected times to next 1 failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>2: Dalayad S-Shape</td>
<td>12401.1541529881</td>
<td>0.24886262198799</td>
<td>1</td>
<td>NA</td>
</tr>
<tr>
<td>3: Goel-Okumoto</td>
<td>62826.767207733</td>
<td>0.930615409966553</td>
<td>1</td>
<td>4551.23466949961</td>
</tr>
<tr>
<td>4: Jelinski-Moranda</td>
<td>59916.2917467156</td>
<td>0.856125458252314</td>
<td>1</td>
<td>4869.80650250625</td>
</tr>
<tr>
<td>5: Weibull</td>
<td>259865.770847692</td>
<td>1.72565369956707</td>
<td>1</td>
<td>2353.05254648438</td>
</tr>
<tr>
<td>6: Geometric</td>
<td>1592716.45836287</td>
<td>1.97747306676007</td>
<td>1</td>
<td>2170.0308926578</td>
</tr>
</tbody>
</table>

Showing 1 to 10 of 10 entries
Evaluate Models
Tab 4 Options

Model assessment based on AIC and PSSE
# AIC and PSSE

<table>
<thead>
<tr>
<th>Model</th>
<th>AIC</th>
<th>PSSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delayed S-Shape</td>
<td>2075.14631533222</td>
<td>296.349252292955</td>
</tr>
<tr>
<td>Geometric</td>
<td>1937.03417425106</td>
<td>84.3270812346017</td>
</tr>
<tr>
<td>Goel-Okumoto</td>
<td>1953.61306630984</td>
<td>23.0712669112105</td>
</tr>
<tr>
<td>Jelinski-Moranda</td>
<td>1960.53413167966</td>
<td>19.6003726994495</td>
</tr>
<tr>
<td>Weibull</td>
<td>1938.16066975807</td>
<td>74.9446562450499</td>
</tr>
</tbody>
</table>

Lower values preferred
Conclusions and Future Research

• Presented open source application to promote collaboration among
  – Members of software reliability research community
  – Users from industry and government organizations

• Application architecture enables integration of models from research literature

• Future research will expand architecture to enable models for other stages of SDLC
Software Reliability Tool

Available online

http://sasdlc.org/lab
Acknowledgement

• This work was partially supported by (i) the Naval Air Warfare Center (NAVAIR) under Award Number N00421-16-P-0521 and (ii) the National Science Foundation (NSF) (#1526128).