The 9 Most Common Validation Errors:

IDENTIFY FREQUENT DEFICIENCIES TO ACCELERATE YOUR VALIDATION PROJECTS

BY FRANK HOUSTON
Senior Validation Consultant, EduQuest, Inc.

TABLE OF CONTENTS

Lessons Learned from 1,720 Observations

What validation problems are you likely to see over and over? When tackling complex validation challenges, you'll save time, money and headaches when you know the most common problems and where to find them.

The following analysis is based on validation work EduQuest performed for a large FDA-regulated company over the past year. The goal was to bring the company's software validation evidence up to the level of the U.S. FDA's current expectations as well as those of the client's own independent auditor.

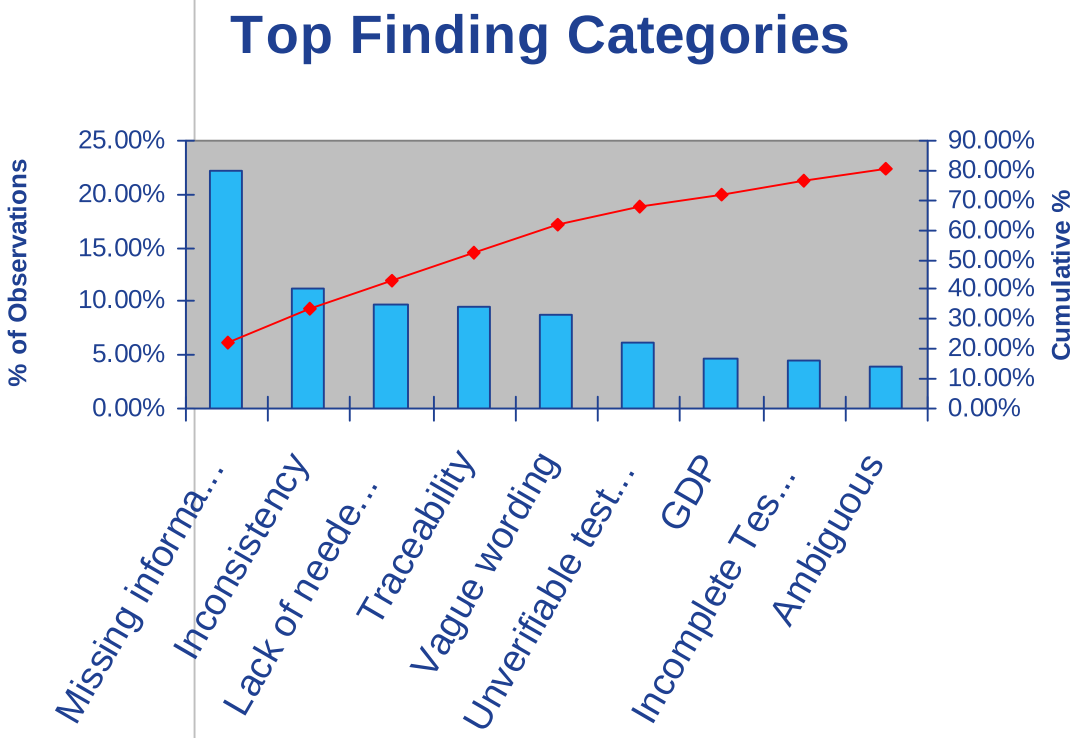
Our efforts yielded 1,720 observations. As part of a "lessons learned" review, we grouped the observations into 22 different categories and, in the second part of our analysis, identified which documents most frequently contained the observations. The results – in our experience – are typical of the problems most companies face.

1896 Urbana Pike, Suite 14
Hyattstown, MD 20871 USA
(Near Washington, DC)
Phone: +1 (301) 874-6031
Email: Info@EduQuest.net
www.EduQuest.net

Applying Pareto Analysis to Common Validation Problems Finds 9 Types of Deficiencies

Through Pareto analysis of the categories of problems, we discovered that about 80% of our observations were clustered around nine types of deficiencies plotted on **Figure 1**. Our case was an exception to the 80/20 rule in that the top nine problem areas represented about 41% of our categories.

FIGURE 1



The most frequent deficiencies we found were:

1. **Missing Information** – Documents or records omitted fundamental information or content that should have been included.

2. **Inconsistency** – Documents contained statements inconsistent with other statements about the same topic

in the same document or in the same validation package. What's more, no explanation or reason was given for the difference. We found that jargon, varying terminology, and contradictions in logic frequently caused these kinds of inconsistencies.

Jargon and varying terminology frequently cause validation inconsistencies.

3. **Lack of Needed Detail** – This deficiency applied mostly to requirements documents. The requirements in the validation package did not adequately describe the characteristics of 1) data, 2) user interactions with business processes, or 3) key processes internal to the software.

4. **Traceability** – We found three frequent traceability problems:

   - The traceability matrix did not account for a traceable specification or an observation step in a test script.

   - The trace was broken. Either a requirement was barren (lacked decedents or a test) or one of the detailed requirements or test results was an orphan (lacked a parent somewhere in the requirement tree).

   - The traceability matrix was incomplete. Requirement details were not explicitly numbered and traced to associated test steps. Requirements were not traced at a detailed level, so the reviewer needed to infer the detailed links between specifications and steps in a test script.

5. **Vague Wording** – Documents used generalities such as "in accordance to an approved procedure", or "applicable regulatory requirements", or "all associated

GxP and business processes”. In addition, documents used vague words such as “may”, “possibly”, “more or less”, and “approximately”.

6. **Unverifiable Test Results** – Expected results were not described sufficiently so that an independent reviewer could compare and verify actual results. The IEEE Standard for Software Test Documentation, Std. 829.1988, Clause 6.2.4 says you should, “...provide the exact value (with tolerances where appropriate) for each required output or feature”. For executed scripts, actual results were not recorded or captured in a way that allowed an independent reviewer to compare them to expected results. For example, “OK” was noted in the actual-result column with no reference to a screen shot.

For executed scripts, actual results often were not recorded to allow comparison with expected results.

7. **GDP** – We found three frequent Good Documentation Practice problems:

   - Hand-recorded data and testing evidence, such as test results, were presented in a way that could cause doubts about their authenticity (for example, cross-outs without initials, date, and reason).

   - Data that confirmed a specific requirement was hard to find in the evidence provided (for example, a busy screen shot crammed with data).

   - Handwritten corrections were made that changed the sense of a requirement or an expected test result, but no discrepancy report or change request was filed (for example, changing an expected result from indicator “Off” to “On”). In GDP, hand corrections are allowed without additional documentation only for obvious typographical errors, such as dropped or transposed letters (for example, correcting "th" or "teh" to "the").

8. **Incomplete Testing** – Test scripts did not fully or adequately test the associated requirement.

9. **Ambiguity** – Text could be interpreted more than one way, so it did not establish a single, unique requirement. The words "either" and "or" in a requirement are strong clues the text is ambiguous.

The words "either" or "or" in a requirement are strong clues the text is ambiguous.

13 Additional Observation Categories

Beyond these top nine categories of deficiencies, we identified 13 other categories from our observations. One might consider our category definitions were somewhat subjective, but for this sort of analysis the objectivity of the definitions was less important than consistency in classifying the observations. For this reason, we reviewed all the classifications several times before locking in the data for our lessons-learned pivot tables. Even so, we noted that between the "Ambiguous" and "Vague Wording" classifications, many observations could have fit in either one.

To help you perform your own analysis, here are the additional categories of deficiencies we identified – ones that did not rise to the level of our most common findings but were still worth noting:

- **Compound Requirement —** Requirements that were not unique; that is, the requirement statement actually stipulated two or more system characteristics. (When the predicate of a requirement sentence contains "and" or a series of commas, or when the requirement is presented as a compound sentence or series of bullets, it's probably a compound requirement. This deficiency was often coupled with traceability problems.)

- **For Your Information** – Here we included comments on the potential to improve a document or process. The issue that generated the comment may or may not have had an impact on a determination of "substantial compliance." We also included remarks on particularly good examples of documentation or development practice.

- **Incomplete Requirements** – Our findings in this category fell into four subcategories:

  1. The requirement in question implied another requirement, possibly complementary, that needed to be explicit to ensure verification.

  2. Regulatory impact analysis and risk assessment indicated a need for requirements that were missing from the User Requirement Specification (URS).

  3. Requirements in a Software Requirements Specification (SRS), a System Design Specification (SDS) or a Configuration Specification (CS) were not sufficient to address the associated URS item. This deficiency was often associated with a broken trace.

  4. System and business process analyses indicated the software had functionality that was used but had not been described in the URS.

- **Rationale** – Statements or assertions were made without supporting rationale or justification. Or, the

Risk assessment indicated a need for requirements that were missing from the URS.

rationale or justification for a particular statement or assertion was not persuasive.

- **Lack of Acceptance Criteria** – Test and validation plans did not establish objective criteria based on the outcomes of various tasks in the validation process, such as vendor audit, testing, and problem resolution. The plans did not include criteria for assessing the seriousness of deviations as a basis for the overall evaluation and acceptance or rejection of the test and validation results.

Test plans did not include objective criteria for assessing the seriousness of deviations.

- **Lack of Process for Resolving Deviations** – A plan, protocol or script lacked a process for resolving deviations (for example, failure to meet expected test results, discovery of unanticipated behavior, or deviations from Good Documentation Practices).
- **Questionable Statement** – A statement appeared to be inaccurate or incorrect.
- **Redundant Requirement** –The same requirement appeared more than once in a specification document.
- **Topical Inconsistency –** The text within a topic pertained to a different topic.
- **Typo** – Typographical errors were observed.
- **Unsupported Deviation** – The summary document omitted reporting on differences between planned activities and those that were actually carried out.
- **Untestable Requirement** – The requirement was not presented in objective, observable or measurable terms. In other words, the requirement did not describe a

system response or characteristic that a reasonable person could sense or measure.

- **Violation** – The text set up or highlighted a violation of procedures or regulations.

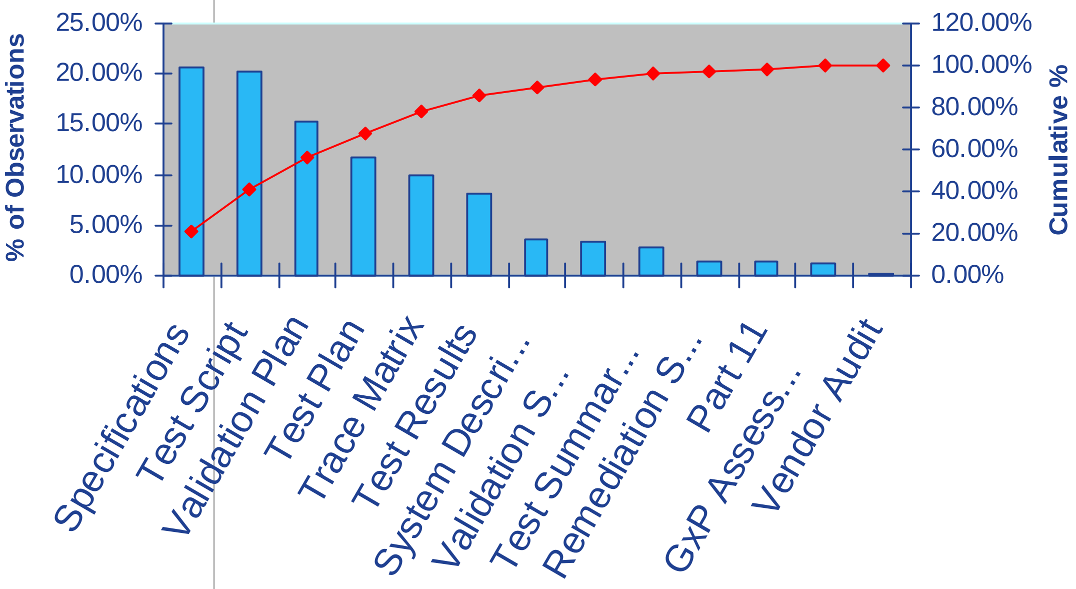
Consider all these categories nothing more than suggestions or starting points for your own list of observations. As you gain experience, you'll inevitably revise or cull out some categories and identify new ones.

Identifying the Most Vulnerable Documents

Taking the next step to document the lessons learned from this project, we then categorized the documents and records where we found the most frequent deficiencies. We discovered that about 85% of findings were concentrated in six key documentation areas, as shown in **Figure 2**.

FIGURE 2

Top Document Types



The top types of flawed documentation were:

1. Specifications (including User Requirements)
2. Test Scripts
3. Validation Plans
4. Test Plans
5. Trace Matrix
6. Test Results

Although the exact order of problem areas may differ in your organization, it's likely these same six documentation areas will float to the top. From our experience, specification documents are usually the biggest pitfall for most companies.

Companies who reduce the frequency of these deviations are more likely to weather FDA inspections.

Fewer Validation Problems and Inspection Success Go Hand-in-Hand

After auditing many companies, large and small, and participating in countless remediation projects, EduQuest has found the results described above typical of companies worldwide.

More importantly, we also have seen first-hand that companies who reduce the frequency of these problems with focused remediation efforts are much more likely to weather future FDA inspections. You can reasonably assume the same would be true if the frequency of such problems were low in the first place.

We recommend you use these results and definitions to assess your own validation projects, or devise your own categories and charts to pinpoint your company's most common problems. Either way, you'll have a major head-start in better allocating validation resources and making needed improvements quickly.

About the Author

Melvin F. (Frank) Houston is a Senior Validation Consultant with EduQuest, Inc., a global team of FDA compliance experts based near Washington, D.C. Frank is a recognized authority on ISO 9000 Quality Standards and Quality System Regulation. He has particular expertise in the validation and auditing of computer-controlled systems in regulated industry. He also is considered an expert in software quality assurance and a champion of software/ system safety.

Before joining EduQuest, Frank served 12 years with the U.S. FDA Center for Devices and Radiological Health (CDRH), where he influenced the formation and execution of FDA software policies for medical devices and Good Manufacturing Practices. He also develops instructional materials for EduQuest's popular "**FDA Auditing of Computerized Systems and Part 11**" training course, which is offered as an open enrollment course several times each year throughout the U.S. and Europe. The highly interactive course — which further explains FDA's rules and expectations for systems validation — also is available for on-site, on-demand delivery. For course details, email MartinHeavner@EduQuest.net.

EduQuest

1896 Urbana Pike, Suite 14
Hyattstown, MD 20871 USA
(Near Washington, DC)
Phone: +1 (301) 874-6031
Fax: +1 (301) 874-6033
Email: Info@EduQuest.net
www.EduQuest.net