Tool-based Analysis of Space Domain Simulators

Tiago L. Alves

September 28, 2010
Background

- Highly specialized consultancy company for software quality and risks
- 10 years of experience helping clients to manage software
- Independent and therefore able to give objective advice
- Decorated with the Innovator Award 2007 and ICT Regie Award 2008

Activity

- Management advisory, fact-based
- Accredited software analysis lab employs analysis tools and models
- Experienced staff transforms analysis data into advice
Services

**Software Risk Assessment**
- In-depth investigation of software quality and associated business risks
- Answers to specific research questions

**Software Monitoring**
- Continuous measurement, feedback, and development consultancy
- Guard quality from start to finish

**Software Product Certification**
- Five levels of technical quality (maintainability)
- Evaluation by SIG, certification by TÜV Informationstechnik
Who is using our services?

Financial and Insurance companies
- ABN AMRO
- ING
- Rabobank
- ING
- LeasePlan
- Interpolis
- Bank Mendes Gans
- globalcollect
- Allianz

Government
- Rijksoverheid
- Kadaster
- Schweizerische Eidgenossenschaft
- Confederazione svizzera
- Confederazione svizzera
- Raad voor Rechtsbijstand
- Politie
- ProRail
- Port of Rotterdam
- RDW

Logistical
- DHL
- KLM
- TNT
- Euromaxx Terminal
- Norfolkline
- SNS Bank
- Volvo OtoFinans

IT
- Getronics Pink Roccade
- ENECO
- energie
- Capgemini
- Exact software
- IBM
- Gasunie
- Alcatel-Lucent
- Electrabel

Other
- PriceWaterhouseCoopers
- PriceWaterhouseCoopers
- PriceWaterhouseCoopers
- PriceWaterhouseCoopers
- PriceWaterhouseCoopers
Case Study - Space Domain Simulators
ESA SimSat and EuroSim

**ESA SimSat**
- Owned by ESA, developed in outsourcing (yearly bid process to choose company)
- Kernel implemented in C/C++
- GUI implemented in Eclipse RCP (Java)
- Development started in 2003
- Soft-real time (communications)

**EuroSim**
- Developed by a consortium (Dutch Space, NLR, TASK24)
- Implemented in C/C++ (interfaces in Java, Ada, Fortran, ...)
- Development started in 1997
- Hard-real time (hardware components)

1. Used to validate space sub-systems
2. Simulate Galileo components
3. Follow ECSS software process standards

Tiago L. Alves, Software Improvement Group / University of Minho
11th International Workshop on Simulation & EDGE Facilities for Space Programmes (SESP 2010)
Purpose of the talk

I. The use of quality ratings to support management

II. How quality ratings can be derived from measurements

III. How to measure software in a meaningful way
If you can’t measure it, you can’t manage it!

Tom De Marco, paraphrasing Lord Kelvin
The Bermuda Triangle of Software Quality

Quality standards and methodologies for process, people, projects!
But how about products?

Tiago L. Alves, Software Improvement Group / University of Minho
11th International Workshop on Simulation & EDGE Facilities for Space Programmes (SESP 2010)
Quality Assurance in a Nutshell

Quality Assurance (QA) happens late in the process (sometimes too late)!

Specs ➔ Development / Maintenance ➔ Testing ➔ Product delivery

Time

ISO/IEC 9126, Part 1

*Software product quality characteristics*

ISO/IEC 9126

Software Product Quality

- functionality
- reliability
- usability
- maintainability
- efficiency
- portability
- analysability
- changeability
- stability
- testability

**International Organization for Standardization**

Tiago L. Alves, Software Improvement Group / University of Minho
11th International Workshop on Simulation & EDGE Facilities for Space Programmes (SESP 2010)
Quality Assessment

Quality model
- Five stars for the 5% best systems; One star for the 5% worst systems
- Risk profiles are used as quality indicators for non-system level metrics
- Validation: better maintenance => faster issue resolution

Heitlager et al. [2]
### Metrics Table

Showing 1 - 15 of 1,600 items [Previous](#) | [Next](#)

<table>
<thead>
<tr>
<th>Name</th>
<th>McCabe complexity</th>
<th>Decision density</th>
<th>LOC/McCabe 1-10</th>
<th>LOC/McCabe 11-20</th>
<th>LOC/McCabe 21-50</th>
<th>LOC/McCabe 51+</th>
<th>LOC/McCabe 1-10 %</th>
<th>LOC/McCabe 11-20 %</th>
<th>LOC/McCabe 21-50 %</th>
<th>LOC/McCabe 51+ %</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScriptHostImpl.java</td>
<td>+</td>
<td>438</td>
<td>14%</td>
<td>1,893</td>
<td>842</td>
<td>0</td>
<td>0</td>
<td>69%</td>
<td>31%</td>
<td>0%</td>
</tr>
<tr>
<td>ANDDataWindow.java</td>
<td>+</td>
<td>375</td>
<td>15%</td>
<td>1,322</td>
<td>342</td>
<td>277</td>
<td>319</td>
<td>59%</td>
<td>15%</td>
<td>12%</td>
</tr>
<tr>
<td>DataTableEntry.java</td>
<td>+</td>
<td>271</td>
<td>22%</td>
<td>493</td>
<td>0</td>
<td>254</td>
<td>292</td>
<td>47%</td>
<td>0%</td>
<td>24%</td>
</tr>
<tr>
<td>CorbaDynamic.java</td>
<td>+</td>
<td>249</td>
<td>22%</td>
<td>63</td>
<td>174</td>
<td>119</td>
<td>683</td>
<td>6%</td>
<td>17%</td>
<td>11%</td>
</tr>
<tr>
<td>GRDDataWindow.java</td>
<td>+</td>
<td>244</td>
<td>14%</td>
<td>1,355</td>
<td>184</td>
<td>26</td>
<td>0</td>
<td>87%</td>
<td>12%</td>
<td>2%</td>
</tr>
<tr>
<td>OfflineLoggerCorrector.java</td>
<td>+</td>
<td>230</td>
<td>12%</td>
<td>236</td>
<td>125</td>
<td>0</td>
<td>1,435</td>
<td>13%</td>
<td>7%</td>
<td>0%</td>
</tr>
<tr>
<td>CommandView.java</td>
<td>+</td>
<td>229</td>
<td>12%</td>
<td>1,137</td>
<td>505</td>
<td>106</td>
<td>0</td>
<td>65%</td>
<td>29%</td>
<td>5%</td>
</tr>
<tr>
<td>LogViewersView.java</td>
<td>+</td>
<td>218</td>
<td>11%</td>
<td>1,172</td>
<td>372</td>
<td>145</td>
<td>0</td>
<td>69%</td>
<td>22%</td>
<td>9%</td>
</tr>
<tr>
<td>ExpressionFormulaObject.java</td>
<td>+</td>
<td>208</td>
<td>16%</td>
<td>113</td>
<td>716</td>
<td>236</td>
<td>0</td>
<td>11%</td>
<td>67%</td>
<td>22%</td>
</tr>
<tr>
<td>PatchedDurationImpl.java</td>
<td>+</td>
<td>201</td>
<td>26%</td>
<td>432</td>
<td>145</td>
<td>121</td>
<td>0</td>
<td>62%</td>
<td>21%</td>
<td>17%</td>
</tr>
<tr>
<td>TypesSwitch.java</td>
<td>+</td>
<td>200</td>
<td>41%</td>
<td>126</td>
<td>0</td>
<td>0</td>
<td>291</td>
<td>31%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>WorkspaceManager.java</td>
<td>+</td>
<td>196</td>
<td>18%</td>
<td>560</td>
<td>136</td>
<td>243</td>
<td>0</td>
<td>60%</td>
<td>14%</td>
<td>26%</td>
</tr>
<tr>
<td>TableControl.java</td>
<td>+</td>
<td>195</td>
<td>16%</td>
<td>448</td>
<td>189</td>
<td>0</td>
<td>485</td>
<td>40%</td>
<td>17%</td>
<td>0%</td>
</tr>
<tr>
<td>GenericTreeEditor.java</td>
<td>+</td>
<td>193</td>
<td>18%</td>
<td>853</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>PrimitiveTypesFactoryImpl.java</td>
<td>+</td>
<td>178</td>
<td>25%</td>
<td>194</td>
<td>0</td>
<td>452</td>
<td>0</td>
<td>30%</td>
<td>0%</td>
<td>70%</td>
</tr>
</tbody>
</table>

---

**Page generated on:** August 23, 2010 | **Software monitor, version 3.2.24 (standard configuration)** | **Copyright (c) 2006-2010 Software Improvement Group**

**Tiago L. Alves, Software Improvement Group / University of Minho**

**11th International Workshop on Simulation & EDGE Facilities for Space Programmes (SESP 2010)**
Quality ratings to support management

SimSat Maintainability

- Volume ★★★★★
- Duplication ★★★★★
- Unit size ★★★★★
- Unit complexity ★★★★★
- Unit interfacing ★★★★★
- Testing ★★★★★

EuroSim Maintainability

- Volume ★★★★★
- Duplication ★★★★★
- Unit size ★★★★★
- Unit complexity ★★★★★
- Unit interfacing ★★★★★
- Testing ★★★★★
Can source code analysis bring information to management?

### ESA SimSat Case Study

<table>
<thead>
<tr>
<th>Simulation Kernel</th>
<th>MMI (Man-Machine Interface)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Background</strong></td>
<td><strong>Background</strong></td>
</tr>
<tr>
<td>• Responsible for the simulation execution</td>
<td>• Responsible for the (graphical) user interface</td>
</tr>
<tr>
<td>• Developed in C/C++</td>
<td>• Developed in Java + Eclipse RCP</td>
</tr>
<tr>
<td>• In maintenance for several years</td>
<td>• Recently built (at the time of the analysis)</td>
</tr>
<tr>
<td><strong>Duplication:</strong> 10%</td>
<td><strong>Duplication:</strong> 11%</td>
</tr>
<tr>
<td><strong>Unit Complexity:</strong> 14%&lt;sup&gt;1&lt;/sup&gt;</td>
<td><strong>Unit Complexity:</strong> 21%&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

Low quality in newly developed code reveals lack of expertise

---

<sup>1</sup>moderate, high and very-high risk

---

Tiago L. Alves, Software Improvement Group / University of Minho
11th International Workshop on Simulation & EDGE Facilities for Space Programmes (SESP 2010)
Can source code analysis bring information to management?

Quality differences revealed different expertise between teams

Tiago L. Alves, Software Improvement Group / University of Minho
11th International Workshop on Simulation & EDGE Facilities for Space Programmes (SESP 2010)
When do you want to hear the bad news?

Measure early, manage early!
Conclusion

Summary

- How to measure software in a meaningful way
- How quality ratings can be derived from measurements
- The use of quality ratings to support management

Usage of quality models

- Act as early-warning for problems
- Give control over the delivered quality

Time to adopt quality models as part of the process!!!
Thank you!

t.alves@sig.eu