Patchwork - A Creative Approach To Revise Existing Validation Methods

Anja Feuerbacher
PhUSE 2014
Contents

• Introduction
• Why should you revise your validation packages?
• Give me an example: How did you do it?
• What was the outcome?
• Were there consequences for the validation process in general?
• Conclusion
Introduction

- Patchwork: old technique reutilizing textile remnants which are sewn together into a larger design, using different patterns and fabrics
- Validation and verification: program fulfills requirements and functions properly
- Regression testing: ensure that modifications do not introduce errors into a working program
- Validation package: instructions document, supporting files (can be: code, metadata, datasets, outputs, log files)
Why should you revise your validation packages?

• Similar tests
• Duplicates
• Unclear or inconsistent instructions
• People make mistakes
• Manual testing
• Software versions
• Unneeded functionality
Give me an example: How did you do it?

- autoexec.sas: applied to all studies defining standard libnames, filenames, global variables, search paths, …
- Step 1: Thorough review of validation instructions document
- Step 2: Set up new document outlining all categories, then cut and paste each test to respective category
- Step 3: Combine similar tests into unique tests, keep tested values and options as metadata
- Step 4: Remove tests for unneeded functionality, align remaining tests
- Step 5: Implement appropriate automation
- Step 6: Review and optimization
What was the outcome?

<table>
<thead>
<tr>
<th></th>
<th>Old validation package</th>
<th>Revised validation package</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td># of pages in instructions document</td>
<td>42</td>
<td>13</td>
<td>-69%</td>
</tr>
<tr>
<td># of tests</td>
<td>36</td>
<td>10</td>
<td>-72%</td>
</tr>
<tr>
<td>- manual</td>
<td>23</td>
<td>1</td>
<td>-96%</td>
</tr>
<tr>
<td>- automatic</td>
<td>13</td>
<td>9</td>
<td>-31%</td>
</tr>
<tr>
<td># of files supporting the process</td>
<td>126</td>
<td>33</td>
<td>-74%</td>
</tr>
<tr>
<td>Execution time</td>
<td>several hours</td>
<td>less than 1 hour</td>
<td>nameless relief!</td>
</tr>
</tbody>
</table>

Table 1: Comparison of the old and revised validation package
 Were there consequences for the validation process in general?

Mid-term:
- Review/revision of other validation packages
- Agreement within programming group that automation is crucial

Long-term:
- Discussions about the process
- Goal: Create validation concept independent of particular tools
Conclusion

• Programmers are reluctant to implement change requests?
  -> Think about your validation concept!

• Good exercise: Just revise one of your validation packages

• Do automated testing from the beginning, supporting unique tests, and using a reasonable metadata approach
  -> Less error-prone, better supports program development
Questions?