Software Reliability Engineering

Karama Kanoun
Karama.Kanoun@laas.fr

Research Group on Dependable Computing and Fault Tolerance
Toulouse, France
OUTLINE

- Motivations
- Methods for software reliability engineering
  - Data collection and analysis
    - Data collection and validation
    - Descriptive statistics
    - Trend analysis
  - Software dependability evaluation
    - Reliability growth models
    - Models in stable reliability
    - Controlled experimentation
- The maturity Process
- Case studies
- References
Why Software Reliability Engineering?

- Increasing role of software in real life systems
- System dependability is more and more synonymous of software reliability
- Difficulties in mastering the software development process and in reducing design faults for complex systems
- Increasing cost of system non-dependability
- Real needs for improving software reliability to improve system dependability and reduce maintenance cost
- Dependability requirements are part of system requirements (as important as functional requirements)
- Quantification is essential
Objectives of software reliability engineering

Short term

- Manage and improve the reliability of the software
- Check the efficiency of development activities
- Estimate the software reliability at the end of validation activities and in operation
- Estimate the maintenance effort to “correct” faults activated during development and residual faults in operation

Long term

- Capitalize experience
- Improve software reliability of successive generations

**Needs for experimental & analytical methods and techniques to reach these objectives**
Software vs hardware reliability

**Hardware**
- Physical faults
- Operational life
- Stable reliability (constant failure rate)
- White-box approach
- Markov models
- Database for components failures

**Software**
- Only design faults
- Development and operation
- Reliability growth (↓ failure rate)
- Usually black-box approach
- Specific models
- Based on data collection
Objectives of Software Reliability Engineering

_supplier point of view_

• During development:
  - development follow up
    (failure intensity, fault density)
  - evaluation of software reliability before operation
    (MTTF, pre-operational failure rate)

• During operation
  - product reliability follow up
    (residual failure rate, MTTF)
  - maintenance planning
    (cumulative number of failures)

Users / customers, operational life

  be confident in the reliability level of the product
  (residual failure rate, MTTF)
Difficulties

- Non-repetitive process
- No relationship between failures and corrections
- Continuous evolution of usage profile
  - According to the development phase
  - Within each phase
- Overselling of reliability growth models
- Judgement on quality of the software developers
- What is software reliability?
  - Residual number of faults, fault density, complexity measures?
  - MTTF, failure intensity, failure rate?
Measures

Static Measures of the product and process (quality oriented)

Usage profile (environment)

Dynamic Measures characterizing occurrence of failures (reliability oriented)

Number of faults
Fault density
Complexity measures
...

Failure intensity
Failure rate
MTTF
Reliability
...

ReSIST courseware — Karama Kanoun — Software Reliability Engineering
### Example:
Percentage of faults and corresponding MTTF (published by IBM)

<table>
<thead>
<tr>
<th>MTTF (years)</th>
<th>5000</th>
<th>1580</th>
<th>500</th>
<th>158</th>
<th>50</th>
<th>15.8</th>
<th>5</th>
<th>1.58</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>34,2</td>
<td>34,3</td>
<td>33,7</td>
<td>34,2</td>
<td>34,2</td>
<td>32,0</td>
<td>34,0</td>
<td>31,9</td>
</tr>
<tr>
<td></td>
<td>28,8</td>
<td>28,0</td>
<td>28,5</td>
<td>28,5</td>
<td>28,5</td>
<td>28,2</td>
<td>28,5</td>
<td>27,1</td>
</tr>
<tr>
<td></td>
<td>17,8</td>
<td>18,2</td>
<td>18,0</td>
<td>18,7</td>
<td>18,4</td>
<td>20,1</td>
<td>18,5</td>
<td>18,4</td>
</tr>
<tr>
<td></td>
<td>10,3</td>
<td>9,7</td>
<td>8,7</td>
<td>11,9</td>
<td>9,4</td>
<td>11,5</td>
<td>9,9</td>
<td>11,1</td>
</tr>
<tr>
<td></td>
<td>5,0</td>
<td>4,5</td>
<td>6,5</td>
<td>4,4</td>
<td>4,4</td>
<td>5,0</td>
<td>4,5</td>
<td>6,5</td>
</tr>
<tr>
<td></td>
<td>2,1</td>
<td>3,2</td>
<td>2,8</td>
<td>2,0</td>
<td>2,9</td>
<td>2,1</td>
<td>2,7</td>
<td>2,7</td>
</tr>
<tr>
<td></td>
<td>1,2</td>
<td>1,5</td>
<td>1,4</td>
<td>0,3</td>
<td>1,4</td>
<td>0,8</td>
<td>1,4</td>
<td>1,4</td>
</tr>
<tr>
<td></td>
<td>0,7</td>
<td>0,7</td>
<td>0,4</td>
<td>0,1</td>
<td>0,7</td>
<td>0,3</td>
<td>0,6</td>
<td>1,1</td>
</tr>
</tbody>
</table>
Overview of a global reliability analysis method

Development Validation & Operation

Data collection → Collected data

Data Validation

Validated data

Data set partition

Types of faults

Impact of failures

Phase Components

Descriptive Analyses

Trend Analyses

Model Application

Descriptive Statistics

Reliability Evolution

Reliability Measures

Objectives of the analysis

Data related to similar previous projects

Feedback to software development process

Capitalize experience
Setting up of a data collection process

Some rules

• Define clearly the objectives and the data to be collected
• Motivate and imply people that will be involved
• Simplify the collection process and reduce the number of data items to be collected
  - Support tools
  - Practical organization of people involved
• Record and analyze data in real-time
• Feedback

Origin of collected data

• Internal: recorded during development and validation
• External: by the customers
Data to be collected

- Background information
  - Product itself: software size, language, functions, current version, workload
  - Usage environment: verification and validation methods, tools, etc.
- Data relative to failures and corrections
  - Date of occurrence, nature of failures, consequences
  - Type of faults, fault location
- Usually, recorded through
  - Failure Reports (FR)
  - Correction Reports (CR)
- Well defined headings, well structured, easy to fill in
- Short tick-off questions
- Manually or automatically
Failure Report (FR)

Required Information

• Serial number (for identification)
• Report editor
• Product reference, version affected (or prototype)
• Date and time of failure occurrence

Desirable Information

• Failure occurrence condition
• Failure criticality or consequences
• Affected function or task
• Action proposed (if any)
Correction Report (CR)

Required information

- Serial number (for identification)
- Report editor
- Date of correction
- Correction nature
- Product reference
- Reference to the FR

Desirable Information

- Identification of the modified components

Integration with already existing data collection programs

Importance of training
Data Validation

Objectives

• check the validity and usability of the information recorded
• Keep only genuine software faults in the database

Elimination of:

• Duplicated data (FR reporting of the same failure)
• FR proposing a correction related to an already existing FR (COR)
• False FR (signalling a false or non identified problem)
• FR proposing an improvement (IMPROVE)
• incomplete FRs or FRs containing inconsistent data (Unusable)
• FR related to a hardware failure
• …
Example 1: a telecommunications equipment
(analyzed at LAAS)

2,146 Failure Reports

Validation ⇒ 1,172 kept in the database

Discarded RFs:

<table>
<thead>
<tr>
<th>Discarded RFs</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicated FRs</td>
<td>816</td>
<td>38.0%</td>
</tr>
<tr>
<td>COR</td>
<td>53</td>
<td>2.5%</td>
</tr>
<tr>
<td>False FR</td>
<td>29</td>
<td>1.4%</td>
</tr>
<tr>
<td>IMPROVE</td>
<td>21</td>
<td>1.0%</td>
</tr>
<tr>
<td>Unusable</td>
<td>20</td>
<td>0.9%</td>
</tr>
<tr>
<td>Hardware</td>
<td>35</td>
<td>1.6%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>974</strong></td>
<td><strong>45.4%</strong></td>
</tr>
</tbody>
</table>
Example 2: a telephone switching system
(analyzed at LAAS)

- 3063 FRs
- Validation → 1853 Software FRs kept in the database
- Discarded RFs:

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>195</td>
<td>(6%)</td>
</tr>
<tr>
<td>Documentation</td>
<td>165</td>
<td>(5%)</td>
</tr>
<tr>
<td>Unusable, duplicated, …</td>
<td>716</td>
<td>(24%)</td>
</tr>
<tr>
<td>Others</td>
<td>134</td>
<td>(4%)</td>
</tr>
<tr>
<td>Total</td>
<td>1210</td>
<td>(39%)</td>
</tr>
</tbody>
</table>
Life cycle of Failure and Correction Reports (FRs/CRs)

1. Identification of an abnormal behavior
2. Interface with users
3. Database
4. FR exists?
   - Yes: Already solved or being solved
   - No: Creation of an FR
5. Analysis & validation
6. Correction Proposal?
   - Yes: Implementation of the corrections
   - No: Report update
7. FR resolved
8. Creation of a CR
9. Specialized team
DESCRIPTIVE STATISTICS

Aim: make syntheses of the observed phenomena

Simple analyses
- Fault typology
- Fault density of components
- Failure / fault distribution among software components (new, modified, reused)

Investigation of relationships
- Fault density / size / complexity
- Fault density / life cycle phase
- Nature of faults / life cycle phases
- Nature of faults / components
- Number of components affected by changes made to resolve an FR

Analyses related to the development / debugging process
Analyses related to the development process

Factors affecting time to locate and solve problems

- The more FRs circulating, the more time it takes to handle each one
- Tendency to resolve the easier FRs first, the remaining ones take more time
- Loss of maintainability with continued changes to resolve faults
- Introduction of new faults while resolving the old

Average time to resolve an FR

Modification request time =

\[ \text{Time when the FR is resolved} - \text{time when it is created} \]

Measures

- Responsiveness of the field support system
- Complexity of maintenance
Case of the switching system of Example 2
Data pre-processing for reliability analysis

Two kinds of data sets can be extracted from FRs and CRs

- Time to failures (or between failures)

\[ t_k = \text{time between failure } k-1 \text{ and } k \]

- Grouped data

  - Number of failures per unit of time, \( n(k) \)
  - Cumulative number of failures \( N(k) \)

\[ n(1) \leq n(k) \leq 2 \]
Time?

- Time between failures
  - Execution time
  - Wall clock or Calendar time
  - Number of executions

- Number of failures per unit of time
  - The length of the unit time depends on:
    - accuracy expected for the dependability measures
    - number of observed failures
    - objectives of the study
**Example A: Times between failures**

Real-time control system (Musa 1)

- 136 failures observed during system test (96 days)

<table>
<thead>
<tr>
<th>#</th>
<th>T1</th>
<th>Dy</th>
<th>#</th>
<th>T1</th>
<th>Dy</th>
<th>#</th>
<th>T1</th>
<th>Dy</th>
<th>#</th>
<th>T1</th>
<th>Dy</th>
<th>#</th>
<th>T1</th>
<th>Dy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>25</td>
<td>422</td>
<td>31</td>
<td>49</td>
<td>816</td>
<td>56</td>
<td>73</td>
<td>810</td>
<td>64</td>
<td>97</td>
<td>261</td>
<td>72</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>2</td>
<td>26</td>
<td>180</td>
<td>32</td>
<td>50</td>
<td>1351</td>
<td>56</td>
<td>74</td>
<td>290</td>
<td>64</td>
<td>98</td>
<td>1800</td>
<td>73</td>
</tr>
<tr>
<td>3</td>
<td>113</td>
<td>9</td>
<td>27</td>
<td>10</td>
<td>32</td>
<td>51</td>
<td>148</td>
<td>56</td>
<td>75</td>
<td>300</td>
<td>64</td>
<td>99</td>
<td>865</td>
<td>73</td>
</tr>
<tr>
<td>4</td>
<td>81</td>
<td>10</td>
<td>28</td>
<td>1146</td>
<td>33</td>
<td>52</td>
<td>21</td>
<td>57</td>
<td>76</td>
<td>529</td>
<td>65</td>
<td>100</td>
<td>1435</td>
<td>74</td>
</tr>
<tr>
<td>5</td>
<td>115</td>
<td>11</td>
<td>29</td>
<td>600</td>
<td>34</td>
<td>53</td>
<td>233</td>
<td>57</td>
<td>77</td>
<td>281</td>
<td>65</td>
<td>101</td>
<td>30</td>
<td>74</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>11</td>
<td>30</td>
<td>15</td>
<td>42</td>
<td>54</td>
<td>134</td>
<td>57</td>
<td>78</td>
<td>160</td>
<td>65</td>
<td>102</td>
<td>143</td>
<td>74</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>17</td>
<td>31</td>
<td>36</td>
<td>42</td>
<td>55</td>
<td>357</td>
<td>57</td>
<td>79</td>
<td>828</td>
<td>66</td>
<td>103</td>
<td>108</td>
<td>74</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>20</td>
<td>32</td>
<td>4</td>
<td>46</td>
<td>56</td>
<td>193</td>
<td>59</td>
<td>80</td>
<td>1011</td>
<td>66</td>
<td>104</td>
<td>0</td>
<td>74</td>
</tr>
<tr>
<td>9</td>
<td>112</td>
<td>20</td>
<td>33</td>
<td>0</td>
<td>46</td>
<td>57</td>
<td>236</td>
<td>59</td>
<td>81</td>
<td>445</td>
<td>66</td>
<td>105</td>
<td>3110</td>
<td>75</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>20</td>
<td>34</td>
<td>8</td>
<td>46</td>
<td>58</td>
<td>31</td>
<td>59</td>
<td>82</td>
<td>296</td>
<td>66</td>
<td>106</td>
<td>1247</td>
<td>76</td>
</tr>
<tr>
<td>11</td>
<td>138</td>
<td>20</td>
<td>35</td>
<td>227</td>
<td>46</td>
<td>59</td>
<td>369</td>
<td>59</td>
<td>83</td>
<td>1755</td>
<td>67</td>
<td>107</td>
<td>943</td>
<td>76</td>
</tr>
<tr>
<td>12</td>
<td>50</td>
<td>20</td>
<td>36</td>
<td>65</td>
<td>46</td>
<td>60</td>
<td>748</td>
<td>59</td>
<td>84</td>
<td>1064</td>
<td>67</td>
<td>108</td>
<td>700</td>
<td>76</td>
</tr>
<tr>
<td>13</td>
<td>77</td>
<td>20</td>
<td>37</td>
<td>476</td>
<td>46</td>
<td>61</td>
<td>0</td>
<td>59</td>
<td>85</td>
<td>1783</td>
<td>68</td>
<td>109</td>
<td>875</td>
<td>77</td>
</tr>
<tr>
<td>14</td>
<td>24</td>
<td>20</td>
<td>38</td>
<td>58</td>
<td>46</td>
<td>62</td>
<td>232</td>
<td>59</td>
<td>86</td>
<td>860</td>
<td>68</td>
<td>110</td>
<td>245</td>
<td>77</td>
</tr>
<tr>
<td>15</td>
<td>108</td>
<td>20</td>
<td>39</td>
<td>457</td>
<td>47</td>
<td>63</td>
<td>330</td>
<td>59</td>
<td>87</td>
<td>983</td>
<td>68</td>
<td>111</td>
<td>729</td>
<td>77</td>
</tr>
<tr>
<td>16</td>
<td>88</td>
<td>20</td>
<td>40</td>
<td>300</td>
<td>47</td>
<td>64</td>
<td>365</td>
<td>61</td>
<td>88</td>
<td>707</td>
<td>69</td>
<td>112</td>
<td>4897</td>
<td>78</td>
</tr>
<tr>
<td>17</td>
<td>670</td>
<td>30</td>
<td>41</td>
<td>97</td>
<td>47</td>
<td>65</td>
<td>1222</td>
<td>62</td>
<td>89</td>
<td>33</td>
<td>69</td>
<td>113</td>
<td>447</td>
<td>79</td>
</tr>
<tr>
<td>18</td>
<td>120</td>
<td>30</td>
<td>42</td>
<td>263</td>
<td>47</td>
<td>66</td>
<td>543</td>
<td>63</td>
<td>90</td>
<td>868</td>
<td>69</td>
<td>114</td>
<td>386</td>
<td>79</td>
</tr>
<tr>
<td>19</td>
<td>26</td>
<td>30</td>
<td>43</td>
<td>452</td>
<td>53</td>
<td>67</td>
<td>10</td>
<td>63</td>
<td>91</td>
<td>724</td>
<td>69</td>
<td>115</td>
<td>446</td>
<td>79</td>
</tr>
<tr>
<td>20</td>
<td>114</td>
<td>30</td>
<td>44</td>
<td>255</td>
<td>53</td>
<td>68</td>
<td>16</td>
<td>63</td>
<td>92</td>
<td>2323</td>
<td>70</td>
<td>116</td>
<td>122</td>
<td>79</td>
</tr>
<tr>
<td>21</td>
<td>325</td>
<td>30</td>
<td>45</td>
<td>197</td>
<td>54</td>
<td>69</td>
<td>529</td>
<td>64</td>
<td>93</td>
<td>2930</td>
<td>71</td>
<td>117</td>
<td>990</td>
<td>79</td>
</tr>
<tr>
<td>22</td>
<td>55</td>
<td>30</td>
<td>46</td>
<td>193</td>
<td>54</td>
<td>70</td>
<td>379</td>
<td>64</td>
<td>94</td>
<td>1461</td>
<td>72</td>
<td>118</td>
<td>948</td>
<td>80</td>
</tr>
<tr>
<td>23</td>
<td>242</td>
<td>31</td>
<td>47</td>
<td>6</td>
<td>54</td>
<td>71</td>
<td>44</td>
<td>64</td>
<td>95</td>
<td>843</td>
<td>72</td>
<td>119</td>
<td>1082</td>
<td>80</td>
</tr>
<tr>
<td>24</td>
<td>68</td>
<td>31</td>
<td>48</td>
<td>79</td>
<td>54</td>
<td>72</td>
<td>129</td>
<td>64</td>
<td>96</td>
<td>12</td>
<td>72</td>
<td>120</td>
<td>22</td>
<td>80</td>
</tr>
</tbody>
</table>
### Example B

**Number of failures per unit of time or Cumulative**

#### Switching system

- 52 failures in operation (15 months)

<table>
<thead>
<tr>
<th>i: unit of time (week)</th>
<th>n(i): number of failures per unit of time</th>
<th>NC(i): cumulative number of failures</th>
<th>NS(i): number of systems in operation at i</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>n(i)</td>
<td>NC(i)</td>
<td>NS(i)</td>
</tr>
<tr>
<td>---</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>18</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>21</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>21</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>22</td>
<td>21</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>22</td>
<td>21</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>25</td>
<td>28</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>25</td>
<td>28</td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>29</td>
<td>28</td>
</tr>
</tbody>
</table>
Trend analysis

Objectives:

- Analyze software reliability evolution
- Identify periods of reliability growth and decrease

[See references 9 or 10]
Reliability growth characterization

- Variable: time to failure
  - $T_1, T_2, \ldots, T_n$: time between failure $i$ and $i-1$
  - Reliability growth: $T_i \leq T_k \forall \ i < k$
  - Prob. $\{T_i < x\} \geq$ Prob. $\{T_k \leq x\} \Rightarrow F_{T_i}(x) \geq F_{T_k}(x) \forall \ i < k \forall x$

- Variable: number of failures
  - $N(t_1), N(t_2), \ldots, N(t_n)$: cumulative number of failures between 0 and $t_i$
  - $H(t_i) = E[N(t_i)] = \text{expectation of } N(t_i)$
  - If $N(t_i)$ is a Non Homogeneous Poisson Process (NHPP):
    - Reliability growth if $H(t_1) + H(t_2) \geq H(t_1 + t_2) \forall \ t_1, t_2 \geq 0$ and $0 \leq t_1 + t_2 \leq T$
    (inequality is strict for at least a pair $t_1, t_2$)

$N(t)$ is a subadditive function
Interpretation of Subadditivity

\[ H(t_1) + H(t_2) \geq H(t_1 + t_2) \quad \forall \ t_1, t_2 \geq 0 \text{ and } 0 \leq t_1 + t_2 \leq T \]

The number of events in an interval of the form \([0, t_2]\) is larger than the number of events taking place in an interval of the same length beginning later (i.e. in the form of \([T, T+t_2]\). The number of failures is decreasing

Graphical interpretation

- \( H(t) = E[N(t)] \) is subadditive over \([0,T]\) if:

\[
a_H(t) = \int_0^t H(x) \, dx - \frac{t}{2} H(t) \geq 0
\]

\( \forall \ t \geq 0 \) and \( 0 \leq t \leq T \)

\( a_H(t) = \text{subadditivity factor} \)
Trend tests

Means

- Raw data $\Rightarrow$ graphical tests
- Analytical tests $\Rightarrow$ quantitative indicators

Raw data

- Times to successive failures
- Number of failures per unit of time
- Cumulative number of failures

Trend indicators

- Empirical (arithmetical) means
- Subadditivity factor
- Laplace factor
Graphical tests: times to failures (Example A)

Times to failures
$t_i$

Cumulative times to failures
$t_1 + \ldots t_k$
Graphical test: grouped data (Example B)

Failure intensity

Cumulative number of failures

unit of time = one week

unit of time = four weeks

ReSIST courseware — Kamaa Kanoun — Software Reliability Engineering
Empirical mean

Global trend

$\xi_k : \text{arithmetical mean of the times to failures (from failure 1 to k)}$

$$\xi_k = \frac{t_1 + t_2 + \ldots + t_k}{k}$$

$\xi_k$ constitute a globally increasing series $\Rightarrow$ reliability growth

$\xi_k$ constitute a globally decreasing series $\Rightarrow$ reliability decrease

The trend is directly observed on the evolution of $\xi_k$
Empirical mean

Local trend

- The data items are grouped into subsets containing m successive data
- The average is evaluated for each subset
- The impact of old data items is eliminated

Example A: m = 8 ⇒ 17 groups (136 failures)
Subadditivity & monotonous growth / decrease

Monotonous growth:
\[ a_H(x) > 0 \text{ increasing} \]

Monotonous decrease:
\[ a_H(x) < 0 \text{ decreasing} \]
Subadditivity & trend change

Decrease - Growth

Growth - Decrease

$H(x)$

$\frac{d}{dx}H(x) > 0$

$\frac{d}{dx}H(x) < 0$

$a_{\gamma}(x) > 0$

$a_{\gamma}(x) < 0$

$T_L$

$T_G$

$T$
Subadditivity & local trend fluctuations

Example: reliability growth with local fluctuations

\[ a_{xt}(x) \geq 0 \text{ non decreasing} \]
Laplace factor

Statistical Test of hypothesis \( \Rightarrow \) Laplace factor \( u \)

Random variable: times to failures \( T_i \) (realization of \( T_i = t_i \))

\[
u(T) = \left( \frac{1}{N(T)} \sum_{i=1}^{N(T)} \sum_{j=1}^{i} t_j - \frac{T}{2} \right) \frac{T}{\sqrt{\frac{1}{12} N(T)}}
\]

- \( \frac{T}{2} \) = mid of the observation interval
- \( c = \frac{1}{N(T)} \sum_{i=1}^{N(T)} \sum_{j=1}^{i} t_j \) = statistical centre

In practice

\( u > 0 \) \( \Rightarrow \) global reliability decrease

\( u < 0 \) \( \Rightarrow \) global reliability growth
• Random variable: # failures per unit of time

\[ u(T) = \frac{\sum_{i=1}^{k} (i-1) n(i) - \frac{k-1}{2} \sum_{i=1}^{k} n(i)}{\sqrt{\frac{k^2 - 1}{12} \sum_{i=1}^{k} n(i)}} \]

\( n(i) = \) # failure during time unit \( i \)

• Can be put in the form:

\[ u(T) = - \frac{a_{\mu T}(T)}{T \sqrt{\frac{1}{12} N(T)}} \]
Laplace factor: local and global trend

\[ u(k) \]

- Global trend: Reliability decrease
- Local trend: Reliability growth
- Local trend changes

- Global trend changes

- Reliability decrease
- Reliability growth
- Reliability decrease
Change of time origin

\[ u(k) \]

\[ T_{L1} \quad T_G \quad T_{L2} \]

\[ k \]

\[ A \quad B \quad C \quad D \]

\[ B - C \quad D \]
Link: graphical tests - Laplace - Subadditivity

- Failure intensity
- Subadditivity factor
- Cumulative number of failures
- Laplace factor
# Link between trend indicators

<table>
<thead>
<tr>
<th>Cumulative number of failures</th>
<th>Failure intensity</th>
<th>Laplace factor</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Graph" /></td>
<td><img src="image2" alt="Graph" /></td>
<td><img src="image3" alt="Graph" /></td>
</tr>
<tr>
<td><strong>Monotonous Growth</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image4" alt="Graph" /></td>
<td><img src="image5" alt="Graph" /></td>
<td><img src="image6" alt="Graph" /></td>
</tr>
<tr>
<td><strong>Monotonous decrease</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image7" alt="Graph" /></td>
<td><img src="image8" alt="Graph" /></td>
<td><img src="image9" alt="Graph" /></td>
</tr>
<tr>
<td><strong>Decrease followed by growth</strong></td>
<td><img src="image10" alt="Graph" /></td>
<td><img src="image11" alt="Graph" /></td>
</tr>
<tr>
<td><img src="image12" alt="Graph" /></td>
<td><img src="image13" alt="Graph" /></td>
<td><img src="image14" alt="Graph" /></td>
</tr>
<tr>
<td><strong>Stability</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image15" alt="Graph" /></td>
<td><img src="image16" alt="Graph" /></td>
<td><img src="image17" alt="Graph" /></td>
</tr>
</tbody>
</table>
How to use trend test results

Control of the efficiency of test activities

- Reliability decrease at the beginning of a new activity: OK
- Reliability decrease during a relatively long period of time: Pb ?
- Reliability growth after reliability decrease: OK
- Sudden reliability growth: caution!
- Stable reliability: saturation

- New tests
- Following phase
- End of test

Application of reliability models

- Trend in accordance with model assumptions
### Application to RADC data sets

**Rome Air Development Center (USA)**

<table>
<thead>
<tr>
<th>System Id.</th>
<th># instructions</th>
<th># programmers</th>
<th># failures</th>
<th>Type of system</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21 700</td>
<td>9</td>
<td>136</td>
<td>RT.C</td>
</tr>
<tr>
<td>2</td>
<td>27 700</td>
<td>5</td>
<td>54</td>
<td>RT.C</td>
</tr>
<tr>
<td>3</td>
<td>23 400</td>
<td>3</td>
<td>38</td>
<td>RT.C</td>
</tr>
<tr>
<td>4</td>
<td>33 500</td>
<td>6</td>
<td>54</td>
<td>RT.C</td>
</tr>
<tr>
<td>5</td>
<td>2 445 000</td>
<td>7</td>
<td>831</td>
<td>RT.Com.</td>
</tr>
<tr>
<td>6</td>
<td>5 700</td>
<td>275</td>
<td>73</td>
<td>RT.C</td>
</tr>
<tr>
<td>7 (14C)</td>
<td>******</td>
<td>8</td>
<td>36</td>
<td>military</td>
</tr>
<tr>
<td>8(17)</td>
<td>61 900</td>
<td>110</td>
<td>38</td>
<td>military</td>
</tr>
<tr>
<td>9(27)</td>
<td>126 100</td>
<td>8</td>
<td>41</td>
<td>military</td>
</tr>
<tr>
<td>10(40)</td>
<td>180 000</td>
<td>8</td>
<td>101</td>
<td>military</td>
</tr>
<tr>
<td>11A(SS1A)</td>
<td>******</td>
<td>8</td>
<td>112</td>
<td>OS</td>
</tr>
<tr>
<td>11B (SS1B)</td>
<td>******</td>
<td>unknown</td>
<td>375</td>
<td>OS</td>
</tr>
<tr>
<td>11C (SS1C)</td>
<td>******</td>
<td>unknown</td>
<td>277</td>
<td>OS</td>
</tr>
<tr>
<td>12 (SS2)</td>
<td>******</td>
<td>unknown</td>
<td>192</td>
<td>TS</td>
</tr>
<tr>
<td>13 (SS3)</td>
<td>******</td>
<td>unknown</td>
<td>278</td>
<td>WP</td>
</tr>
<tr>
<td>14 (SS4)</td>
<td>******</td>
<td>unknown</td>
<td>196</td>
<td>OS</td>
</tr>
</tbody>
</table>

RT: Real-time  
C: control  
Com.: commercial  
WP: word Processing  
TS: Time sharing  
OS: Operating system  
******: not given
### Laplace factor

<table>
<thead>
<tr>
<th>System</th>
<th>Laplace factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-9.10</td>
</tr>
<tr>
<td>2</td>
<td>-5.73</td>
</tr>
<tr>
<td>3</td>
<td>-6.13</td>
</tr>
<tr>
<td>4</td>
<td>-8.59</td>
</tr>
<tr>
<td>6</td>
<td>-3.64</td>
</tr>
<tr>
<td>7</td>
<td>-2.14*</td>
</tr>
<tr>
<td>8</td>
<td>-4.65</td>
</tr>
<tr>
<td>9</td>
<td>-5.16</td>
</tr>
<tr>
<td>10</td>
<td>-9.60</td>
</tr>
<tr>
<td>11A</td>
<td>-1.36*</td>
</tr>
<tr>
<td>11B</td>
<td>-0.73*</td>
</tr>
<tr>
<td>11C</td>
<td>-5.15</td>
</tr>
<tr>
<td>12</td>
<td>+0.74*</td>
</tr>
<tr>
<td>13</td>
<td>-5.64</td>
</tr>
<tr>
<td>14</td>
<td>-1.78*</td>
</tr>
</tbody>
</table>

* : stable reliability
System 2: times to failures

Time to failures, ti

# failures
System 2: Laplace factor

Variable: time to failure
System 2: failure intensity

\[ n(k) \]

\( k \): unit of times = 5000 seconds of execution time
**System 2: Laplace factor**

**Variable:** # failures

**Graph 1:**
- Failure #31
- Failure #41
- $u(k)$ vs. $k = \text{unit of time}$

**Graph 2:**
- $u(k)$ vs. $k = \text{unit of time}$

**Unit of time = 5000 seconds of execution time**
System 4: arithmetical mean

![Graph showing the arithmetical mean](image)
System 9

Arithmetical mean

Laplace Factor

# failures

ReSIST courseware — Karama Kanoun — Software Reliability Engineering
System 14

Arithmetical mean

Laplace factor

ReSIST courseware — Karama Kanoun — Software Reliability Engineering
Conclusion

Some systems can be modeled by an exponential distribution

- System for which $-2 < u < 2$

Impact of the operational profile

- Systems 11 A, B, C are 3 copies of the same program used in different environments

Benefits from trend analysis

- Understanding of the underlying processes
- Follow up of the development process in real-time, fast feedback
- Helpful for reliability model application
SOFTWARE RELIABILITY EVALUATION

Objectives

- Evaluate measures characterizing the software reliability and its evolution

Methods

- Evaluation from data collected on the software during testing and / or operation
  - With fault removal
  - Without fault removal

Reliability growth models

Measures:

- Failure rate
- Failure intensity
- Cumulative number of failures
- MTTF

Models in stable reliability

Measures:

- Test duration without failure, required to reach a target reliability
- Probability of accepting/rejecting a piece of software
- Probability of failure in operation

[See reference 1]
OUTLINE

Reliability growth models
  • Presentation of some reliability growth models
  • Reliability growth models and trend analysis
  • Application of reliability growth models
  • Tools

Models in stable reliability

Other approaches
RELIABILITY GROWTH MODELS

Modeling difficulties

- Corrections + specification changes ⇒ varying behavior
  ⇒ absence of repetitive phenomenon ⇒ absence of statistics

- Variations in the usage environment

- No direct relationships between failures and corrections

Objectives of reliability growth models:

Estimation of dependability measures as resulting from the above variations
⇒ restrictive assumption for some models: correction after each failure
RELIABILITY GROWTH MODELS

Failure rate models
(Failure rate equations & relationship between successive failure rates)
- Deterministic, piecewise Poisson Process models: Jelinski Moranda, Musa
- Stochastic, doubly stochastic process model: Littlewood-Verrall

Failure intensity models: succession of failures
(based on Non-Homogeneous Poisson Process (NHPP))
- Exponential model (Goel Okumoto)
- Hyperexponential model (Kanoun-Laprie)
- S-Shaped model (Yamada et al)

Selection depends on
- Objectives
  Development follow-up, evaluation of operational MTTF and residual failure rate
- Trend displayed by the data set
Jelinski Moranda model: assumptions

First software reliability model (1972)

Assumptions

- **H1**: The total number of faults is finite \( N_0 \)
- **H2**: No fault introduction while correcting detected faults: each activated fault is corrected before new executions
- **H3**: Faults are independent and their manifestation rate is constant
- **H4**: Inputs are selected randomly and tests are representative of operational profile
- **H5**: All failures are observed
Jelinski Moranda model: equations

Parameters

\[ N_0 = \text{total number of faults} \]
\[ \Phi = \text{fault manifestation rate} \]
\[ \lambda(i) = \text{failure rate of the i-th failure} \]
\[ T_i = \text{random variable: time between failures i-1 and i (observation = ti)} \]

Relations

\[ \lambda(i) = \Phi [N_0 - (i - 1)] = \frac{di}{dt} \quad i = 1, 2, \ldots, N_0 \]

\[ \text{Prob. (} T_i < t_i \text{)} = \Phi (N_0 - i + 1). \exp \{ \Phi (N_0 - i + 1).t_i \} \]

\[ MTTF_i = \frac{1}{\lambda(i)} = \frac{1}{\Phi [N_0 - (i - 1)]} \]

\[ N(t) = N_0 [1 - \exp (-\Phi t)] = \text{number of faults detected at t} \]

Parameters to be estimated: \( N_0, \Phi \)
Jelinski-Moranda model: $\lambda(t)$

- The failure rate is constant and tends to 0 when $t$ tends to $\infty$
Musa model

Assumptions similar to the Jelinski-Moranda model

Parameters definition

\( M_0 = \text{number of faults in the software} \)
\( N_0 = \text{number of failures} \)
\( B = \text{fault reduction factor: number of faults / number of failures} \quad M_0 = B \cdot N_0 \)
\( C = \text{compression factor (execution time in operation / in test)} \)
\( \Phi = \text{fault manifestation rate} \)

Relations

\[ \lambda(i) = B \cdot C \cdot \Phi \cdot (N_0 - i + 1) \quad \text{MTTF}(i) = \frac{1}{B \cdot \Phi \cdot (N_0 - i + 1)} \]

\( N(t) = N_0 \left[ 1 - \exp\left(-B \cdot C \cdot \Phi \cdot t\right) \right] = \text{number of failures observed at } t \text{ (execution time)} \)

Parameters to be estimated: \( N_0, \Phi \) (B product characteristics; C operational profile)
Littlewood-Verrall model

Stochastic relationship between the successive failure rates

Distinction

- Input uncertainty: \( \lambda_i \)
- Impact of corrections uncertainty: \( \lambda_1, \lambda_2, \ldots, \lambda_i \) series of random variables

Randomness of inputs

\[ f(T_i | \lambda_i) = \lambda_i \exp(-\lambda_i t) \]

\( f \): probability density function (pdf)

\( T_i \): time to failure i since failure i-1 (time to failure i)

Impact of corrections

\[ f(\lambda_i | \alpha, \Psi) = \frac{[\Psi(i)]^\alpha \lambda_i^{\alpha-1} \exp(\Psi(i)) \lambda_i}{\Gamma(\alpha)} \]

\( \Psi \): programmer skill and programming difficulty
Littlewood-Verrall model

Distribution of $T_i$

\[
f(\ t_i|\alpha, \Psi) = \int_0^\infty f(t_i|\lambda_i). f(\lambda_i|\alpha, \Psi) \ d\lambda_i = \frac{\alpha [\Psi(i)]^\alpha}{[t_i + \Psi(i)]^{\alpha+1}}
\]

Reliability growth represented by growth of $\Psi(i)$:

\[
\Psi(i) = \beta_1 + \beta_2 \cdot i
\]

\[
\lambda_i(t) = \frac{\alpha}{t + \Psi(i)}
\]

\[
MTTF_i = \frac{\Psi(i)}{\alpha - 1}
\]

Parameters: $\alpha$, $\beta_1$, $\beta_2$

<table>
<thead>
<tr>
<th>curve</th>
<th>$\alpha$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>1</td>
<td>30</td>
<td>100</td>
</tr>
<tr>
<td>C2</td>
<td>3</td>
<td>60</td>
<td>150</td>
</tr>
<tr>
<td>C3</td>
<td>4</td>
<td>80</td>
<td>100</td>
</tr>
</tbody>
</table>
NHPP models

Based on Non-homogeneous Poisson Process (NHPP)

• Definition

\[ P\{ N(t+dt) - N(t) = 1 \} = h(t) \, dt \]
\[ P\{ N(t+dt) - N(t) \geq 2 \} = o(dt) \]

\([ N(t_0) ], [N(t_1) - N(t_0) ], \ldots, [N(t_n) - N(t_{n-1})] \), \( t_0 < t_1 < \ldots < t_n \)
are random variables with independent increments

• Properties

number of events on \([t_1, t_2]\)

\[ E[N(t_2) - N(t_1)] = \int_{t_1}^{t_2} h(t) \, dt = H(t_2) - H(t_1) \]

Prob. \{N(t) = n | N(t_0) = n_0\} = \frac{[H(t) - H(t_0)]^{n-n_0}}{(n - n_0)!} \exp\{- [H(t) - H(t_0)]\}
\exp \{ - [H(t) - H(t_0)] \}
n > n_0 \text{ and } t > t_0
Exponential Model (EXP)

Failure intensity

\[ h(t) = a \cdot b \cdot \exp(-bt) \]

parameters to be estimated: \( a, b \)

Cumulative number of failures

\[ H(t) = a \left[ 1 - \exp(-bt) \right] \]
Hyperexponential Model (HE)

Failure intensity

\[ h(t) = \frac{\omega \zeta_{\text{sup}} e^{-\zeta_{\text{sup}} t} + \omega \zeta_{\text{inf}} e^{-\zeta_{\text{inf}} t}}{\omega e^{-\zeta_{\text{sup}} t} + \omega e^{-\zeta_{\text{inf}} t}} \]

\[ 0 \leq \omega \leq 1, \; \omega + \omega = 1 \text{ and } \zeta_{\text{inf}} \leq \zeta_{\text{sup}} \]

\[ \zeta_{\text{inf}} = \text{residual failure rate} \]
Hyperexponential Model (HE)

\[ \begin{align*}
H(t) &= E[N(t)] = -\ln \left[ \omega e^{-\zeta_{\text{sup}} t} + \omega e^{-\zeta_{\text{inf}} t} \right] \\
\text{MTTF} &= \frac{\omega \zeta_{\text{sup}} e^{-\zeta_{\text{sup}} s} + \omega \zeta_{\text{inf}} e^{-\zeta_{\text{inf}} s}}{\omega e^{-\zeta_{\text{sup}} s} + \omega e^{-\zeta_{\text{inf}} s}} \\
\lambda(t' | s) &= h(s+t) \\
s &= \text{time of occurrence of failure } i
\end{align*} \]

Parameters to be estimated: \( \omega, \zeta_{\text{inf}}, \zeta_{\text{sup}} \)
S-Shaped model (SS)

Failure intensity

\[ h(t) = a b^2 t \exp(-bt) \]

parameters to be estimated: \( a, b \)

Cumulative number of failures

\[ H(t) = a \left[ 1 - (1 + b t) \exp(-bt) \right] \]
Model in practice

- Pre-processing of failure data
  - Trend analysis \( \Rightarrow \) reliability growth ?
- Parameter determination from observed failure data
  - Inference procedures
- Prediction of next failure(s)
  - Evaluation of reliability measures based on observed data
- Model validation \( \Rightarrow \) confidence in evaluation
  - Checking agreement between Predictions / Observations
    - Predictive analysis
    - Retrodictive analysis
Model application: predictive analysis

OBSERVATIONS

Time to failures, # failures

Trend tests

Validation criteria

PREDICTIONS

MTTF

Failure rates

Failure intensity

Cumulative number of failures

Model application

Numerical values of the parameters

Measure values
Model application: retrodictive analysis

Validation criteria

Observations

Time to failures, # failures

Trend tests

Predictions

MTTF

Failure rates

Failure intensity

Cumulative number of failures

Model application

Numerical values of the parameters

Measure values
Trend tests & models

- Trend test: identification of periods of reliability growth / decrease

- Reliability growth models are selected depending on the trend displayed by the observed data set

<table>
<thead>
<tr>
<th>Failure intensity</th>
<th>Applicable models</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="n(k)" alt="n(k)" /> grown</td>
<td><strong>Models with reliability growth</strong></td>
</tr>
<tr>
<td><img src="n(k)" alt="n(k)" /> stable</td>
<td>Models in stable reliability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Failure intensity</th>
<th>Applicable models</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="n(k)" alt="n(k)" /> decreased</td>
<td>Models with reliability decrease followed by reliability growth</td>
</tr>
</tbody>
</table>
Combined use in real-time of trend tests & models

Identify the trend

Apply an appropriate model

Trust model results as long as the usage conditions are not modified
  • Test of the same function(s)
  • No addition of new users or new sites
  • No specification changes

In case of significant variation
  • Apply the trend test including the new data items:
    - Reliability growth: trust the previous estimations
    - Reliability decrease: wait for reliability growth
    - Reliability growth after reliability decrease: new data partitioning

and application of reliability growth models
<table>
<thead>
<tr>
<th>Unit tests</th>
<th>Static Verification</th>
<th>End of Validation</th>
<th>Operation</th>
</tr>
</thead>
</table>
| Trend analysis | Trend analysis + Reliability growth models | Limits: 10^{-3}/h - 10^{-4}/h | Trend analysis + Reliability growth models or models in stable reliability

**Operational profile?**

**Enough data?**

**High relevance**

**Examples:**

- E10-B (Alcatel ESS):
  - 1400 systems, 3 years
  - \( \lambda = 5 \times 10^{-6}/h \quad \lambda_c = 10^{-7}/h \)

- ABB Atom Nuclear I&C Appli.
  - 8000 systems, 4 years
  - \( \lambda : 3 \times 10^{-7} / h \quad \lambda_c = 4 \times 10^{-8}/h \)
Conclusion

Method

- Rigorous progressive analysis of the software behavior
- Deep thoughts about the system and the analyzed data
- Better results from reliability growth models

Applicability

- General method: applicable to hardware design faults
- Should be integrated to the various phases of the development:
  - early phases: analyses of data and trend tests
  - validation and operational life: application of models (in addition)

The method has been applied to several real-life systems
(hardware and software)

Needs for tools
Example of Tool: SoRel (developed at LAAS)

Trend Tests
- Arithmetical mean
- Kendall test
- Laplace test
- Spearman test

Models
- Hyperexponential (Kanoun-Laprie)
- Exponential (Goel-Okumoto)
- S-Shaped (Yamada et al.)
- Doubly Stochastic (Littlewood-Verrall)

[See reference 6]
## Experience with SoRel

<table>
<thead>
<tr>
<th>System</th>
<th>Languages</th>
<th>Volume</th>
<th>Observation</th>
<th>Phases</th>
<th># Systems</th>
<th># FR and/or CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>E10-B</td>
<td>Assembler</td>
<td>100 k-bytes</td>
<td>3 years</td>
<td>Val. / Op.</td>
<td>1400</td>
<td>58 FR / 136 CR</td>
</tr>
<tr>
<td>TROPICO-R 1500</td>
<td>Assembler</td>
<td>300 k-bytes</td>
<td>27 months</td>
<td>Val. / Op.</td>
<td>15</td>
<td>465 FR/CR</td>
</tr>
<tr>
<td>TROPICO-R 4096</td>
<td>Assembler</td>
<td>350 k-bytes</td>
<td>32 months</td>
<td>Val. / Op.</td>
<td>42</td>
<td>210 FR/CR</td>
</tr>
<tr>
<td>TROPICO-RS</td>
<td>Assembler</td>
<td>420 k-bytes</td>
<td>47 months</td>
<td>Op.</td>
<td>37</td>
<td>212 FR/CR</td>
</tr>
<tr>
<td>TROPICO-RA</td>
<td>CHILL</td>
<td>815 KLOC</td>
<td>68 months</td>
<td>Val. / Op.</td>
<td>146</td>
<td>3063 FR/CR</td>
</tr>
<tr>
<td>Telecom. Equipt</td>
<td>PLM-86</td>
<td>$5 \times 10^5$ inst.</td>
<td>16 months</td>
<td>Val.</td>
<td>4</td>
<td>2150 FR</td>
</tr>
</tbody>
</table>
MODELS IN STABLE RELIABILITY

- Apply when no program evolution nor failure resolution is occurring
- Operational testing (end of validation) — certification
  or when the system is in operation without fault correction
- Residual faults: expected to induce a reduced failure rate
- Two types of inferences
  - Experiments without failures:
    Hypothesis testing evaluate a lower bound on the software reliability
    or an upper bound on the failure probability (for a given confidence level)
  - Experiments with only a few failures observed (all known faults are not fixed)
    1) Hypothesis testing (assessment of lower bounds) or
    2) Evaluation of an unbiased estimator of the failure probability per execution
    (the first approach is better when the number of failures is very low)
Reliability evaluation when testing reveals no failure

Hypothesis testing when testing reveals no failure

\[ \text{Prob } \{ \text{accepting } "p \leq p_0" \text{ while it is false } \} \leq \alpha \]

- \( p \) = actual probability of failure and \( p_0 \) required probability of failure (objectives)
- \( \alpha \) = risk error and \( (1- \alpha) \) = confidence level

Amount of execution / time required

- \( N \) = number of executions without failure,
- \( T \) = test duration without failure

\[ N \geq \frac{\ln (\alpha)}{\ln (1-p_0)} \]

(results from \( (1 - p)^N < \alpha \))

- Discrete time:

\[ T \geq - \frac{\ln (\alpha)}{\lambda_0} \]

Continuous time:
Measure:

Test duration without failure, required to reach a target reliability objective

**Discrete time**: Number of program executions without failure

<table>
<thead>
<tr>
<th>Target failure probability $p_0$</th>
<th>Risk: $\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$10^{-1}$</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>23</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>230</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>2303</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>23026</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>230259</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>2302585</td>
</tr>
</tbody>
</table>
Measure:

Test duration without failure, required to reach a target reliability objective

**Continuous time**: Testing times for some values of $\lambda_0$ and $\alpha$

<table>
<thead>
<tr>
<th>Target failure rate $\lambda_0$</th>
<th>Risk $\alpha$</th>
<th>Time unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>$10^{-1}$</td>
<td>Days</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>$10^{-2}$</td>
<td>Months</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>Years</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>$10^{-5}$</td>
<td></td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$10^{-1}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
<th>$10^{-6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>10</td>
<td>20</td>
<td>1</td>
<td>1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>3.2</td>
<td>6.4</td>
<td>9.6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>2.6</td>
<td>5.3</td>
<td>7.9</td>
<td>10.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>26.2</td>
<td>52.3</td>
<td>78.9</td>
<td>105.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>262.8</td>
<td>525.7</td>
<td>788.6</td>
<td>1051.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Other example: stable reliability in operation

Problem

- The software system is in operation, some failures have been observed, their consequences are acceptable, even if the faults have been identified
- Modifications are not performed or, only a few modifications are introduced without perception of any reliability growth / decrease

Aim

Evaluate the operational failure rate

Method

- Constant failure rate, Homogeneous Poisson Process $\Rightarrow$ Markov process
- Average observed MTTF, associated confidence level
- Usually good results
Product-in-a-process approach

Supplement current approaches to software reliability evaluation with information

- Process
- Past field experience

**Product-in-a-process assessment**

Validation of a product = validation of (n+1)th product
with information about: ITSELF + PREVIOUS PRODUCTS

**Framework**: Bayesian probabilities

- $\theta$: conditional probability of failure upon execution / failure rate
- Prior and posterior distributions: conjugate distributions
  - Beta distribution / Gamma distribution

[See reference 4]
\[ \theta = k_1 \theta_c + k_2 \theta_p \quad k_1 + k_2 = 1 \]

\( \theta \) point Bayesian estimate

\( \theta_c \) conventional estimate (validation of the product in isolation)

\( \theta_p \) prior estimate (field experience of previous products)

Field produce much more data than validation of new software

\( k_2 > k_1 \Rightarrow \) prior estimate dominates conventional estimate

Example:

Satellite control system

\( \theta_c = 11.6 \ 10^{-3}/h \) (6 months)

\( \theta_p = 2.8 \ 10^{-3}/h \) (21 months)

\( k_1 = 0.2 \); \( k_2 = 0.8 \)

\[ \Rightarrow \ \theta_p = 4.7 \ 10^{-3}/h \]

\[ \Rightarrow \text{observed (17 months): } 3.9 \ 10^{-3}/h \]
Conclusion

- Software systems under development and in operation
- With fault removal \(\Rightarrow\) Reliability growth models
  - For several reasons reliability decrease
    (new specifications, environment change, new usage profile, etc.)
  - Identify the trend before model application
  - Good results under certain conditions, for short term objectives
  - Long term objectives ? other new approaches (product-in-a-process approach)
- Without fault removal \(\Rightarrow\) stable reliability
  - Some of the work related to statistical testing could be adapted to operation
  - Two situations: with a few failures or without failures
  - Limitations due to prohibitive test time needed to achieve high reliability objectives
  - Interesting when several systems are under use (example of avionics systems)
  - Test acceleration methods
Off-the shelf software components
Dependability benchmarking

- No information available from component development
- Evaluation based on controlled experimentation

- Ad hoc
- Standard

Dependability benchmarking
Evaluation of dependability measures / features
in a non-ambiguous way → comparison

Properties
Reproducibility, repeatability, portability, representativeness, acceptable cost

[See reference 12]
Context: User point of view

**Computer System**

**Operating System**

Which OS for my computer system?

- Limited knowledge: functional description
- Limited accessibility and observability
- Limited intrusiveness and interference

⇒ Black-box approach ⇒ robustness benchmark
Operating System Benchmarking and Associated Measures

Faults = corrupted parameters of system calls

Measures
- POS: OS Robustness [\%SEr  \%SXP  \%SPc  \%SHg  \%SNS]
- Texec: OS reaction time in the presence of faults
- Tres: OS Restart time after fault insertion
Experimental setup

Host Machine

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

API

Target Operating System

Hardware

System under benchmarking

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)

Interception & Substitution of system calls & Observation OS reaction

Control Machine

Target Operating System

Hardware

API

Activity (Workload)
Measurements

Experiments with Workload completion

Workload Completion Time
OS Reaction time
Restart time

tWStart (n)
tResume (n)
tResponse (n)
tExpEnd (n)
tExpStart (n+1)

System Call to intercept
Workload End
Experiments without Workload completion

Timeout >> Workload completion duration

Experiment End

OS Reaction time

Restart time

System Call to intercept

Workload End

$tWStart (n)$

$tResume (n)$

$tResponse (n)$

$tExpEnd (n)$

$tExpStart (n+1)$
OS reaction time (Workload = PostMark)

Windows

Linux

- In the presence of faults
- Without parameter corruption
OS Restart time

Windows

Linux

Graphs showing OS restart time for Windows and Linux with different versions. Each graph has two bars for each version:
- Red bar: In the presence of faults
- Green bar: Without parameter corruption

ReSIST courseware — Karama Kanoun — Software Reliability Engineering
Detailed OS Restart times

**Windows XP**

**Linux 2.2.26**

- Workload Abort/hang
- check disk
Summary

Development Validation & Operation

Data collection

Collected data

Data Validation

Validated data

Data set partition

Types of faults

Consequences of failures

Components

Objectives of the analysis

Feedback to software development process

Descriptive Analyses

Descriptive Statistics

Trend Analyses

Reliability Evolution

Model Application

Reliability Measures

Capitalize experience

Objective of the analysis

Data related to similar previous projects

ReSIST courseware — Karama Kanoun — Software Reliability Engineering
SOFTWARE PROCESS IMPROVEMENT (SPI) (The maturity process)

- To obtain consistent quality of the software
  ⇒ control the production process ⇒ improve the software process

- The engineering method:
  • Observe existing solutions
  • Propose better solutions
  • Build / develop
  • Measure and analyze
  • Repeat the process until no more improvements possible
  ⇒ evolutionary / continuing improvement oriented approach

Models for process maturity or organization maturity
Aim: assess the organization maturity level

[See reference 11]
Some existing methods / models

- Crosby: *Satisfaction by Quality Scheme to software development*

- Weinberg: The *Software Engineering Culture Patterns*

- Humphrey: *A Maturity Framework* ⇒ The Capability Maturity Model

Other approaches:

- AT&T: *Quality Program*

- Fujitsu: *Concurrent-Development Process Model*

- IBM: *The Cleanroom Software Development Process*

- IBM Communication Systems: *The Defect Prevention Process*

- ODC (Orthogonal Defect Classification)

- etc.
Cost and reliability evolution, taking into account process improvement

Cost

Total manufacturing cost

Basic manufacturing cost

Cost of reliability

Cost of scrap / rework

Reliability

without process improvement

with process improvement

PROCESS IMPROVEMENT

RELIABILITY IMPROVEMENT

COST REDUCTION
Example of benefits from SPI introduction

- **IBM** (cleanroom approach):
  Productivity increase = 70% for development and 100% for testing

- **IBM** (defect prevention approach):
  Fault density divided by 2 with an increase of 0.5 % of the product resources

- **Fujitsu** (concurrent development process):
  Release cycle reduction = 75 %

- **AT&T** (quality program):
  Customer reported problems divided by 10
  Maintenance program divided by 10
  System test interval divided by 2
  New product introduction interval divided by 3
  Importance of operational profile (principal cost in SRE): ≠ test efficiency
Example of benefits from SPI introduction (Cont’d)

☞ Raytheon (Electronic Systems), CMM:
- Rework cost divided by 2 after two years of experience
- Productivity increase = 190%
- Product quality: multiplied by 4

☞ Raytheon (Equipment Division), CMM:
- Rework cost divided by 4 (elimination of $15.8 million in rework cost)
- Productivity multiplied by 2
- Return on investment 7.7-to-1

☞ Hughes Aircraft (Software Engineering Division, Fullerton CA):
- 1987: level 2 ⇒ recommendations & actions ⇒ level 3 in 1990
- Return on investment of process improvement initiative: 5-to-1

☞ Motorola (Arlington Heights), mix of methods:
- Fault density reduction = 50 within 3.5 years
CASE STUDIES

- TROPICO-R 1500 [See reference 3]
  Reliability analysis and evaluation

- TROPICO-R 4096 [See reference 7]
  Software decomposition
  Reliability analysis and evaluation

- Three generations of TROPICO-R [See reference 8]
  Comparative evolution: fault density and reliability
TROPICO-R 1500

**Characteristics**
- Language: Assembly
- Size: 300 k-bytes
- Validation: 10 months, 297 failures / corrections
- Field trial: 4 months, 55 failures/corrections
- Operation: 13 months, 109 failures/corrections
- Total: 461

**Data**
- Number of failures / unit of time
  - unit of time: 10 days
  - observation duration: 81 units of times
- Times to failures
  - operational life only
## Data set

<table>
<thead>
<tr>
<th>Validation</th>
<th>Field test</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>u. time</td>
<td>u. time</td>
<td>u. time</td>
</tr>
<tr>
<td></td>
<td>CNF</td>
<td>CNF</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>45</td>
<td>34</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>74</td>
<td>36</td>
</tr>
<tr>
<td>7</td>
<td>82</td>
<td>37</td>
</tr>
<tr>
<td>8</td>
<td>98</td>
<td>38</td>
</tr>
<tr>
<td>9</td>
<td>106</td>
<td>39</td>
</tr>
<tr>
<td>10</td>
<td>115</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td>120</td>
<td>41</td>
</tr>
<tr>
<td>12</td>
<td>134</td>
<td>42</td>
</tr>
<tr>
<td>13</td>
<td>139</td>
<td>55</td>
</tr>
<tr>
<td>14</td>
<td>142</td>
<td>56</td>
</tr>
<tr>
<td>15</td>
<td>145</td>
<td>57</td>
</tr>
<tr>
<td>16</td>
<td>153</td>
<td>58</td>
</tr>
<tr>
<td>17</td>
<td>157</td>
<td>59</td>
</tr>
<tr>
<td>18</td>
<td>174</td>
<td>60</td>
</tr>
<tr>
<td>19</td>
<td>183</td>
<td>61</td>
</tr>
<tr>
<td>20</td>
<td>196</td>
<td>62</td>
</tr>
<tr>
<td>21</td>
<td>200</td>
<td>63</td>
</tr>
<tr>
<td>22</td>
<td>214</td>
<td>64</td>
</tr>
<tr>
<td>23</td>
<td>223</td>
<td>65</td>
</tr>
<tr>
<td>24</td>
<td>246</td>
<td>66</td>
</tr>
<tr>
<td>25</td>
<td>257</td>
<td>67</td>
</tr>
<tr>
<td>26</td>
<td>277</td>
<td>68</td>
</tr>
<tr>
<td>27</td>
<td>283</td>
<td>69</td>
</tr>
<tr>
<td>28</td>
<td>286</td>
<td>70</td>
</tr>
<tr>
<td>29</td>
<td>292</td>
<td>71</td>
</tr>
<tr>
<td>30</td>
<td>297</td>
<td>72</td>
</tr>
</tbody>
</table>

CNF: cumulative # of failures
Test de Laplace & data partitioning

Data Partitioning

- Validation & field trial
  - P1: \{1, 14\}
  - P2: \{15, 42\}

- Operation
  - P3: \{43, 54\}
  - P4: \{55, 81\}
Model Application
(Number of failures)

- Validation & field trial, application of the S-Shaped model

### Cumulative # of failures

<table>
<thead>
<tr>
<th>Unit of time</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>50</td>
<td>100</td>
<td>150</td>
<td>200</td>
<td>250</td>
<td>300</td>
<td>350</td>
<td>400</td>
</tr>
</tbody>
</table>

- C1: calibrated from {1,8}
- C2: calibrated from {15,27}
- C3: calibrated from {15,29}

<table>
<thead>
<tr>
<th></th>
<th>$R_{9,14}$</th>
<th>$R_{28,42}$</th>
<th>$R_{30,42}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>2,6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td>28,4</td>
<td>31,2</td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td>5,8</td>
</tr>
</tbody>
</table>
Model application
(Number of failures)

Operational life, application of the S-Shaped model (SS)

- Cumulative # of failures

![Graph showing cumulative number of failures for different systems over time.]

Residue

<table>
<thead>
<tr>
<th>System</th>
<th>$R_{51,55}$</th>
<th>$R_{74,81}$</th>
<th>$R_{76,81}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4</td>
<td>1,8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td>4,3</td>
<td>5,3</td>
</tr>
<tr>
<td>C6</td>
<td></td>
<td></td>
<td>3,5</td>
</tr>
</tbody>
</table>

Prediction for next quarter (all systems)

2 failures the next month
& 1 failure / month the next two months
Model application
(times to failures)

Operational life, average system, application of the Hyperexponential model

Laplace Test

Software residual failure rate for an average system

\[ \lambda_{\text{sof}} = 1.3 \times 10^{-4} / h \] (all consequences)

Hardware failure rate (known from a different study)

\[ \lambda_{\text{har}} = 4 \times 10^{-6} / h \] (leading to system unavailability)

apply reliability growth models to failures leading to total unavailability

\[ \lambda_{\text{har}} \ll \lambda_{\text{sof}} \]
Model application according to software components & to failure consequences

- Other switching system E-10-B
  - Hyperexponential model

<table>
<thead>
<tr>
<th>Component</th>
<th>$\lambda_r (10^{-7} / \text{h})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephony</td>
<td>7,5</td>
</tr>
<tr>
<td>Defense</td>
<td>27,4</td>
</tr>
<tr>
<td>Exploitation</td>
<td>7,3</td>
</tr>
<tr>
<td>Executive</td>
<td>8,3</td>
</tr>
<tr>
<td>All corrections</td>
<td>47,5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Consequence</th>
<th>$\lambda_r (10^{-7} / \text{h})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>General unavailability</td>
<td>1,2</td>
</tr>
<tr>
<td>Partial unavailability</td>
<td>7,9</td>
</tr>
<tr>
<td>Exploitation treat. delay</td>
<td>3,7</td>
</tr>
<tr>
<td>Loss of a hardware unit</td>
<td>3,1</td>
</tr>
<tr>
<td>All failures</td>
<td>38,2</td>
</tr>
</tbody>
</table>
TROPICO-R 4096

Characteristics
- Language: Assembly
- Size: 335 k-bytes
- Validation: 8 months, 76 failures/corrections
- Operation: 24 months, 134 failures/corrections
- Total: 210

Data
- Number of failures/unit of time
  - observation period: 32 months
- Times to failures
  - for operational life
Software decomposition and # of systems

Decomposition

<table>
<thead>
<tr>
<th>Component</th>
<th>Volume</th>
<th># failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephony</td>
<td>75 k-bytes</td>
<td>74 (34-40)</td>
</tr>
<tr>
<td>Defense</td>
<td>117 k-bytes</td>
<td>67 (20-47)</td>
</tr>
<tr>
<td>Interface</td>
<td>115 k-bytes</td>
<td>61 (20-41)</td>
</tr>
<tr>
<td>Management</td>
<td>44 k-bytes</td>
<td>31 (13-18)</td>
</tr>
</tbody>
</table>

Number of systems

![Graph showing the number of systems over time.](chart)
Laplace Test

Operation: all systems

Operation: average system

Validation
Laplace Test for the software components

Telephony

Interface

Defense

Management
Failure intensity: Hyperexponential model application
Residual failure rates (Hyperexponential model)

<table>
<thead>
<tr>
<th>Component</th>
<th>Residual failure rate</th>
<th>Size (Kb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephony</td>
<td>$1.2 \times 10^{-6}$ /h</td>
<td>75</td>
</tr>
<tr>
<td>Defense</td>
<td>$1.4 \times 10^{-5}$ /h</td>
<td>103</td>
</tr>
<tr>
<td>Interface</td>
<td>$2.9 \times 10^{-5}$ /h</td>
<td>115</td>
</tr>
<tr>
<td>Management</td>
<td>$8.5 \times 10^{-6}$ /h</td>
<td>42</td>
</tr>
<tr>
<td>Sum</td>
<td>$5.3 \times 10^{-5}$ /h</td>
<td>335</td>
</tr>
</tbody>
</table>

Observed failure intensity

Failure intensity estimated by HE
(Residual failure rate: $5.7 \times 10^{-5}$ /h)

Sum of the failure intensities of the components estimated by HE
Maintenance planning

Estimated # failures from 20 to 32:
- Exponential: 33
- Hyperexponential: 37
- S-Shaped: 9

Observed: 34
Maintenance planning

Estimated # failures from 20 to 32: 40

Observed: 34
Software Reliability Analysis of Three Successive Generations of a Switching System

Outline

• The products investigated
• Data collected
• Statistics on failures and faults
• Residual failure rates
• Conclusion
Products & Software

• Three products

  - TROPICO-R 1500 (PRA)
  - TROPICO-R 4096 (PRB)
  - TROPICO-RS (PRC)

• Software components (Applicative & Executive software)

  - Elementary Implementation Blocks (EIB)
  - Functions
    - Telephony (TEL)
    - Defense (DEF)
    - Interface (INT)
    - Management (MAN)
## Software decomposition and size

### PRA

<table>
<thead>
<tr>
<th>EIB</th>
<th>Size (Kbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEL</td>
<td>6 72</td>
</tr>
<tr>
<td>DEF</td>
<td>9 93</td>
</tr>
<tr>
<td>INT</td>
<td>10 113</td>
</tr>
<tr>
<td>MAN</td>
<td>4 42</td>
</tr>
<tr>
<td>Sum</td>
<td>29 320</td>
</tr>
</tbody>
</table>

### PRB

<table>
<thead>
<tr>
<th>EIB</th>
<th>Size (Kbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEL</td>
<td>6 75</td>
</tr>
<tr>
<td>DEF</td>
<td>12 117</td>
</tr>
<tr>
<td>INT</td>
<td>10 115</td>
</tr>
<tr>
<td>MAN</td>
<td>4 44</td>
</tr>
<tr>
<td>Sum</td>
<td>32 351</td>
</tr>
</tbody>
</table>

### PRC

<table>
<thead>
<tr>
<th>EIB</th>
<th>Size (Kbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEL</td>
<td>8 111</td>
</tr>
<tr>
<td>DEF</td>
<td>12 130</td>
</tr>
<tr>
<td>INT</td>
<td>10 129</td>
</tr>
<tr>
<td>MAN</td>
<td>4 51</td>
</tr>
<tr>
<td>Sum</td>
<td>34 421</td>
</tr>
</tbody>
</table>
Two types of EIBs: new — reused (modified / unchanged)

According to # of EIBs

PRB
- 50% (75% in Applicative Soft.)
- 13%
- 37% (84% in Executive)

PRC
- 18%
- 6%
- 76%

According to the size of EIBs

PRB
- 67%
- 12%
- 21%

PRC
- 34%
- 2%
- 64%
Test environment and failure data

• Software test program
  Steps: unit tests, integration tests, validation tests, field tests
  Validation tests: functional, quality, performance, overload tests

• Failure reports & Trouble reports (FRs & TRs)
  • Date of failure occurrence (static analysis ≠ date of detection)
  • Description of system configuration in which the failure was observed
  • Type: hardware, software, documentation, affected EIBs
  • Analysis: identification— classification of faults (coding, specification, etc.)
  • Solutions
  • Regression testing

• An FR is a failure report and also a correction report

• Rediscoveries are not recorded
Data Collection

# PRA systems

# PRB systems

# PRC systems

ReSIST courseware — Karama Kanoun — Software Reliability Engineering
Statistics on Failures and Faults

<table>
<thead>
<tr>
<th></th>
<th># FR (# TR)</th>
<th># CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRA</td>
<td>465</td>
<td>637</td>
</tr>
<tr>
<td>PRB</td>
<td>210</td>
<td>282</td>
</tr>
<tr>
<td>PRC</td>
<td>212 (105)</td>
<td>394</td>
</tr>
</tbody>
</table>

>70 % of failures led to modification of one EIB

<table>
<thead>
<tr>
<th># corrected EIBs</th>
<th># FR in PRA</th>
<th># FR in PRB</th>
<th>#FR+TR in PRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>362 (77.8%)</td>
<td>165 (78.6%)</td>
<td>228 (71.9%)</td>
</tr>
<tr>
<td>2</td>
<td>72 (15.5%)</td>
<td>33 (15.7%)</td>
<td>69 (21.8%)</td>
</tr>
<tr>
<td>≥ 3</td>
<td>31 (6.7%)</td>
<td>12 (5.7%)</td>
<td>20 (6.3%)</td>
</tr>
</tbody>
</table>

identify EIBs which are dependent w.r.t failure occurrence
(2 pairs of strongly dependent EIBs)
### Statistics on Failures and Faults (cont’d)

#### PRA

<table>
<thead>
<tr>
<th></th>
<th># FR</th>
<th># CF</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEL</td>
<td>146</td>
<td>190</td>
<td>72</td>
</tr>
<tr>
<td>DEF</td>
<td>138</td>
<td>164</td>
<td>93</td>
</tr>
<tr>
<td>INT</td>
<td>170</td>
<td>191</td>
<td>113</td>
</tr>
<tr>
<td>MAN</td>
<td>78</td>
<td>92</td>
<td>42</td>
</tr>
<tr>
<td>Sum</td>
<td>532</td>
<td>637</td>
<td>320</td>
</tr>
</tbody>
</table>

#### PRB

<table>
<thead>
<tr>
<th></th>
<th># FR</th>
<th># CF</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEL</td>
<td>74</td>
<td>102</td>
<td>75</td>
</tr>
<tr>
<td>DEF</td>
<td>67</td>
<td>71</td>
<td>117</td>
</tr>
<tr>
<td>INT</td>
<td>61</td>
<td>68</td>
<td>115</td>
</tr>
<tr>
<td>MAN</td>
<td>31</td>
<td>41</td>
<td>44</td>
</tr>
<tr>
<td>Sum</td>
<td>233</td>
<td>282</td>
<td>351</td>
</tr>
</tbody>
</table>

#### PRC

<table>
<thead>
<tr>
<th></th>
<th># FR (# TR)</th>
<th># CF</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEL</td>
<td>65 (52)</td>
<td>155</td>
<td>111</td>
</tr>
<tr>
<td>DEF</td>
<td>63 (21)</td>
<td>88</td>
<td>130</td>
</tr>
<tr>
<td>INT</td>
<td>72 (27)</td>
<td>112</td>
<td>129</td>
</tr>
<tr>
<td>MAN</td>
<td>25 (10)</td>
<td>40</td>
<td>51</td>
</tr>
<tr>
<td>Sum</td>
<td>225 (110)</td>
<td>395</td>
<td>421</td>
</tr>
</tbody>
</table>

† 90% of FRs led to modification of only one Function
Distribution of faults among functions

PRA
- 14%
- 30%
- 26%

PRC
- 10%
- 28%
- 40%
- 22%

PRB
- 15%
- 24%
- 25%
- 36%

Legend:
- TEL
- DEF
- INT
- MAN
Distribution of faults per EIB type

**PRB**
- 7% Unchanged
- 10% Modified
- 83% New

**PRC**
- 0% Unchanged
- 42% Modified
- 58% New
## Average fault density

### versus EIB size

<table>
<thead>
<tr>
<th>Size</th>
<th>PRA</th>
<th>PRB</th>
<th>PRC (all faults)</th>
<th>PRC (only faults relative to FRs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIB size &gt; 15 Kb</td>
<td>1.80</td>
<td>1.08</td>
<td>0.99</td>
<td>0.65</td>
</tr>
<tr>
<td>10 Kb &lt; EIB size &lt; 15 Kb</td>
<td>2.02</td>
<td>0.68</td>
<td>0.58</td>
<td>0.47</td>
</tr>
<tr>
<td>5 Kb &lt; EIB size &lt; 10 Kb</td>
<td>2.31</td>
<td>0.60</td>
<td>0.96</td>
<td>0.52</td>
</tr>
<tr>
<td>EIB size &lt; 5 Kb</td>
<td>2.56</td>
<td>0.71</td>
<td>0.62</td>
<td>0.56</td>
</tr>
<tr>
<td>Average fault density</td>
<td>2.10</td>
<td>0.76</td>
<td>0.79</td>
<td>0.55</td>
</tr>
</tbody>
</table>

### data collected during operation

<table>
<thead>
<tr>
<th></th>
<th>PRA</th>
<th>PRB</th>
<th>PRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>After 13 months</td>
<td>0.34</td>
<td>0.35</td>
<td>0.30</td>
</tr>
<tr>
<td>After 24 months</td>
<td>-</td>
<td>0.47</td>
<td>0.60</td>
</tr>
</tbody>
</table>
Fault density evolution / EIB type (Operation)

Unchanged EIBs

Modified EIBs

1 year

2 years
## Residual failure rates

<table>
<thead>
<tr>
<th></th>
<th>PRA</th>
<th>PRB</th>
<th>PRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEL</td>
<td>$2.6 \times 10^{-5}$ / h</td>
<td>$1.2 \times 10^{-6}$ / h</td>
<td>$4.3 \times 10^{-5}$ / h</td>
</tr>
<tr>
<td>DEF</td>
<td>$4.3 \times 10^{-5}$ / h</td>
<td>$1.4 \times 10^{-5}$ / h</td>
<td>$1.9 \times 10^{-5}$ / h</td>
</tr>
<tr>
<td>INT</td>
<td>$4.2 \times 10^{-5}$ / h</td>
<td>$2.9 \times 10^{-5}$ / h</td>
<td>$3.2 \times 10^{-5}$ / h</td>
</tr>
<tr>
<td>MAN</td>
<td>$1.4 \times 10^{-6}$ / h</td>
<td>$8.5 \times 10^{-6}$ / h</td>
<td>$9.9 \times 10^{-6}$ / h</td>
</tr>
<tr>
<td>Sum</td>
<td>$1.124 \times 10^{4}$ / h</td>
<td>$5.27 \times 10^{-5}$ / h</td>
<td>$1.03 \times 10^{-4}$ / h</td>
</tr>
</tbody>
</table>
Conclusion

- PRA & PRB
  - Similar development environment $\Rightarrow$ reliability improvement

- PRC
  - Learning process interrupted $\Rightarrow$ reliability improvement?

- PRA, PRB & PRC
  - Residual failure rates: same order of magnitude
  - Failure rate of the software = sum of the failure rates of its components

- Additional experimental studies
  $\Rightarrow$ factors impacting the reliability of a family of products
References


7. Experience in software reliability: from data collection to quantitative evaluation, K. Kanoun, M. Kaâniche, and J. C. Laprie, 4th International Symposium on Software Reliability Engineering, Denver (USA), 3-6 November 1993, pp.234-245.


