Interim Report, IEEE Working Group 982.1, Measures of the Software Aspects of Dependability

Robert V. Binder
Chair, IEEE 982.1 Working Group
Overview

982.1–202x Measures of the Software Aspects of Dependability

IEEE – Institute of Electrical and Electronics Engineers  https://ieee.org

Background
- Motivation
- Goals
- Work to date
- Changes and improvements

Key questions
- What is Dependability?
- How is it different than …
- Can it be measured?
- Pre- versus post-release?

This presentation is snapshot of work in progress.
All concepts, terms, etc. are subject to change.
What the process seemed like to me
An ideal definition would make it possible to answer questions like

- Is system X more dependable than system Y?
- How has system X’s dependability changed over time?
- What can we do to make system X more (less) dependable?
- How do we know system X is dependable?

Such a definition should make clear how dependability is different from:

- Trustworthiness
- Safety
- Security
IEEE 982.1-2005 defines *Dependability* as

[The] trustworthiness of a computer system such that reliance can be justifiably placed on the service it delivers. Reliability, availability, and maintainability are aspects of dependability. (adapted from Lyu [B12])


- Owing to ambiguities and the circularity of this definition, and to support international harmonization, the 982.1 working group is considering adopting the IEC definition.
What is Software Dependability?

**dependability, <of an item>:** ability to perform as and when required

Note 1 to entry: Dependability includes availability (192-01-23), reliability (192-01-24), recoverability (192-01-25), maintainability (192-01-27), and maintenance support performance (192-01-29), and, in some cases, other characteristics such as durability (192-01-21), safety and security.

Note 2 to entry: *Dependability is used as a collective term for the time-related quality characteristics of an item (emphasis added)*

What is Software Dependability?

Recall the key questions:

1) Is system X more dependable than system Y?
2) How has system X’s dependability changed over time?
3) What can we do to make system X more (less) dependable?
4) How do we know system X is dependable?

Neither definition of Dependability can provide a concise answer

- Both definitions are *rubrics* – neither defines Dependability as a measureable quality attribute.
- Why can’t we quantify Dependability?
A Duty Cycle model was developed to experiment with quantification.

Basic idea: set a budget for total cycle time and time to spend in each mode. Use the ratio of mode time to total cycle time as the weight for the corresponding quality attribute.

### Can Dependability be Quantified?

<table>
<thead>
<tr>
<th></th>
<th>OFFLINE</th>
<th>READY</th>
<th>ACTIVE</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interval (1 year)</strong> hrs</td>
<td></td>
<td></td>
<td></td>
<td>8760.0</td>
</tr>
<tr>
<td><strong>Operational Budget</strong> hrs</td>
<td>120</td>
<td>7688</td>
<td>952</td>
<td>8760.0</td>
</tr>
<tr>
<td><strong>Dependability Threshold</strong></td>
<td>0.90000</td>
<td>0.99999</td>
<td>0.99000</td>
<td></td>
</tr>
<tr>
<td><strong>Anti-D Tolerance</strong></td>
<td>0.1</td>
<td>0.00001</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td><strong>Dependability Threshold, hrs</strong></td>
<td>108.0</td>
<td>7687.9</td>
<td>942.5</td>
<td>8738.4</td>
</tr>
<tr>
<td><strong>Dependability Budget</strong></td>
<td>1.233%</td>
<td>87.762%</td>
<td>10.759%</td>
<td></td>
</tr>
<tr>
<td><strong>D-Factor Weight</strong></td>
<td>1.370%</td>
<td>87.763%</td>
<td>10.868%</td>
<td>100.000%</td>
</tr>
<tr>
<td><strong>Actual Hours</strong> hrs</td>
<td>300</td>
<td>7570</td>
<td>890</td>
<td>8760.0</td>
</tr>
<tr>
<td><strong>Actual MAR Metric</strong></td>
<td>?</td>
<td>0.98466</td>
<td>0.94432</td>
<td></td>
</tr>
</tbody>
</table>
In a word, No.

Problems

• Variability and scale of development (“big” $M$) confounds results
• Use “small” $m$, limited to file refresh without upstream effort?
• Restrict model to availability and reliability?

...a weighted average approach lacks statistical meaning. Specifically, there is no textbook method by which one can compute variance, confidence intervals, or related statistical inferences for these metrics. – WG member comment
A **Quality Attribute** is a property of a tangible object that may be measured using an **Operational Definition**.

- Availability and Reliability are measurable quality attributes.

A **Quality Outcome** is a result defined with one or more quality attributes.

- It is an abstract category for characterizing related system effects that have practical importance to interested parties.
- Typically not quantifiable.
### Dependability

**Dependability is the ability to perform as and when required.**

<table>
<thead>
<tr>
<th>WHEN</th>
<th>Operating Conditions</th>
<th>Functionality</th>
<th>Safety</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>* maintainable</td>
<td>* Intended use does not harm interested parties</td>
<td>* No data spills Hack resistant Tamper resistant</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Available</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Correct, Timely Resilient</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nominal</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Off-Nominal</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Malicious</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Degraded</td>
<td></td>
<td>?</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

**WHEN**

**Operating Conditions**

**AS** – required behavior in test or field
All Quality Outcomes derived from same primary data sources

Quality Outcomes

- Trustworthiness
- Dependability
- Safety
- Security

Pre-Release (Development) measurements
- Intrinsic SOI Indicators
  - Artifacts
  - Activity
  - Risks
  - Competence
  - Provenance

Post-Release (Field Operation) measurements
- Measureable SOI Behavior
  - Maintainability
  - Availability
  - Reliability
  - Performance
  - Efficiency
  - ...

- SOI Usage & Environment
  - Nominal
  - Off Nominal
  - Malicious
  - Degraded

SOI – System of Interest

Common Data Model

Testing of Trustworthy Systems
Pre-release analysis and prognostication

Quality Outcomes

- Trustworthiness
- Dependability
- Safety
- Security

Assurance Cases

Dependability Cases

Safety Cases

Security Cases

Intrinsic Indicators

- Static Analyzers
- Process Metrics

Measureable Behavior

- Testing
- Reliability Growth

Usage Characterization

- Market/Threat Studies
- Operational Profile

<Outcome> Case: • claim • argument • evidence

Common Data Model
Towards Comparing Dependabilities

- System X
- System Y

- Maintainability
- Availability
- Functionality
- Performance
- Utilization
- Safety Incidents/Exposures
- Exploited Vulnerabilities/LOC
- Latent Vulnerabilities/LOC

- Build 1024
- Build 2048

Testing of Trustworthy Systems

#UCAAT
982.1 Working Group highlights to date

- Leaning towards IEC "ability to perform as and when required"
  - "As" – measureable quality attributes with threshold and target values
  - "When" – typical and adverse operational profiles
- Introduced the idea of a “Quality Outcome” to distinguish abstract categories from measurable properties.
  - Dependability is a Quality Outcome
  - Quality Outcomes bundle quantity attributes
- “Little m” Maintainability (field updates, no-code re-configuration)
- Operational profiles for typical and adverse conditions are necessary for adequate assessment of pre-release dependability
- A common data model for pre- and post-release items will facilitate devops performance

This presentation is snapshot of work in progress.
All concepts, terms, etc. are subject to change.
What the process seems like now
THE TRUTH IS OUT THERE
Thank you!
Any questions?

rbinder@ieee.org