More Reliable Software
Faster and Cheaper – An Overview

John D. Musa
j.musa@ieee.org

Most Important Software Problem

1. Customers demand (in average order):
   A. More reliable software
   B. Faster (includes “with greater agility”)
   C. Cheaper
2. Your success in meeting demands affects market share, profitability, your career
3. Demands conflict, causing risk and overwhelming pressure

Software Reliability Engineering –
Developed to Address the Problem

1. It differs from other approaches by being primarily quantitative.
2. You add and integrate software reliability engineering (SRE) with other good processes and practices; you do not replace them.
3. With SRE you control the development process, it doesn’t control you:
   A. Development process is not externally imposed.
   B. You use quantitative information to choose the most cost-effective software reliability strategies for your situation.

Outline

1. Nature of software reliability engineering (SRE)
2. SRE process (with illustration)
How Does SRE Work?

1. Increase effective resources
   A. Quantitatively characterize expected use
   B. Focus resources (for example, review time, unit code and test time, test cases, test time) on most used and most critical functions
   C. Maximize test effectiveness by making test highly representative of field

2. Apply resources to maximize customer value by matching quality needs
   A. Set quantitative objectives for major quality characteristics (reliability and/or availability, delivery time, price)

3. Choose software reliability strategies to meet objectives
   B. Track reliability in system test against objective as one of the release criteria

SRE - A Proven, Standard, Widespread Best Practice

1. Proven practice
   A. Example: AT&T International Definity PBX [6, pp 167-8]
      a. Reduced customer-reported problems by factor of 10
      b. Reduced system test interval by factor of 2
      c. Reduced total development time by 30%
      d. No serious service outages in 2 years of deployment
SRE - A Proven, Standard, Widespread Best Practice

B. AT&T Best Current Practice since 5/91
(based on widespread practice, documented strong benefit/cost ratio, probing review) [6, pp 219-254]

2. Standard practice
   B. IEEE standard since 1988
   C. AIAA standard since 1993

SRE Is Widely Applicable

1. Technically speaking, you can apply SRE to any software-based product, beginning at start of any release cycle.

2. Economically speaking, the complete SRE process may be impractical for small components (involving perhaps less than 2 staff months of effort), unless used in a large number of products. It may still be worthwhile to implement it in abbreviated form.

3. Independent of development technology and platform

4. SRE requires no changes in architecture, design, or code - but it may suggest changes that would be beneficial.

Activities of SRE Process and Relation to Software Development Process

1. Define the Product
2. Implement Operational Profiles
3. Engineer “Just Right” Reliability
4. Prepare for Test
5. Execute Test
6. Guide Test

Requirements and Architecture | Design and Implementation | Test
Reliability and Availability Definitions

1. **Reliability**: the probability that a system or a capability of a system will continue to function without failure for a specified period in a specified environment. The period may be specified in natural or time units.
   - **Natural unit**: unit other than time related to amount of processing performed by software-based product, such as runs, pages of output, transactions, telephone calls, jobs, semiconductor wafers, queries, or API calls
   - **Failure intensity (FI)**: failures per natural or time unit, an alternative way of expressing reliability

2. **Availability**: the average (over time) probability that a system or a capability of a system is currently functional in a specified environment

---

Illustration - FONE FOLLOWER (FF) - Product Description

1. Subscriber calls FF, enters planned phone numbers (forwarders) to which calls are to be forwarded vs time.
2. FF forwards incoming calls (voice or fax) from network to subscriber as per program. Incomplete voice calls go to pager (if subscriber has one) and then voice mail.

---

Activities of SRE Process and Relation to Software Development Process

1. **Define the Product**
2. Implement Operational Profiles
3. Engineer "Just Right" Reliability
4. Prepare for Test
5. Execute Test
6. Guide Test

---

Define the Product

1. Who is supplier?
2. Who are customers and users?
3. List associated systems
   - *associated system*: base product or system specially related to it that is tested separately
   - **A. Base product**
   - **B. Major variations of base product (for substantially different environments, platforms, or configurations)**
   - **C. Frequently used supersystems of base product or variations**
Activities of SRE Process and Relation to Software Development Process

1. List Associated Systems
2. Implement Operational Profiles
3. Engineer “Just Right” Reliability
4. Prepare for Test
5. Execute Test
6. Guide Test

Operation

Operation: major system logical task performed for initiator, which returns control to system when complete.

Illustrations - FF:
Process fax call, Phone number entry, Audit section of phone number database

Operational Profile

Operational profile (OP): complete set of operations with probabilities of occurrence

Illustration - FF:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Occur.</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process voice call, no pager, ans.</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>Process voice call, pager, ans.</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>Process fax call</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>Process voice call, pager, ans. on page</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Develop Operational Profiles – Step by Step

Develop operational profile for base product and each variation (supersystems have same operational profiles as those of their related base product or variations).
Develop Operational Profiles – Step by Step

1. Identify initiators of operations [1,3]
2. Create operations list [1,3]
3. Review operations list [1,3]
4. Determine occurrence rates [1,3]
5. Determine occurrence probabilities [1,3]

Steps 1, 2, 3 are mostly the same across base product and variations. New release often requires only slight change from previous release, all steps.

Apply Operational Profiles

Apply operational profile and criticality information to increase efficiency of:
1. Development of all developed software
2. Test of all associated systems

Apply Operational Profiles to Increase Development Efficiency

For developed software in base product and each variation:
1. Look for alternatives or eliminate low use, noncritical operations (Reduced Operation Software or ROS)
2. Select operations where looking for opportunities for reuse will be most cost-effective
3. Plan a more competitive release strategy (operational development)

Operational Development - Illustration

Proportion of operations developed
Proportion of use represented

Release 1
Release 2
Release 3
Apply Operational Profiles to Increase Development Efficiency

4. Allocate development resources:
   A. Among operations - for system engineering, architectural design, requirements reviews, design reviews
   B. Among modules - for code, code reviews, unit test ([3], pp. 121 - 122)

Apply Operational Profiles to Increase Test Efficiency

1. Distribute new test cases of release among new operations of base product and variations (Prepare for Test activity)
2. Invoke test based on operational profile (Prepare for Test and Execute Test activities)

Activities of SRE Process and Relation to Software Development Process

1. Define the Product
   2. Implement Operational Profiles
   3. Engineer “Just Right” Reliability
   4. Prepare for Test
   5. Execute Test
   6. Guide Test

Failure and Fault

Failure
Departure of system behavior in execution from user needs

Fault
Defect in system implementation that causes the failure when executed

User-oriented
Developer-oriented
Engineer “Just Right” Reliability - Step by Step

1. Define failure consistently over life of product release and clarify with examples [1,3]
2. Choose common reference measure for all failure intensities [1,3]
3. Set system failure intensity objective (FIO) for each associated system [1,3]
4. For any software you develop:
   A. Find developed software FIO [1,3]
   B. Choose software reliability strategies to meet developed software FIO and schedule objectives with lowest development cost [1,3]

Prepare for Test

1. Specify new test cases for new operations
   A. Distribute new test cases to new operations based on operational profile [1,3]
      Illustration - FF:
      Allocate 17% of test cases to Proc. fax call operation
   B. Detail new test cases for each new operation by selecting from possible choices of input variable values with equal probability [1,3]
      Illustration - FF:
      Forwardee = Local calling area
2. Specify test procedure, based on the test operational profile [1,3]
Execute Test

1. Determine and allocate test time among associated systems and types of test (feature, load, regression) [1,3]
2. Invoke test based on test operational profile [1,3]
3. Identify system failures and when they occurred - use data in Guide Test [1,3]

Invoke Test

- Base Product
- Supersystems

\[ F \text{ = Feature test} \]
\[ R \text{ = Regression test} \]

Certify reliability
Track reliability growth

Guide Test

Process system failure data gathered in test to:
1. Track reliability growth of developed software of base product and variations
2. Certify reliability of:
   A. Base product and variations that customers will acceptance test
   B. Supersystems
3. Guide product release
Track Reliability Growth

1. Execute CASRE software reliability estimation program to obtain FI / FIO ratio
2. Plot FI / FIO ratio against time
3. Interpret plot

Interpret Plot : Illustration - FF

When FIO not reached on schedule, practical solutions are:
1. Defer features
2. Rebalance major quality characteristics
3. Increase work hours

Possible causes:
1. Poor change control
2. Poor control of test execution, resulting in test selection probabilities varying in time

Investigate significant upward trends
**Interpret Plot: Illustration - FF**

Terminate test at FI/FIO = 0.5 (allows for estimation error)

**Certify Reliability – Using Reliability Demonstration Chart**

Fail. No. | Mcalls at Failure | Normalized Units
---|---|---
1 | 0.00375 | 0.75
2 | 0.00625 | 1.25
3 | 0.025 | 5

Failure intensity objective: 200 failures / Mcalls

**SRE and You**

1. SRE gives you a powerful way to engineer software-based products so you can be confident in the availability and reliability of the product you deliver as you deliver it in minimum time with maximum efficiency.
2. With SRE you control the process; it doesn’t control you.
3. SRE is a vital skill for being competitive.

**To Explore Further**

1. More Reliable Software Faster and Cheaper, classroom or distance learning course: [http://members.aol.com/JohnDMusa/](http://members.aol.com/JohnDMusa/)
2. SRE website: The essential guide to software reliability: [http://members.aol.com/JohnDMusa/](http://members.aol.com/JohnDMusa/)
   A. SRE Orientation (overviews of different lengths)
   B. Courses (classroom and distance learning)
   C. Consulting information
   D. Practitioners’ Corner (extensive user experiences with SRE and important application examples, advice on deploying SRE, comprehensive standards information)
To Explore Further

E. Resources for Everyone (download free failure intensity estimation program CASRE, join free SRE professional organization, access SRE Network, view conference information, learn from Question of the Month, use glossary)

F. Researchers’ Corner (access to failure interval data and enormous debugging history archive, access to comprehensive lists of open source projects likely to have free access to all kinds of data)

G. Professors’ Corner (how to teach SRE, slides and material for SRE courses, network to other professors teaching SRE)


5. Musa, J.D., More Reliable Software Faster and Cheaper. Overview of SRE for managers and anyone wanting a fast, broad understanding of the topic. Download from SRE Website [2]. (Click on “Overview”)


8. SRE Network: Communicate by email with hundreds of people interested in field. See SRE Website [2].