A Management Approach to Software Validation Requirements

Nadica Hrgarek
Fürstenweg 144 C Top 40, 6020 Innsbruck, Austria
nadica.hrgarek@gmail.com

Abstract. *Structured software validation approach varies from organization to organization. All medical device manufacturers shall determine what needs to be validated and how much validation is enough to ensure regulatory requirements are met.*

The goal of this paper is to describe a management approach to improve the software validation process for medical device manufacturers in order to reduce the validation time and cost.

Validation of computerised systems can generate a lot of documentation, be very costly (e.g. validation of an ERP system) for most organizations, but also can provide business and regulatory benefits. The paper discusses the challenges of computer systems validation and highlights some validation methods to help medical device companies to comply with FDA and ISO requirements for the validation of non-product software which includes off-the-shelf (OTS) software.

Keywords. FDA, medical device, OTS software, software validation, validation

1 Introduction

All medical device manufacturers shall determine what needs to be validated and how much software validation is enough to ensure regulatory requirements are met. In an FDA (Food and Drug Administration) regulated company, validation is typically mandatory when the software affects product identity, safety, strength, efficacy or quality.

In the medical device industry, the non-product software used as part of production or the quality system must be validated for its intended use. Intended use means, we don't validate Excel application, we validate how we are using it [21]. For example, if an Excel spreadsheet having multiple functions is used to log corrective and preventive actions (CAPAs) too, only the aspect of CAPAs logging should be defined as the intended use. If the software is used to support a quality decision, produces data for management review or if it affects the product, it has to be validated.

No medical device manufacturer wants to receive a form FDA-483 notice of inspectional observations and/or warning letter from the US FDA. FDA regulates the quality of the device manufacturing, monitors device problems and approves changes to device design that affect safety or efficacy.

FDA-483 form lists any observed deficiencies of the manufacturer's quality system. If the FDA inspector observes significant non-compliance during an inspection or if a company has consistently failed to address previous FDA inspection observations, the FDA may issue a warning letter.

A warning letter is strong notice to a manufacturer that its practices are deficient and must be immediately corrected. Warning letters are public [9], but both FDA-483s and warning letters require promptly and clear formal response to FDA. Failure to address issues in warning letter can lead to further FDA actions (e.g. product seizures, export restrictions, product recalls, etc.).

Failures to adequately validate computer software for its intended use according to an established protocol when computers or automated data processing systems are used as part of production or the quality system, as required by 21 CFR 820.70(i) [8] may result to receive FDA-483 form and/or warning letter.

The FDA is issuing more and more warning letters and recalling more and more medical devices with software defects. The FDA's analysis of 3140 medical devices recalls conducted between 1992 and 1998 reveals that 242 of them (7.7%) are attributable to software failures [5].

The role of the management in computer system validation is to set enterprise-wide validation policies

and provide resources for software validation activities (e.g. time, people, money, equipment, etc.).

2 Software validation process in medical device industry

Validation is a process of obtaining evidence and determining that a final software system meets the user's needs and expectations.

According to IEEE 610.12-1990 [22] a **validation** is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

FDA [5] defines **software validation** as confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled.

If the software can affect the product quality, it must be validated prior to initial use [1]. It is important to consider how a software failure would impact the medical device in production.

The validation of non-product software typically includes evidence that all user requirements have been fulfilled and implemented correctly and completely (see Figure 1). These requirements shall be traceable to software requirements and all verification/ validation activities shall be performed.

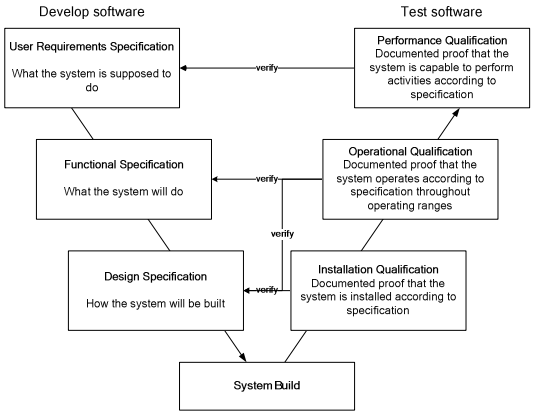

Figure 1. GAMP validation model [17]

2.1 Regulatory requirements for software validation

The computer system validation is not a new regulatory requirement and regulatory agencies (e.g. FDA, Notified Bodies for medical devices) are looking at computer systems during inspections/ external audits. Computer systems are evaluated as part of the quality system.

The computerized system includes hardware, software, peripheral devices, personnel and documentation (e.g. manuals, standard operating procedures) [10].

Software validation is a subset of computer system validation. Computer system validation also includes equipment and hardware qualification.

"EC Guide to Good Manufacturing Practice, Annex 11 Computerised Systems" [2] addresses the use of computerised systems in the pharmaceutical industry.

The FDA's "General Principles of Software Validation" [5] is very good reference for software verification and validation activities usually performed in the medical device industry.

21 Code of Federal Regulations Part 11 [4] is an FDA regulation that outlines the criteria for acceptance of electronic records and signatures. It applies to all industry segments regulated by the FDA (e.g. current Good Manufacturing Practices, Good Laboratory Practices and Good Clinical Practices).

GAMP4 [18] is a set of good automated manufacturing practices for validation of automated systems in the pharmaceutical and regulated life science industry.

The reference documents, guidelines and standards regarding requirements for software validation to consider are:

- EC Guide to Good Manufacturing Practice, Annex 11 Computerised Systems,
- EC Guide to Good Manufacturing Practice, Annex 15 Qualification and Validation [3],
- FDA 21 CFR Part 11 Electronic Records; Electronic Signatures – Scope and Application,
- FDA 21 CFR Part 820 Quality System Regulation,
- FDA General Principles of Software Validation,
- FDA Computerized Systems Used in Clinical Investigations [7],
- FDA Reviewers and Compliance on Off-The-Shelf Software Use in Medical Devices [6],
- GAMP4 (Good Automated Manufacturing Practices) Guide for Validation of Automated Systems,
- ISO 13485:2003 Medical devices – Quality management systems – Requirements for regulatory purposes [15],
- ISO/TR 14969 Medical devices – Quality management systems – Guidance on the application of ISO 13485:2003 [16].

2.1.1 FDA requirements

FDA software validation requirements are stated in paragraph §820.70(i) which addresses automated processes. This paragraph indicates that validation should focus on intended use of software in the automated system used as part of production or the quality system. All software changes shall be validated before approval and issuance [8].

Specific requirements for validation of medical device software [14] are found in paragraph

§820.30(g). FDA requires any medical device software product developed or acquired after June 1, 1997 to be subject to applicable design control provisions (21 CFR §820.30).

According to the FDA's General Principles of Software Validation Guidance document [5], the validation requirements apply to software used as components/integral parts in medical devices, to software that is itself a medical device (e.g. blood establishment software, medical image analysis, PACS system), and to software used in the production of a device or in implementation of the device manufacturer's quality system.

FDA 21 CFR Part 11 does not concern only the electronic signatures, but also electronic records, which most systems have, but may or may not be compliant with Part 11 requirements. 21 CFR Part 11 (effective since August 20, 1997) regulation provides criteria for acceptance of electronic records and signatures as equivalent to paper records and handwritten signatures on paper.

The FDA is looking very thoroughly at software validations during the inspection of the quality management system. If the process is software controlled, during an FDA inspection [11], the FDA investigator may:

- review firm's procedures for providing electronic and paper copies,
- review the overall security of electronic record keeping systems,
- review validation documentation (e.g. software requirements document, validation master plan, validation protocols, test cases and actual results, validation summary report, evidence of software version and change control, requirements, traceability matrix) to confirm that the software will meet user needs and is fit for its intended use,
- review any applicable vendor purchasing data for OTS software (i.e. vendor qualification),
- review training records of IT and technical employees,
- review company's corrective action plan and progress, etc.

Some examples of typical deficiencies observed by FDA in computer system validation and worth a warning letter [12], [13] are:

- no established software validation procedure,
- no adequate validation procedure for computerised spreadsheets used for in-process and finished product testing analytical calculations,
- failure to validate software in manufacturing,
- the electronic data did not correlate with the paper records,
- no software change control,
- no evaluation of the impact of software changes on other parts of the system,
- lack of appropriate controls to assure that changes in or deletion of records are done only by authorized personnel (i.e. audit trail capability),
- user access levels for the computer software were not established and documented,
- unrestricted access to server room,
- no documentation to ensure that the system operated as intended by the vendor and performed according to the user requirement specifications,
- no testing of the system after installation on the production environment,
- failure to conduct and/or document input/output checks of the computer system,
- no testing of the system at and outside the expected ranges,
- employees using computer system before training,
- no training records to indicate that employees are trained in the software and its applications, etc.

2.1.2 ISO 13485:2003 requirements

ISO 13485:2003 is a quality management system standard for medical device manufacturers.

To satisfy ISO 13485 software validation related requirements, a simple inventory and risk assessment of each software element used in the quality management system and in the manufacturing process is required.

ISO 13485 does not require compliance to FDA 21 CFR Part 11 requirements but any manufacturer in the medical device industry with the intent to market in the US must be aware of this requirement and comply. 21 CFR Part 11 requirements are more prescriptive. All computer systems running in an FDA regulated environment require Part 11 compliance, and therefore, need to be fully validated.

ISO 13485:2003 [15], clause "7.3.6 Design and development validation" refer to the validation of software in the product.

ISO 13485:2003 [15], clause "7.5.2.1 General requirements" refer to the validation of software used in production and service. "The organization shall establish documented procedures for the validation of the application of computer software (and changes to such software and/or its application) for production and service provision that affect the ability of the product to conform to specified requirements. Such software applications shall be validated prior to initial use." [15]

ISO 13485:2003 [15], clause "7.6 Control of monitoring and measuring devices" refer to the validation of software used in measurement. "When used in the monitoring and measurement of specified requirements, the ability of computer software to satisfy the intended application shall be confirmed. This shall be undertaken prior to initial use and reconfirmed as necessary." [15]

ISO/TR 14969:2004 [16] guidance on the application of ISO 13485:2003 explains the following software validation related requirements:

- 7.3.4.1 General n) If computer software has been used in design computations, modelling or analyses, has the software been appropriately validated, verified and placed under configuration control? [16]
- 7.3.4.1 General o) Have the inputs to such software, and the outputs, been appropriately verified and documented? [16]
- 7.5.2.1.3 Computer software used in process control – The requirements of ISO 13485 regarding the validation of the application of computer software used in process control apply, whether or not such software is purchased, developed, maintained, or modified for automated production or process control purposes. [16]
- 7.6.1 The requirements refer explicitly to monitoring and measuring devices, including test software. [16]
- 7.6.4 Software applications related to the control and/or calibration of monitoring and measuring devices should be validated. Examples include software used for a) controlling the instrument calibration process, b) determining the control or calibration status of instruments based on the data generated during the process, and c) scheduling the calibration of equipment, if the scheduling is not backed up by a manual (e.g. calibration label or other system). [16]

2.2 Software validation process and deliverables

According to FDA [5], planning, verification, testing, traceability, configuration management, and many other aspects of good software engineering (…) are important activities that together help to support a final conclusion that software is validated.

The content and sequence of software validation activities are defined in validation procedures.

The software validation process typically starts with the identification and prioritization of systems which needs to be validated based on a documented risk assessment.

User requirements (see Figure 1) need to be identified and prioritized too. User requirements are defined in user requirements specification (URS).

The next step is to define the system requirements (i.e. functional and non-functional) and develop specifications (e.g. system requirements specification, software requirements specification).

The validation master plan (VMP) is the key document in the validation process. It describes acceptance criteria, all required validation activities with assigned responsibilities, priorities and timings for actions.

The validation protocols (e.g. IQ – Installation Qualification, OQ – Operational Qualification, PQ – Performance Qualification) describe the procedure and the steps within the procedure that will be followed in order to validate a specific computer system.

Test report provides outcome of each test case defined in validation protocols. Outcome of all tests is summarized in the validation summary report.

Examples of potential automated processes that require software validation include computer systems with software related to the following processes:

- corrective and preventive action,
- internal and supplier audits,
- complaint handling,
- manufacturing process,
- labelling and packaging,
- distribution,
- process validation,
- design control,
- device tracking,
- monitoring of clinical trials,
- document control, etc.

Manufactures should document their rationale for those automated processes that do not require software validation.

Examples of computerized systems that may require validation include:

- manufacturing execution (MES) systems,
- electronic batch record systems (EBRS) for pharmaceutical companies,
- software that records and maintains the device history record (DHR) for medical device companies,
- enterprise resource planning (ERP) systems,
- laboratory information management systems (LIMS),
- programmable logic controllers (PLC) in manufacturing equipment,
- clinical data systems,
- maintenance and calibration systems,
- label print systems,
- packaging systems,
- training database systems,
- networks,
- Excel spreadsheets or databases used to track lots, corrective and preventive actions, customer complaints,
- adverse events reporting systems,
- statistical process control (SPC) systems,
- electronic document management systems (EDMS),
- software used to implement electronic signatures for documents required by regulation, etc.

Software validation deliverables for largest applications may include, but are not limited to:

- *System inventory list and assessment information* – used to determine which systems need to be validated,
- *Vendor assessment report*,

- *Hazard analysis*,
- *Project plan*,
- *Security plan*,
- *Project glossary*,
- *Software change request*,
- *Validation master plan (VMP)*,
- *Test plan*,
- *Validation strategy*,
- *Requirement specification*: *user requirements specification*, *system requirements specification*,
- *Detailed design specification* – usually written by the software vendor,
- *Technical architecture specification*,
- *Risk analysis/risk management plan*,
- *Traceability matrix*,
- *Code*,
- *Source code review report*,
- *Design review report*,
- *User manual* or *user instructions*,
- *Validation protocols (test cases)* – IQ, OQ and PQ protocols,
- *Test report*,
- *Deviation (issue) report*,
- *Validation summary report*,
- *Training plan*,
- *Training records*,
- *Data migration plan*, etc.

3 Validation methods

Very often is difficult to determine a clear method of exactly how to validate computer systems. Since computer systems usually operate in different production environments and software testing is not sufficient to establish software confidence, the effective validation requires a mixture of appropriate validation methods.

Selection of validation techniques should reflect the complexity of the software and the risk associated with the intended use of the software.

3.1 Installation qualification (IQ)

The installation qualification includes the verification of the installation requirements, equipment specifications and installation.

The IQ protocol should verify that the application (hardware/software) was installed correctly and that the necessary documentation (e.g. user manuals, work instructions, backup procedures, etc.) are in place to support key users.

3.2 Operational qualification (OQ)

The operational qualification is tied to the functional requirements specification.

All functional requirements and security shall be tested in the test environment. Testing may include unit tests and integration tests.

3.3 Performance qualification (PQ)

The performance qualification is based on the user requirements specification.

The PQ tests are completed on the entire system and require evidence for user acceptance. Therefore, trained key users of the system must be involved in PQ testing and in review and approval. The PQ shall also confirm that the procedures/work instructions developed for the users perform as expected.

3.4 Testing

The testing process must be systematic and clearly documented (e.g. test plan, test cases, test reports, test summary report). Poorly tested software may contain the following defects: allowed invalid entries, missing functionality, multi user conflicts, interface issues, security holes, etc.

The validation of software is not just testing. The detail of validation tests should be based on risk. Types of tests to run are: regression tests, system tests, database tests, stress tests, security tests, boundary tests, functional tests, IQ, OQ, PQ, etc.

3.4 FMEA

FMEA (Failure Mode and Effects Analysis) is a risk analysis method which can be applied to hardware and software. It identifies ways in which software can fail to meet critical requirements, estimates the risk of an unanticipated failure and helps to prioritize actions which should be taken to improve the software.

3.5 Reviews

Reviews (e.g. design reviews, code reviews) must be conducted by independent staff who is not directly involved in the software design or implementation.

4 Cost-effective validation approach

Software is made and used by humans who make mistakes, and therefore it is not perfect. Software failures increase with software complexity. Due to the software complexity, any change in software may have an impact on another software module or on the entire software system. Software failure in clinical applications can mean lost lives (e.g. infusion pumps delivering the wrong rate of medicine, pacemakers reset to unsafe parameters due to external radiation sources, Therac-25, etc.).

Validation of computerized systems can generate a lot of documentation (see 2.2), be very costly (e.g. validation of an ERP system) and time consuming for most companies, but also can provide business and regulatory benefits.

When implementing a complex ERP system (e.g. SAP, Oracle), a company must carefully document

the validation process to ensure compliance with the FDA's regulatory requirements.

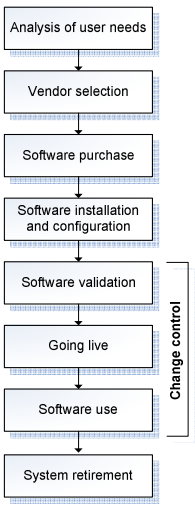

Figure 2. General OTS software acquisition process

4.1 Start validation with requirements

Requirements need to be easily understood by the users, testable and measurable. Documented user/software requirements are a starting point for verification testing and validation. Validation is difficult when poorly planned, so preparation for software validation should begin very early in the software lifecycle.

Not only functional requirements (per use cases), but also non-functional requirements for software (e.g. maintainability, security, portability, etc.) shall be documented, prioritized and tested too. Traceability matrix ensures all requirements have been defined and tested.

4.2 Supplier audit

When the non-product software is developed by software vendor and not by the FDA-regulated device manufacturer, supplier audit [20] may be required in order to assess the adequacy of the vendors' software verification and validation activities. According GAMP guideline for the computer validation [18], the decision whether to perform a supplier audit should be documented and based on a risk assessment and categorisation of the systems components.

Supplier audits for the non-product software which includes OTS software may be conducted by the device manufacturer or by a qualified third party [5]. Vendor quality management system must comply with regulatory requirements for software validation and the software vendor must have good documented specifications and executed tests. Validation effort regarding specification, IQ and OQ may be reduced by referring to vendor specification and qualification.

4.3 Risk-based validation

Performing a risk-based assessment helps to define the scope and focus of the computer system validation effort. Taking risk-based validation approach helps to save time, optimizes resources and results in a better validation.

The level of effort for software validation activities depends on the risks associated with the application (i.e. higher risk requires larger effort). Some computerized systems may not require software validation if the output of the automated process is 100% verified, reviewed or checked. For example, data generated by non-validated spreadsheets and queries can be used if verification or review by independent staff is performed and documented. However this could be more costly and time consuming than validation.

Computerized systems with the highest safety risk priority shall be included at the top in the validation master plan. The validation costs for safety-critical systems are usually significantly higher than for non-critical systems.

4.4 Use generic validation documents

Use of templates or generic documents with most text filled-out is useful to assist users in documenting validation and to save time in generating validation documentation.

4.5 Implement standard software

Open source software is developed by many people over many different enterprises which ultimately implicates many different user requirements. Regulations do not prevent use of open source software, but do require validation and documentation.

Implementation of pre-qualified standard software may significantly reduce validation effort.

4.6 Develop metrics

It is important to develop appropriate metrics (e.g. time spent, staff months, number of requirements, test coverage, number of errors found, etc.) to measure the progress of validation activities. Measurement is a prerequisite for process improvement.

4.7 Involve and train personnel

Experienced and trained QA and IT professionals should be included as strategic partners from the start of the validation project.

The members of the development or implementation team should be available to support IQ, OQ and PQ tests and correct any error identified

by users. Correction needs to be under change control (see Figure 2).

4.8 Involve system validation consultants

Manual software validation tests conducted by the in-house validation engineers or external consultants can double a company's compliance costs. Manual testing is slow, intensive and often causes inaccurate test results because it is being done by humans.

If the company is required to perform software validation of many systems in a short period of time, validation consultants can help to automate software validation process by implementing automated tools for validation and validation management.

If the company doesn't have trained software validation specialists, it is recommended to involve system validation consultants who provide validation expertise and experience with all types of computerized systems across the enterprise. Their job is to help ensure regulatory compliance with software validation requirements and reduce overall project costs. System validation consultants can help to develop software validation policies and procedures, to execute software tests and document test results (e.g. test plan, test protocols, test scripts, test reports), etc. They may also provide the following services: training on computer system validation, conduct software vendor audits, computer system implementation support, legacy system validation, Part 11 compliance, etc.

4.9 Validation package

Many software vendors provide validation module/package for clients in regulated industries. The validation package usually includes a standard set of validation documentation and a suite of installation and qualification services. The use of validation package can significantly reduce the validation costs and effort.

4.10 Continuously improve software validation process

Software quality assurance should act proactive and focus on preventing software defects (i.e. errors are caught and corrected before going live).

Too complex software validation process can make implementation of new systems slow and difficult. Internal audit is a good tool to identify weak points (i.e. areas for improvement) in the validation in order to improve the validation process over time.

5 Benefits of software validation

If a company is not regulated by the FDA, this does not mean computer systems should not be validated. Computer system validation should be part of any good business practice.

Some benefits of computer system validation are the following:

- ensures that the software meets the customer needs,
- reduces legal liability risk for device manufacturers (i.e. reduces the risk of a failure that could result in a patient harm),
- can increase the usability and reliability of the device [5],
- reduces regulatory risk,
- ensures accurate, reliable and consistent automated processes,
- provides validation documentation required by FDA and other regulatory agencies (e.g. Health Canada, TÜV, European Economic Community, etc.),
- reduces labor costs in the long run by increasing employee efficiency (i.e. reducing 100% verification checks),
- ensures return on investment (ROI) because validation discovers costly defects and failures early, before the system is used in production environment,
- good business practice – computer systems and their output are corporate assets,
- promotes continuous process improvement – software is used to automate and improve processes,
- in combination with project management ensures projects are implemented on schedule and on budget,
- easier maintenance, etc.

6 Conclusion

Medical device manufacturers are operating in a highly regulated industry. Companies must satisfy the regulatory and management requirements for cost-effective validation.

There is no one software validation model that is the best for all organizations, and each organization should find a model that works best within the organization.

Selection of cost-effective validation strategies/ techniques should reflect the complexity of the non-product software and the risk associated with the intended use of the software (e.g. system risk to product safety, efficacy and quality; system risk to data integrity, authenticity and confidentiality).

7 Acknowledgments

I would like to thank Peter Rattke and Kerri-Anne Bowers for their helpful comments and thoughtful review. I am very grateful to Professor Dr. Zdravko Krakar for his support, inspiration and guidance.

References

[1] Brower, G.N.: **Validation of Commercial-Off-the-Shelf (COTS) Software**, Journal of Validation Technology, Vol. 5, No. 4, 1999, pp. 318-323.

[2] European Commission: **EC Guidelines to Good Manufacturing Practice, Annex 11 Computerised Systems**, Brussels, 2003.

[3] European Commission: **EC Guidelines to Good Manufacturing Practice, Annex 15 Qualification and validation**, Brussels, 2001.

[4] Food and Drug Administration et al.: **Guidance for Industry Part 11, Electronic Records; Electronic Signatures – Scope and Application**, USA, 2003.

[5] Food and Drug Administration, Center for Devices and Radiological Health: **General Principles of Software Validation; Final Guidance for Industry and FDA Staff**, USA, 2002.

[6] Food and Drug Administration, Center for Devices and Radiological Health: **Guidance for Industry, FDA Reviewers and Compliance on Off-The-Shelf Software Use in Medical Devices**, USA, 1999.

[7] Food and Drug Administration, Office of the Commissioner: **Guidance for Industry Computerized Systems Used in Clinical Investigations**, USA, 2007.

[8] Food and Drug Administration: **FDA 21 CFR Part 820 Quality System Regulation**, USA, 2006, pp. 138-151.

[9] Food and Drug Administration: **FDA's Electronic Freedom of Information Reading Room – Warning Letters and Responses**, available at `http://www.fda.gov/foi/warning.htm`, Accessed: 22nd May 2008.

[10] Food and Drug Administration: **Glossary of Computerized System and Software Development Terminology**, available at `http://www.fda.gov/ora/Inspect_ref/igs/gloss.html`, Accessed: 22nd May 2008.

[11] Food and Drug Administration: **Guide to Inspections of Quality Systems**, 1999, available at `http://www.fda.gov/ora/inspect_ref/igs/qsit/QSITGUIDE.PDF`, Accessed: 25th May 2008.

[12] Food and Drug Administration: **Warning Letter**, 2008, available at `http://www.fda.gov/foi/warning_letters/s6741c.htm`, Accessed: 18th May 2008.

[13] Food and Drug Administration: **Warning Letter**, 2008, available at `http://www.fda.gov/foi/warning_letters/s6730c.htm`, Accessed: 18th May 2008.

[14] International Electrotechnical Commission: **CEI/IEC 62304:2006 Medical device software – Software life cycle processes**, Geneva, 2006.

[15] International Organization for Standardization: **ISO 13485:2003 Medical devices – Quality management systems – Requirements for regulatory purposes**, 2nd edition, Geneva, 2003.

[16] International Organization for Standardization: **ISO/TR 14969 Medical devices – Quality management systems – Guidance on the application of ISO 13485:2003**, Geneva, 2004.

[17] International Society for Pharmaceutical Engineering: **GAMP Good Practice Guide: Testing of GxP Systems**, ISPE, 2005.

[18] International Society for Pharmaceutical Engineering: **GAMP-Leitfaden zur Validierung automatisierter Systeme**, Version 4, 2001.

[19] Lohrey, K: **Cost and Benefit Analysis of Validation Strategies**, Pharmaceutical Engineering, Vol. 27, No. 2, 2007, pp. 1-8.

[20] Miller, A.: **COTS Software Supplier Identification and Evaluation**, In Proceedings of the RTO IST Symposium on Commercial Off-The-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS", paper published in RTP MP-48, Brussels, Belgium, 2000.

[21] Phan, T.T.: **Technical Considerations for the Validation of Electronic Spreadsheets for Complying with 21 CFR Part 11**, Pharmaceutical Technology, No. January 2003, 2003, pp. 50-62.

[22] The Institute of Electrical and Electronics Engineers, Inc.: **IEEE Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology**, IEEE Computer Society Press, New York, 1990.