Improving Verification & Validation in the Medical Device Domain

M.S. Sivakumar, Valentine Casey, Fergal McCaffery, Gerry Coleman

Regulated Software Research Group,
Dundalk Institute of Technology & Lero, Ireland

{Sivakumar.Madhavakuruppu, Val.Casey, Fergal.McCaffery, Gerry. Coleman}@dkit.ie

Abstract

The benefits of effective verification and validation activities in the medical device domain include increased usability and reliability, decreased failure rate and recalls and reduced risks to patients and users. Though there is guidance on verification and validation in multiple standards in the medical device domain, these are difficult for the manufacturer to implement, as there is no consolidated information on how they can be successfully achieved. The paper highlights three major areas for improvement in the medical device software development domain. This research is based on an analysis of available literature in the field of verification and validation in generic software development, safety-critical and medical device software domains. Additionally, we also performed a review of the standards and process improvement models available in these domains.

Keywords: Medical device standards, Medical device software verification and validation, V&V, Medical device software process assessment and improvement, Medi SPICE

1. Introduction

Verification and Validation (V&V) are amongst the most important activities in the software development lifecycle and consume up to 50% of project development time [1], [2] and up to 50% of the total cost [3]. The concepts of V&V emerged in the late 1960s and 1970s as the use of software in military and nuclear power systems increased. Initially, individual projects began to develop their own standards to address the need for V&V. Then government and industry began to develop V&V standards so they would have a specified approach for contract procurement and for monitoring the technical performance of V&V efforts. V&V standards and guidelines serve large, heterogeneous communities and are applicable to many types of software. These include the following:

1. The US Air Force's AFSCIAFLC 600-5, Software Independent Verification and Validation [4]

2. The American Nuclear Society's ANS 10.4, Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry [4]

3. The NASA Jet Propulsion Laboratory's JPL D-576, Independent Verification and Validation of Computer Software: Methodology [4]

4. The Food and Drug Administration (FDA) Center for Devices and Radiological Health (2002) General Principles of Software Validation (GPSV) [5] for medical devices

While both V&V play a key role in software development, there is a level of ambiguity in the use of these terms. This is evident in the literature as the following definitions demonstrate:

- Berling, and Thelin argue that verification aims at checking that the system as a whole works according to its specifications and validation aims at checking that the system behaves according to the customers' intentions [6]

- Arthur, Groner, Hayhurst, and Holloway cite in one of their papers that verification refers to the process of examining each development phase to ensure that the output of a particular phase satisfies all the pertinent requirements of the previous phase, is internally acceptable, and can support the development effort in the next phase. Validation, on the other hand, is an activity primarily concerned with software testing. During validation you execute the system and compare the test results to the requirements [7]

- Wallace and Fujii state that verification involves evaluating software during each lifecycle phase to ensure that it meets the requirements set forth in the previous phase and validation involves testing software or its specification at the end of the development effort to ensure that it meets its requirements (that it does what it is supposed to). They go on to say that while 'verification" and "validation" have separate definitions, you can derive the maximum benefit by using them synergistically and treating "V&V" as an integrated definition. [4]

As these definitions highlight, the terms verification and validation are often used interchangeably whereas in the strict sense the objectives of these two processes are different. While they may both employ the same techniques or environment while performing these processes.

The main activities of verification and validation are reviews and testing. Software testing cannot be conducted until the software product is built. Since faults need to be found early, software inspections are conducted before the product has been implemented [6]. The difference between validation and verification can be explained by looking at the purpose of the tests performed. The use of prototypes to test if requirements can be addressed is an example of a verification practice. The evaluation of a prototype by the users to test if the product fulfills their needs is an example of a validation practice. In other words, we can say that the verification should ask "Are you meeting the specified requirements?" and "Are you building the product right?" In the same way we can say that the validation should ask "Are you meeting the operational need?" "Does this product meet its intended use in the intended environment?" and "Are you building the right product?" [8]. ANSI/IEEE Std 1012 [9], Standard for Software Verification and Validation Plans provides a set of minimum V&V tasks for each of the lifecycle phases in a software project.

2. Research Approach

We performed our research to undertake this study, the following phases:

- The V&V processes were reviewed in detail to understand how they were addressed in the context of generic software development. The review included relevant literature, standards and process improvement models such as Capability Maturity Model® Integrated (CMMI®) [10], and ISO/IEC 15504-5 [11].

- Research was undertaken in the domain of safety-critical software development and V&V in this context. This included conducting a review of how the V&V processes were addressed by safety-critical software development standards such as DO-178B [12] and Automotive SPICE [13]

- Medical device software V&V was extensively researched. To this end a detailed analysis of the medical device standards was undertaken which included ISO/IEC 13485 [14], ISO/IEC 62304 [15], ISO/IEC 14971 [16] and the Food and Drug Administration (FDA) Guidelines for Software Validation (FDA GPSV).

- From the comparative analysis performed, we arrived at a set of findings which suggest areas for improving V&V practices in the medical device software domain

3. Verification and Validation in Generic Software Development

Two important reference models which are widely used in the context of software process improvement are the Capability Maturity Model® Integrated (CMMI®) and ISO/IEC 15504-5. They address the software verification and validation processes in the following ways: CMMI® in line with ANSI/IEEE Std 1012-1986 recommends a lifecycle view for verification and validation activities. It defines verification as "Confirmation that work products properly reflect the requirements specified for them". In other words, verification ensures that 'you built it right' and validation as "Confirmation that the product, as provided (or as it will be provided), will fulfill its intended use". Therefore, validation ensures that 'you built the right product'. The verification and validation processes are part of the engineering processes category and both are level 3 process areas in the staged model.

The verification process area in CMMI® is used by many of the other process areas irrespective of the category they fall into. These processes include Project Planning, Measurement & Analysis and other support process areas. The verification process is used to verify the work products created from the performance of these processes. Verification is also used extensively in the context of the engineering process areas which includes Requirements Management, Requirements Development, Technical Solution and Product Integration. The model also provides guidance in terms of examples of methods like peer reviews, statement coverage testing, and branch coverage testing that could be used in this context.

The validation process area incrementally validates products against the customer's needs. Validation may be performed in the operational environment or simulated operational environment. Coordination with the customer on the validation requirements is an important element of this process area. The scope of the validation process area includes validation of products, product components, selected intermediate work products, and processes. These validated elements may often require re-verification and revalidation.

Validation activities can be applied to all aspects of the product in any of its intended environments, such as operation, training, manufacturing, maintenance, and support services. Like the verification process area, validation is also performed during the course of the product development as it moves from each phase of the lifecycle. For example in the requirements phase, the model suggests analysis, simulations, prototyping and demonstrations as possible techniques for validation. Both validation and verification activities often run concurrently and may use portions of the same environment.

Similar to CMMI®, verification and validation have two distinct processes in ISO/IEC 15504-5. Verification and validation are under the supporting life cycle process category and the source for both these processes are   ISO/IEC 12207 AMD1 [17].

In ISO/IEC 15504-5, the purpose of the verification process is to confirm that each software work product and/or service of a process or project properly reflects the specified requirements. The tasks pertaining to verification include: development of a verification strategy, development of criteria for verification, performing the activity of verification, determination of actions based on verification results and making the results available to the stakeholders.

As per ISO/IEC 15504-5, the purpose of the validation process is to confirm that the requirements for a specific intended use of the software work product are fulfilled. Validation aims to confirm by examination and provision of objective evidence that software or system specifications conform to user needs and intended uses, and the particular requirements implemented by the software product can be consistently fulfilled.   The tasks pertaining to validation include: the development of a validation strategy and criteria for validation, performing the validation activities and the identification of problems, providing the validation data and making the results available to customers and other stakeholders.

The state of adoption of formal process management for V&V varies across organizations depending on the size and nature of business/products being developed. A study on the state of practice of V&V in industry [18] reveals the following: the process is more emphasized in large organizations while smaller organizations have identified inconsistencies in the documentation standard, but did not consider non-adherence to processes as being too harmful.

Though CMMI®, pays attention to V&V, it is still rather modest in its focus on these areas compared to other elements of the development processes [19]. As stated, industry-wide experiences indicate that validation and verification activities typically consume about 30-50% of development budgets [20]. In May of 2007, a working group was formed by ISO/IEC to produce a new software testing standard (the ISO/IEC 29119). The development of the standard is expected to be completed by May - October 2012 [21].

Our analysis points to the fact that the general software development standards and process improvement models need to be adapted to meet the specific requirements of medical device software development, which is highly regulated and safety-critical in nature.

4. Verification and Validation in Safety-critical Domains

In order to review the V&V processes for software development in the safety-critical medical device domain, a relevant approach was to determine how verification and validation is addressed in the safety-critical software development domains of aerospace, avionics, space, nuclear and automotive. As with medical devices, software in these areas is a critical element of complex, potentially dangerous problems and failures in these products can result in loss of life, significant environmental damage, and major financial loss [22].

In the development of safety-critical systems, the speed of technology and rising demand for improved productivity has created a host of new challenges. These include issues such as, known safety techniques are not applied, ability to demonstrate (certify) that safety requirements have been met is inadequate, and automated code generators, automated testing tools etc. which are used to improve productivity may have associated problems which may be poorly understood [23]. In addition, designed products are required to meet a very high-level of reliability, security, and performance in safety-critical areas as many of them could pose serious threat to users/consumers or the general public. Therefore, ensuring that such systems meet their predefined requirements and that they perform as expected is a challenging issue [24].

It has been found that there is a relationship between the increasing occurrence of system accidents and the increasing usage of software [25]. Within the safety-critical software arena, different standards/certifications are available for different industries. These include the DO-178B for Aerospace, and Automotive SPICE and ISO 26262 [26] in the Automotive industry. IEC 60880 [27] describes the European standards for certification of nuclear power generating software. IEC 61508 [28] describes a general-purpose hierarchy of safety-critical development methodologies that has been applied to a variety of domains ranging from medical instrumentation to electronic switching of passenger railways. Though these standards address verification and validation in sufficient detail, their role is not to address process improvement. In addition, there are some authors [29] who consider that V&V assessment in CMMI® is not adequate when dealing with safety-critical software, and they proposed a new framework for V&V assessment, focused on safety-criticality. This framework is defined using safety standards together with the CMMI® V&V process areas and the ISO 9001 standard [30].

Some of the key differences between safety-critical standards and generic-software development standards/models highlighted in our research were as follows:

a. Risk Management

Risk Management involves the identification and management of risk. Risk assessment is a function of impact and the probability of occurrence. A risk based approach to safety allows the hazards associated with a system to be identified and prioritized. The risk assessment involves calculation of level of risk associated with a hazard.

b. Integrity levels

Several standards use the concept of integrity levels to define the application of effort and rigour appropriate to the criticality of a component. In DO-178B, there are software levels starting from A to E (A being highest and E being lowest). ISO 26262 assigns the value of the ASILs (Automotive Safety Integrity Levels) from D (most critical), to A (least critical), based on IEC 61508

c. Stress on Independence

Independence in verification and validation is important because the developer of the software product may have a biased opinion about their own product and therefore possibly miss performing certain verification/validation activities.

d. Qualification of tools [31]

Section 12.2 of DO-178B, specifies that the objective qualification of tools used in development (which includes verification and validation tools) is to ensure that the tool provides confidence at least equivalent to that of the processes eliminated, reduced, or automated.

We examined these factors for the purpose of identifying a set of improvements for the medical device software domain.

5. Improvement in V&V in the Medical Device Software Domain

The applicable standards in the problem domain of medical device software development are ISO 13485, ISO/IEC 62304, ISO/IEC 14971 and FDA GPSV. We analysed how the areas of verification and validation are addressed by these standards in detail as well as in the relevant literature. We then compared our findings to standards and process improvement models in general software development and other safety-critical domains, which included the aerospace and automotive industry. As a result of this analysis we identified three key areas which need to be addressed for improvement in the context of verification and validation for medical device software development. These are as follows:

5.1 Having Distinct Processes for Verification and Validation

ISO 13485 represents the requirements for a comprehensive management system for the design and manufacture of medical devices. As per ISO 13485, verification is performed to ensure that the design and development outputs have met the design and development input requirements. Design and development validation is carried out to ensure that the resulting product is capable of meeting the requirements for the specified application as defined in relation to its intended use. Validation should be completed prior to the delivery or implementation of the product. However, the level of detail available for verification and validation in ISO 13485 is not sufficient for an effective implementation of these processes in the context of medical device software.

To address this, manufacturers rely on ISO/IEC 62304, which provides a framework of life cycle processes with activities and tasks necessary for the development and maintenance of medical device software. In this context the standard defines the software development lifecycle as a conceptual structure spanning the life of the software from the definition of its requirements to its release and maintenance. In ISO/IEC 62304, verification is confirmed through the provision of objective evidence that the specified requirements have been fulfilled. ISO/IEC 62304 identifies the milestones at which the completeness of specified deliverables are verified thus stressing the importance of verification. ISO/IEC 62304 expects that verification will be covered as part of each lifecycle activity from requirements to software release. Though the standard does address software verification, it considers validation to be a system level process and is therefore outside its scope even when the system is entirely software.

The FDA's GPSV outlines the general verification and validation principles that the FDA considers applicable to the validation of medical device software or the validation of software used to design, develop, or manufacture medical devices. As per the FDA GPSV, software verification provides objective evidence that the design outputs of a particular phase of the software development lifecycle meets all of the specified requirements for that phase. Software verification looks for consistency, completeness, and correctness of the software and its supporting documentation, as it is being developed, and provides support for a subsequent conclusion that software is validated. Even though validation is defined in Section 3.1.2 of the FDA GPSV, it is not covered thereafter in the guidelines.

ISO 14971 is a standard for performing risk based activities in the medical device domain. ISO 14971 states that verification is "confirmation, through the provision of objective evidence, that specified requirements have been fulfilled". Two distinct types of verifications are required by the standard: (1) Verification to confirm that the risk control measure has been implemented in the final design. (2) Verification to ensure that the measure as implemented actually reduces the risk. In some instances, a validation study can be used for verifying the effectiveness of the risk control measure.

5.1.1 Improving Verification and Validation through separation of these two processes

Since ISO 14971 does not provide guidance on the verification and validation activities that should be performed manufacturers rely on ISO 13485 and ISO/IEC 62304 for implementation of verification and validation. As a result it can be difficult for the software manufacturers to meet the requirements of the risk management standard effectively.

As the processes of V&V are critical to software development in the medical devices domain, there needs to be a mechanism by which manufacturers could assess where they stand with respect to these processes and identify possible improvement opportunities. Manufacturers would also require a benchmark in terms of verification and validation processes to evaluate a potential vendor's capability in these processes. Hence it becomes necessary that a set of distinct processes be defined for verification and validation. Further, manufacturers would greatly benefit from the definition of a set of practices for verification and validation which relate to each of the engineering processes.

5.2 Independence in Verification and Validation

The IEEE Standards for Software Verification and Validation state that classical Independent Verification and Validation (IV&V) is generally required for the development of software systems deemed "critical" in nature, i.e., those which can result in loss of life, mission or significant social or financial loss [32]. The results of a study [7] highlight the difference in fault detection capabilities between two methods – non-independent V&V and Independent V&V (IV&V). The results indicate that IV&V provided a significant value-added component to the software development process. Independence is an important factor addressed by DO-178B and gives specific guidance on the subject.

ISO/IEC15504-5 and Automotive SPICE state that "degrees of independence" is something each project has to plan as part of its verification and validation strategy. The FDA GPSV addresses independence in Section 4.9, but leaves it to the discretion of device manufacturers on how this is to be achieved. ISO/IEC 62304 does not mandate independence. Independence is addressed in ISO 13485 in section 5.5.1 - Responsibility and Authority, where it states: "Top management shall establish the interrelation of all personnel who manage, perform and verify work affecting quality, and shall ensure the independence and authority necessary to perform these tasks".

5.2.1 Improving Verification and Validation by making them independent of development

Section F.3 of ISO 14971 states that the risk management plan should identify the personnel with responsibility for the execution of specific risk management activities, for example, reviewer(s), expert(s), independent verification specialist(s), individual(s) with approval authority (see 3.2). This assignment can be included in a resource allocation matrix defined for the design project. From the statement, we can assume that the standard recognizes the need for independent verification specialist(s). Since ISO 14971 relies on other standards like the ISO 13485 for implementation of verification and validation, the manufacturers may have to look for additional guidance for implementation. As we have seen in the case of ISO 13485 and the FDA GPSV, though independence in verification and validation is addressed implicitly, there needs to be further guidance on which phases/engineering activities would require independence in verification and validation.

5.3 Qualification of Tools

The tools to be used in software development should be evaluated to ensure they adequately address the requirements for which they are being procured. In a safety-critical domain like Aerospace, the software verification process objectives for software development tools are described in the DO-178B standard. Tools are classified into software development tools and software verification tools. The qualification criteria for software verification tools are specifically addressed in the standard. ISO 13485 and ISO/IEC 62304 do not address this requirement, which is essential for safety-critical software development.

In terms of qualification of tools in software development and verification and validation activities, the FDA GPSV states that most of the automated equipment and systems used by device manufacturers are supplied by third party vendors and are purchased Off-The-Shelf (OTS). It gives guidance on validation of OTS software and automated equipment in section 6.3. The standard states that the vendor's life cycle documentation, such as testing protocols and results, source code, design specification, and requirements specification can be useful in establishing that the software has been validated. The standard however states that documentation may not be available from the product vendors as they might not be willing to share proprietary information. The standard suggests a number of practices which could be adopted by the device manufacturer given the lack of available documentation from the vendors.

5.3.1 Improving Verification and Validation by using appropriate tools

Given what the standards propose, qualification of tools in the software development and V&V context needs to be addressed consistently in the medical device domain.

Due to delivery schedule pressures and shortfall in properly trained software engineers, the development of medical device software and systems has not kept pace with software assurance techniques practiced in other safety-critical domains such as avionics [33]. Apart from this, the lack of standard architectures for medical device software and the lack of integrated standard principled engineering tool support for analyzing software have created challenges to be dealt with [34].

Thus, there is a need for the development of an integrated model, which incorporates best practice and regulatory compliance, which is domain specific to medical device software development. To address this Medi SPICE [35], comprising of a process reference model and a process assessment model for software development in the medical devices domain is being developed by the authors. Medi SPICE brings together the best practice from 15504-5 and the regulatory requirements of the medical device standards which include ISO 13485, ISO/IEC 62304, and ISO 14971and FDA GPSV. Medi SPICE will also incorporate and address the findings from this research. The authors believe that with the publication of Medi SPICE more specific guidance will be available for process design, assessment, and improvement in the medical device software industry [35].

6. Conclusions

The areas we have identified for improvement will provide guidance for the definition of the V&V processes in Medi SPICE. Further to the definition of a set of processes and the associated practices related to V&V, the processes should be piloted in organizations within the medical device software development industry. Based on the results obtained and the feedback from the medical device software development industry, the processes will be evaluated and continuously improved.

7. Acknowledgments

This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/I1299, the SFI Principal Investigator Programme, grant number 08/IN.1/I2030 (the funding of this project was awarded by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund), and supported in part by Lero - the Irish Software Engineering Research Centre (http://www.lero.ie) grant 03/CE2/I303_1

8. References

1. E. Bernard, B. Legeard, X. Luck and F. Peureux.: “Generation of test sequences from formal specifications: GSM 11-11 standard case study,” Software: Practice and Experience, vol. 34, no. 10, 2004, pp. 915-948

2. S. Rayadurgam and M. P. E. Heimdahl.: “Test-sequence generation from formal requirement models,” Proc. High Assurance Systems Engineering, 2001. Sixth IEEE International Symposium on, 2001, pp. 23-31

3. G. Myers, C. Sandler, T. Badgett and T. Thomas.: The Art of Software Testing, Second Edition, Wiley (2004)

4. Wallace, D.R.; Fujii, R.U.: "Software verification and validation: an overview," Software, IEEE , vol.6, no.3, pp.10-17, May 1989

5. US FDA Center for Devices and Radiological Health (2002) General Principles of Software Validation; Final Guidance for Industry and FDA Staff. CDRH, Rockville

6. Berling, T.; Thelin, T..:, "An industrial case study of the verification and validation activities," Software Metrics Symposium, 2003. Proceedings. Ninth International , vol., no., pp. 226- 238, 3-5 Sept. 2003

7. Arthur, J.D.; Groner, M.K.; Hayhurst, K.J.; Holloway, C.M.: "Evaluating the effectiveness of independent verification and validation," Computer , vol.32, no.10, pp.79-83, Oct 1999

8. Paula Monteiro, Ricardo J. Machado, Rick Kazman.: "Inception of Software Validation and Verification Practices within CMMI Level 2," icsea, pp.536-541, 2009 Fourth International Conference on Software Engineering Advances, 2009

9. IEEE Standard for Software Verification and Validation Plans," IEEE Std 1012-1986 , vol., no., pp.I, 1986

10. CMMI Product Team (2006) Capability Maturity Model® Integration for Development Version 1.2. Software Engineering Institute

11. ISO/IEC 15504-5:2006 (2006) Information technology — Process Assessment — Part 5: An Exemplar Process Assessment Model. ISO, Geneva, Switzerland

12. RTCA, RTCA DO-178B, Software Considerations in Airborne Systems and Equipment Certification, 1992

13. Automotive SPICE Process Assessment Model 2.2, Automotive SIG (21 August 2005).

14. ISO 13485:2003 (2003) Medical devices — Quality management systems — Requirements for regulatory purposes. Second edn. ISO, Geneva, Switzerland

15. ANSI/AAMI/IEC 62304:2006 (2006) Medical device software—Software life cycle processes. AAMI, Arlington

16. ISO 14971:2007 (2007) Medical Devices — Application of risk management to medical devices. Second edn. ISO, Geneva

17. ISO/IEC 12207:1995/Amd.1 (2002) Information Technology — Software life Cycle Processes Amendment 1. ISO, Geneva, Switzerland

18. Andersson, C.; Runeson, P. .: "Verification and validation in industry - a qualitative survey on the state of practice," Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n , vol., no., pp. 37- 47, 2002

19. Jacobs, J.C.; Trienekens, J.J.M.:, "Improving verification and validation in hardware/software environments. Introduction to the workshop," Software Technology and Engineering Practice, 2002. STEP 2002. Proceedings. 10th International Workshop on , vol., no., pp. 121- 122, 6-8 Oct. 2002

20. E. Kit, , Software Testing in the Real World, Addison-Wesley (1995)

21. ISO/IEC 29119 Software Testing, http://www.softwaretestingstandard.org/projecttimeline.php (Accessed on 18 April 2011)

22. Why Software Requirements Traceability Remains a Challenge, Cross Talk Jul/Aug 2009 Issue, http://www.crosstalkonline.org/storage/issue-archives/2009/200907/200907-Kannenberg.pdf (Accessed on 18 April 2011)

23. Page 1, Heimdahl, M.P.E..:, "Safety and Software Intensive Systems: Challenges Old and New," Future of Software Engineering, 2007. FOSE '07 , vol., no., pp.137-152, 23-25 May 2007

24. Alawneh, L.; Debbabi, M.; Hassaine, F.; Jarraya, Y.; Soeanu, A.: , "A unified approach for verification and validation of systems and software engineering models," Engineering of Computer Based Systems, 2006. ECBS 2006. 13th Annual IEEE International Symposium and Workshop on , vol., no., pp.10 pp.-418, 27-30 March 2006

25. Paper on System Safety in Computer-Controlled Automotive Systems - Nancy G. Leveson, MIT, http://sunnyday.mit.edu/papers/sae.pdf (Accessed on 18 April 2011)

26. ISO/DIS 26262 Road vehicles - Functional safety. ISO, Geneva, Switzerland

27. IEC 60880:2006 (2006) Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions. IEC, Geneva, Switzerland

28. IEC/TR 61508:2005 (2005) Functional safety of electrical/electronic/ programmable electronic safety related systems. BSI, London

29. A. Y. Kyung, P. Seung-Hun, B. Doo-Hwan, C. Hoon-Seon and J. Jae-Cheon:, A Framework for the V&V Capability Assessment Focused on the Safety-Criticality, Proc. Software Technology and Engineering Practice, 2005. 13th IEEE International Workshop on, 2005, pp. 17-24

30. “ISO 9001:2000 - Quality management systems - Requirements,” 2000

31. K. A. Eastaughffe, A. Cant, M.A. Ozols.: A Framework for Assessing Standards for Safety-critical Computer-Based Systems, isess, pp.33, Fourth IEEE International Symposium and Forum on Software Engineering Standards, 1999

32. IEEE 1998. IEEE Standards for Software Verification and Validation, Page 58

33. Insup Lee, George J. Pappas, Rance Cleaveland, John Hatcliff, Bruce H. Krogh, Peter Lee, Harvey Rubin, Lui Sha.: High-Confidence Medical Device Software and Systems,"Computer, vol. 39, no. 4, pp. 33-38, Apr. 2006

34. Jetley, R.; Iyer, S.P.; Jones, P.L.; Spees, W.; , "A Formal Approach to Pre-Market Review for Medical Device Software," Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International , vol.1, no., pp.169-177, 17-21 Sept. 2006

35. Medi SPICE: "An Update", Fergal Mc Caffery, Alec Dorling, Valentine Casey, the 10th International Spice Conference (SPICE 2010), 18th to 20th May 2010, Pisa, Italy