13

March 2011

testing experience

The Magazine for Professional Testers

printed in Germany

print version 8,00 €

free digital version

www.testingexperience.com

ISSN 1866-5705

Testing @ Domains –

How does Finance, Automotive, Medical etc test? Do we have to take care of the domains?

Conferences Special

iqnite DEUTSCHLAND 2011

Testing in the pharmaceutical contract research organization domain

by Julie Lacroix

Domain constraints

Performing software testing in a Contract Research Organization (CRO)[1] changes your testing vision for the rest of your life. The pharmaceutical industry is heavily regulated, and software testing used in support of clinical research is not an exception. First, the vocabulary used in the pharmaceutical world is quite different; test plans and test reports are known as validation protocols and validation reports; user acceptance testing is known as validation testing, etc... Secondly, one of the worst nightmares of IT professionals working in CROs is THE DOCUMENTATION. Everything needs to be documented to demonstrate what actions were really performed. One rule in the pharmaceutical world is: "If it's not documented, it doesn't exist". So, you can imagine how much paper, printouts, and test cases need to be produced to prove the existence of testing activities on software used in support of clinical studies. In addition, every activity performed as part of testing needs to be clearly explained and documented in Standard Operating Procedures (SOPs). These procedures are regularly inspected and challenged by clients and agencies during audits and inspections.

Regulatory agencies' rules and guidelines also add constraints to software testing. The most known regulatory agency is the FDA[2] (USA). CRO clients, however, come from around the world; therefore testing procedures must be compliant with many regulatory agencies. One of the most famous regulations in the pharmaceutical domain is the CFR21 Part 11[3]. This regulation covers how software and companies must deal with electronic records and electronic signatures for clinical study data. Security of data, training of staff responsible for developing, maintaining or using the software, audit trail properties, and components of electronic signatures are explained as part of the regulation. Additionally, clinical study data (in paper or electronic form) needs to be readily available during the retention period required, i.e., 25 years. I'll let you imagine the backward compatibility challenges, and associated testing issues, where electronic records are concerned.

Domain expectations

In order to help the industry to comply with these standards, the FDA published guidance on software validation in 2002[4]. This guidance explains the FDA's point of view and their expectations regarding software validation. In this guidance, principles of software validation are listed and explained. Table 1 presents the 10 principles and a short description of each. The European Commission Health and Consumers Directorate-General[5] also have created a guideline regarding computerized systems (Volume 4 Annex 11[6]). This guideline was issued in 1997 and revised in January 2011. The guideline lists 17 principles related to computerized systems. From these principles, the fourth one concerns software validation and is subdivided into 8 key elements. Table 2 presents these key elements from the European Commission and a short description of each. When both guidelines are compared, the principles and expectations are mostly equivalents. However, the new edition of the European Commission guideline adds more details. This may result in new approaches for future audits performed by European regulatory agencies.

Algorithme Pharma Approach

Algorithme Pharma[7], a CRO located in Montreal, Canada, appointed a software validation team (known as the testing team to the IT world) to deal with these industry expectations for software testing. Software used by Algorithme staff ranges from in-house developed to well-established vendor-supplied, from desktop to web-based applications, and from clinical to biostatistics applications. This rich environment requires adoption of a strong and efficient testing strategy to ensure that all these applications are compliant with regulations and behave as expected. The approach adopted by Algorithme Pharma is requirements-based

1 *http://en.wikipedia.org/wiki/Contract_research_organization*

2 *http://www.fda.gov/*

3 *http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=11&showFR=1*

4 *http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM085371.pdf*

5 *http://ec.europa.eu/health/documents/eudralex/index_en.htm*

6 *http://ec.europa.eu/health/files/eudralex/vol-4/annex11_01-2011_en.pdf*

7 *http://www.algopharm.com/*

Table 1. Principles of Software Validation FDA

Principles of Software Validation	Description
Requirements	Documented software requirements specification provides a baseline for both validation and verification. The software validation process cannot be completed without an established software requirements specification.
Defect Prevention	Software quality assurance needs to focus on preventing the introduction of defects into the software development process and not on trying to "test quality into" the software code after it is written.
Time and Effort	To build a case that the software is validated requires time and effort. Preparation for software validation should begin early. The final conclusion that the software is validated should be based on evidence collected from planned efforts conducted throughout the software lifecycle.
Software Life Cycle	Software life cycle contains specific verification and validation tasks that are appropriate for the intended use of the software. The guidance does not recommend any particular life cycle models – only that they should be selected and used for a software development project.
Plans	The software validation process is defined and controlled through the use of a plan. The validation plan defines "what" is to be accomplished through the software validation.
Procedures	Procedures establish "how" to conduct the software validation effort. The procedures should identify the specific actions or sequence of actions that must be taken to complete individual validation activities, tasks and work items.
Software Validation After a Change	When any change is made to the software, the validation status of the software needs to be re-established. Whenever software is changed, a validation analysis should be conducted not just for validation of the individual change, but also to determine the extent and impact of that change on the entire software system.
Validation Coverage	Based on the software's complexity and safety risk – not on firm size or resource constraints. The selection of validation activities, tasks and work items should be commensurate with the complexity of the software design and risk associated with the use of the software for the specified intended use.
Independence of Review	Validation activities should be conducted using the basic quality assurance precept of 2independence of review". When possible, an independent evaluation is always better, especially for higher risk applications.
Flexibility and Responsibility	Specific implementation of these validation principles may be quite different from one application to another. Software validation process should be commensurate with the safety risk associated with the system, device, or process.

Bibliography

FDA. (2002). General Principles of Software Validation; Final Guidance for Industry and FDA staff.

Table 2. European Principles of Software Validation

Principles of Software Validation	Description
4.1 Life Cycle	Validation documentation and reports should cover the relevant steps of the life cycle.
4.2 Change Control	Validation documentation should include change control records and reports on any deviations observed during the validation process.
4.3 Inventory	An up to date listing of all relevant systems and their functionality should be available.
4.4 User Requirements Specifications (URS)	URS should describe the required functions of the computerized system and be based on documented risk assessment and impact.
4.5 Quality Management	Ensure that the system has been developed in accordance with an appropriate quality management system.
4.6 Customized Computerized Systems	Process is in place that ensures the formal assessment and reporting of quality and performance measures for all life-cycle stages of the system
4.7 Evidence	Evidence of appropriate test methods and test scenarios should be demonstrated.
4.8 Migration	If data are transferred to another data format or system, validation should include checks that data are not altered in value and/or meaning during the migration.

Bibliography

European Commission. (2010). Volume 4: Good Manufacturing Practice Medicinal Products for Human and Veterinary Use, Annex 11: Computerized Systems.

testing[8] completed with a risk-based testing technique[9]. Software used as part of clinical study is assessed against the compliance level of the application for 21 CFR part 11 and the risk related to this application.

The risk is evaluated on five factors:

- criticality on human life
- regulatory experience
- vulnerability if down
- industry distribution
- type of data (electronic, paper, mix of both) processing into the software

Once these factors are identified and evaluated, a risk level is determined. The risk analysis determines the extent of testing needed to satisfy both regulatory and internal quality standards. Criteria used to determine the testing level cover all Software Development Life Cycle (SDLC) phases, activities and good practices: requirement definition, formal design activities, peer reviews, known bugs, regression testing, acceptance testing etc... The goal of this approach is to maximize testing activities to reduce risk to an acceptable level (risk zero is the ultimate goal). The risk-based approach helps to identify the applications or parts of given software the testing team needs to devote more attention to and test more deeply. When an acceptable balance between risk and testing is reached, for a particular piece of software, this equilibrium needs to be preserved during subsequent software updates. For example strong change control procedures must be followed and regression testing must be performed.

The testing team at Algorithme Pharma has a double role. The primary goal of the team is to perform validation testing on applications used as part of clinical studies. The goal of the validation is to confirm the "fitness for use" of the application for its intended function. The second role is to perform verification testing for software developed in-house. Both roles require different types of expertise and testing techniques. Therefore different professional backgrounds are necessary in the testing team. To ensure testing expertise, Algorithme Pharma adopted and supports the ISTQB®[10] certification. This certification scheme was chosen over the other ones available due to its worldwide recognition and for being accepted as an industry standard. The background to assess the "fit for purpose" aspect is quite different to the pure software testing background. The ideal background for the "fitness for use" is a subject matter expert working directly in operations. However, such power users cannot work in the operations and test software on a regular basis. The compromise adopted by Algorithme Pharma was to build a testing team to design and execute verification tests (system testing) and to design validation tests (acceptance testing), while power users are responsible for the execution of the validation tests. However, testing staff needs to acquire a deep knowledge of the operations in order to design realistic and appropriate test cases. This knowledge is gained by assisting in departmental training and by observing operational tasks performed by the power users on their day to day operations. Another challenge for the "fitness for use" goal resides in the variety of functions existing in the company: clinical operations, bioanalytical laboratory, biostatistics and quality assurance. Different software solutions exist for these departments, therefore the testing team needs to understand each department's operation rules in order to design good test cases. To reduce the amount of scientific knowledge required by each tester, the testing team staff is assigned to projects based their domain knowledge to keep the testers in their area of expertise as much as possible.

The testing team is also involved in the requirements analysis phase for each project. This allows the testing team to understand the requests for change directly from the users, and to review final requirements based on their understanding. Therefore requirement static testing can be effectively performed. It also reduces the time needed for test case design since the testing team already knows the scope and content of the latest software iteration. The iterative SDLC model and the early involvement of the testing team allow the design, code and test case design to be performed in parallel (V-model)[11]. Testing techniques performed on all software versions include some exploratory testing and extensive scripted verification tests at system level and very detailed scripted test cases on user acceptance level in order to achieve 100% requirements-based testing and to address risks identified during initial analysis.

The software development model adopted by Algorithme Pharma is the "iterative time box"[12]. This model was preferred for its short return on investment (ROI) time, change management and to satisfy evolving business rules. For the users, this development model permits new functionalities to be put into production rapidly and to improve their efficiency. Two user acceptance testing techniques are performed by the power users: a formal execution of designed tests cases (requirement-based testing), and a more informal user exploratory session (documented in what's called "User Acceptance Report"). The later technique is performed using a copy of live data. Therefore, a user can choose a real life complex project and confirm that the new version fixes and/or improves the situation. This additional testing activity raises the confidence level of the users regarding the new software iteration by acting like early training and facilitates the implementation of the iteration into operation.

So, what's the difference?

Algorithme Pharma established an efficient and compliant approach for software testing. By combining the requirements-based approach, considered as one of the most rigorous testing techniques, with the risk-based approach, software used by Algorithme Pharma for clinical study meets regulatory agencies' expectations and internal standards. The Algorithme Pharma testing approach has successfully passed audits by both our customers and regulatory agencies.

What is the difference between software testing in a pure IT domain and in the pharmaceutical domain? There is not much difference regarding testing techniques, tools, concepts, etc... One of the differences resides mainly in the extension of residual anomalies in software, i.e., defects that escape testing. Let's say an anomaly is introduced in a calculation in biostatistics software. A drug formulation meets the acceptance criteria of regulatory

8 An approach to testing in which test cases are designed based on test objectives and test conditions derived from requirements. ISTQB Glossary of Testing Terms, version 2.1, April 1st, 2010.

9 http://en.wikipedia.org/wiki/Risk-based_testing

10 http://istqb.org/display/ISTQB/Home

11 A framework to describe the software development lifecycle activities from requirements specification to maintenance. The V-model illustrates how testing activities can be integrated into each phase of the software development lifecycle. ISTQB Glossary of Testing Terms, version 2.1, April 1st, 2010.

12 Rapid Development, Steve McConnell, Microsoft Press, 1996

agencies following a clinical study. The drug formulation could be approved and will be distributed in pharmacies and hospitals. In the worst case scenario, the drug formulation may cause the death of patients. Because human life can be impacted somewhere, software testing in the CRO domain must be taken very seriously. The domain requires a very formal process to document the testing activities. The documentation aspect is really the main difference. The documentation must be able to reproduce, step-by-step, the testing phase performed before the deployment of software in the operations environment. Testing is not enough in the domain, testing must be accompanied by the appropriate documentation. In conclusion, since a picture is worth a thousand words, this cartoon represents the documentation reality in the pharmaceutical domain...

> biography

Julie Lacroix
has over 12 years of experience at Algorithme Pharma including 6 years as a Software Validation Manager. She is also a board member of the Canadian Software Testing Board (CSTB) as the Vice-President Examinations. Julie's competencies are team management, system and user acceptance testing along with regulations and compliance. Her previous experience in the Bioanalytical Laboratory of Algorithme Pharma helps her to better understand user expectations for software. Julie is ISTQB certified at the Foundation Level and at the Advanced Level as Test Manager and as Test Analyst. http://ca.linkedin.com/in/julielacroix3258