Validation: A DataFax Administrator's Perspective

Darryl Pahl
DF/net Research Inc.

Rethinking Validation

- How did it get so out of control?
- Too much effort on meeting needs of others
- Re-balance the scales
- Meet regulatory requirements and still add value

What validation means to me

1) Information in equals data out
2) DataFax installed correctly on a solid foundation
3) Sufficient documentation to meet regulatory concerns

What Hasn't Worked

Suggestions As to What Might

Foundation Qualification

Out with the old, in with the new

- Enough already with IQ, PQ, OQ
- Generic, dated terms from manufacturing
- Not as appropriate for software systems
- Concept valid, but classic definitions limiting for DataFax

Invent your own terms

- Adapt IQ, PQ, OQ to your own needs
- Foundation Qualification
- Data Flow Qualification
- Ongoing Quality Assurance
- Use a glossary to map terms

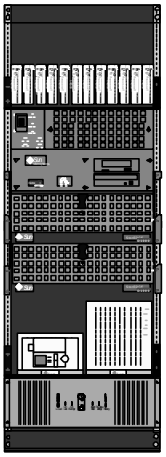
Meaningless IQ

- Installation? of what? DataFax? Hardware? Solaris?
- Yes to all of the above
- Scope of work often limits IQ
- Result is valid DataFax on shaky ground

Foundation Qualification

- Success based on the foundation
- Foundation includes everything imaginable
- Simple, clear, concise documentation of what you know
- Explore what you don't know

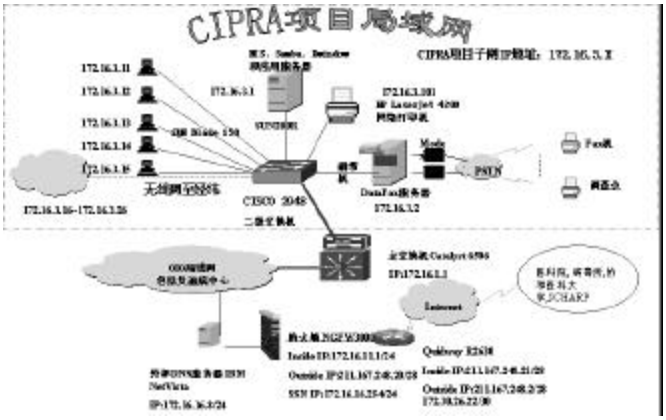
Simple, Clear, Concise Documentation



- Spend time on content, not format
- Diagrams, simple configuration notes
- Logical and physical layout of system

Example

- Diagrams are clearer, quicker, easier



Meaningless Requirements

- Requirements critical to validation
- Requirements should drive the testing plan – not the reverse
- If you can't test it - you can't require it

Example

Bad Requirement:	Good Requirement:
DataFax must produce good quality data	*The data exported from DataFax must be appropriately equivalent to the data written on the CRFs*

Good Requirements

- Limit requirements to those that answer:

- *"Still use DataFax without this?"*
- *"Requirement specific enough to test for it?"*
- *"Requirement, not desirement?"*
- *"Limited to data flow?"*

Data Flow Qualification

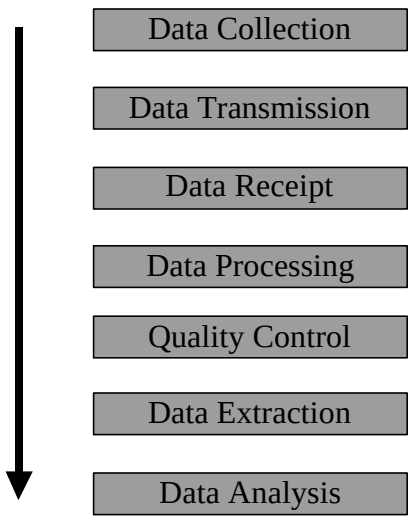
Barking up the wrong tree

- Validation often has wrong focus
- Bogs down because direction and purpose isn't clear
- Too much time testing what you know works
- Not enough on what you hope works

Data Flow Qualification

- DataFax is all about data flow
- Define each step of the process
- “Holistic” DataFax testing centered on data flow

DataFax Data Flow



Detailed Testing Plans

- You need to have testing plans
- Too much detail takes too much effort to create / approve
- Too much detail leads to robotic execution
- Which leads to missed steps

Flexible, high-level planning

- Testing should be an experiment
- “Lab notes” as documentation
- Build in ability to go in different directions
- Prove what does work, stress what may not work

Example:

Requirement:

- *DataFax is able to receive email faxes from specified Internet-ready fax machines at sites via the Internet.*

Working Hypothesis:

1) *A testing fax machine with similar characteristics as at the sites placed outside of the corporate firewall can be used to send email to the address datafax@company.com. These faxes should appear in DataFax.*

Example:

Limitations:

- *Not all models of fax machines can be tested. Test fax machine represents common model at sites.*
- *Sites may have Internet connections of varying quality which can not be duplicated in this test.*

Stress Testing:

2) *What happens if the Internet connection is down at the site?*
3) *What happens if the corporate mail server is down? How do we know it is down? How does the site know? Does email bounce?*

Example:

Testing Notes:

1) *Hypothesis Test Result*

   _____
   _____
   _____

2) *Stress Test (Internet connection down)*

   _____
   _____
   _____

Screenshot Hell

- Proof of execution leads to mountains of work and paper
- Paper doesn't prove anything
- Burden that only deters honest people

Forget Screenshots

- Screenshot only if it adds value
- If proof is required, use a witness
- Two people working on validation side by side adds value
- Hiring a temp to perform screenshot does not

Ongoing Quality Assurance

Great – it's over

- Time for your validation binders to gather dust on the shelf !
- Bad memories fade fast
- Next time to validate – DataFax 4.0 !

Validation Should Be Ongoing

- DataFax is validated, how about study setups? QC reports?
- Dynamic testing contained in ongoing Data QA
- Test what can't effectively be tested as part of foundation or data flow

Ongoing Data QA

- Helpful tools:

- deSCRIBE tool (Phil Kirsch)
- CDSI Study Setup Worksheets
- SOP for study testing
- DFprintdb

Change Control

- Don’t make this a solution looking for a problem
- Keep it simple – use email alias to send changes to
- Sort by date, submitter, search, archive, access remotely

Periodic Review

- Periodic review keeps problems from compounding
- Don’t model periodic review like a trip to the dentist
- Make review part of routine

Take home message

- How did it get so out of control? Lack of balance
- Meet your own needs first. Be egocentric
- Re-balance the scales towards usefulness

Questions?

Darryl Pahl
DF/Net Research, Inc.
darryl@dfnetconsulting.com