
Patchwork - A Creative Approach To Revise Existing Validation Methods



Anja Feuerbacher
PhUSE 2014

- 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

- 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?

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

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?

