“Cleanroom Software Development - An Empirical Evaluation”

RICHARD W. SELBY, MEMBER, IEEE, VICTOR R. BASILI, SENIOR MEMBER, IEEE, AND F. TERRY BAKER

Presented by
Moshfiqur Rahman, Robert Hellebrand
Cleanroom

• Properties
  – Conforms the requirements
  – High operational reliability
  – Easily readable sourcecode

• History
  – Introduced by IBM Federal Systems Division (FSD)
  – Deny entry of defects
  – Develop software with certifiable reliability
Components of Cleanroom

• Software Life Cycle of Executable Increments
• Formal Methods
  – Specification
  – Design
• No program execution
• Statistically based independent testing
Roadmap of study

• Introduce Cleanroom to developers
• Use GQM approach to get measures
• Execute study
  – Two groups (cleanroom, control group)
• Analyze data
Context of study

- Electronic message system
- About 1500 LOC
- Project duration: 6 weeks
- Requirements are given
- Team experience of 1.6 years in average
- Hawthorne effect avoided
- Access to independent testing
- Motivated by grading
### Collected data

<table>
<thead>
<tr>
<th>Team</th>
<th>Cleanroom</th>
<th>Source Lines</th>
<th>Executable Statements</th>
<th>Procedures &amp; Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>yes</td>
<td>1681</td>
<td>813</td>
<td>55</td>
</tr>
<tr>
<td>B</td>
<td>yes</td>
<td>1626</td>
<td>717</td>
<td>42</td>
</tr>
<tr>
<td>C</td>
<td>yes</td>
<td>1118</td>
<td>573</td>
<td>42</td>
</tr>
<tr>
<td>D</td>
<td>yes</td>
<td>1046</td>
<td>477</td>
<td>30</td>
</tr>
<tr>
<td>E</td>
<td>yes</td>
<td>1087</td>
<td>624</td>
<td>32</td>
</tr>
<tr>
<td>F</td>
<td>yes</td>
<td>1213</td>
<td>440</td>
<td>35</td>
</tr>
<tr>
<td>G</td>
<td>yes</td>
<td>1196</td>
<td>581</td>
<td>31</td>
</tr>
<tr>
<td>H</td>
<td>yes</td>
<td>1876</td>
<td>550</td>
<td>51</td>
</tr>
<tr>
<td>I</td>
<td>yes</td>
<td>1305</td>
<td>608</td>
<td>23</td>
</tr>
<tr>
<td>J</td>
<td>yes</td>
<td>1052</td>
<td>658</td>
<td>24</td>
</tr>
<tr>
<td>a</td>
<td>no</td>
<td>824</td>
<td>410</td>
<td>26</td>
</tr>
<tr>
<td>b</td>
<td>no</td>
<td>1429</td>
<td>633</td>
<td>18</td>
</tr>
<tr>
<td>c</td>
<td>no</td>
<td>2264</td>
<td>999</td>
<td>46</td>
</tr>
<tr>
<td>d</td>
<td>no</td>
<td>1629</td>
<td>626</td>
<td>67</td>
</tr>
<tr>
<td>e</td>
<td>no</td>
<td>1310</td>
<td>459</td>
<td>43</td>
</tr>
</tbody>
</table>

*Fig. 3. System statistics.*
Analysis and Interpretation

• Requirement conformance

Fig. 4. Requirement conformance of the systems.
Analysis and Interpretation

• Successful test cases

Fig. 5. Percentage of successful test cases during operational testing (without duplicate failures).
### Analysis and Interpretation

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Yes, they were effective for testing all parts of the program</td>
</tr>
<tr>
<td>5.5</td>
<td>We used them but felt that they were only appropriate for certain parts of the program</td>
</tr>
<tr>
<td>8.5</td>
<td>We used them occasionally, but they were not really a major contributing factor to the development</td>
</tr>
<tr>
<td>0</td>
<td>Did not really use them at all</td>
</tr>
</tbody>
</table>

**Fig. 6. Breakdown of responses to the attitude survey question, “Did you feel that you and your team members effectively used off-line review techniques in testing your project?” (Responses are from Cleanroom teams.)**

\[
\text{Mann-Whitney signif. } = .065
\]
Analysis and Interpretation

- Extense of Computer use

Fig. 7. Connect time in hours during project development.\textsuperscript{9}
Analysis and Interpretation

• Number of system releases

Mann-Whitney signif. = .006

Fig. 8. Number of system releases.
Analysis and Interpretation

Spearman correlations: $-0.85$ (signif. = 0.02) with source lines; $-0.70$ (signif. = 0.03) with number separately compilable modules; $-0.57$ (signif. = 0.09) with number procedures and functions.

Fig. 10. Relationship of program size versus missing program execution.
Conclusion

- Requirements are met more often (91%)
- Better predictability
- Schedules are met
- More comments, less control flow complexity
- 81% of developers would use it again
Thanks for your attention

• Questions?