Metadata of the chapter that will be visualized in SpringerLink

Book Title	Systems, Software and Services Process Improvement
Series Title	
Chapter Title	A Framework for Taxonomy Based Testing Using Classification of Defects in Health Software-SW91
Copyright Year	2019
Copyright HolderName	Springer Nature Switzerland AG

Corresponding Author	Family Name	**Rajaram**
	Particle	
	Given Name	**Hamsini Ketheswarasarma**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Dundalk Institute of Technology
	Address	Dundalk, Ireland
	Email	Hamsini.Ketheswarasarma@dkit.ie
	ORCID	http://orcid.org/0000-0002-3294-3906
Author	Family Name	**Loane**
	Particle	
	Given Name	**John**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Dundalk Institute of Technology
	Address	Dundalk, Ireland
	Email	
	ORCID	http://orcid.org/0000-0002-9285-5019
Author	Family Name	**MacMahon**
	Particle	
	Given Name	**Silvana Togneri**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Dundalk Institute of Technology
	Address	Dundalk, Ireland
	Email	
	ORCID	http://orcid.org/0000-0003-0179-2436
Author	Family Name	**McCaffery**
	Particle	

Given Name	**Fergal**
Prefix	
Suffix	
Role	
Division	
Organization	Dundalk Institute of Technology
Address	Dundalk, Ireland
Email	
ORCID	http://orcid.org/0000-0002-0839-8362

Abstract	Taxonomy based testing is an efficient approach to find software defects at earlier phases of medical device software development. It allows the creation of goal oriented test cases while brainstorming test ideas. This approach is adaptable into standard testing processes such as the ISTQB and the 29119-2. It uses a new standard, a defect taxonomy SW91 which is identified as a consensus standard by the US Food and Drug Administration (FDA). This paper presents steps to be followed in implementing taxonomy based testing at a medical device software organisation which follows the software development process explained in IEC 62304:2006+A1:2015. Finally, the future work section explains how this framework can be tailored to testing techniques and how its efficiency and benefits will be evaluated via expert reviews and implementation at medical device software organisations.

Keywords (separated by '-')	Taxonomy based testing - Defect taxonomy - Medical device software - IEC 62304:2006+A1:2015 - ISTQB - 29119-2

A Framework for Taxonomy Based Testing Using Classification of Defects in Health Software-SW91

Hamsini Ketheswarasarma Rajaram(✉), John Loane, Silvana Togneri MacMahon, and Fergal McCaffery

Dundalk Institute of Technology, Dundalk, Ireland
Hamsini.Ketheswarasarma@dkit.ie

Abstract. Taxonomy based testing is an efficient approach to find software defects at earlier phases of medical device software development. It allows the creation of goal oriented test cases while brainstorming test ideas. This approach is adaptable into standard testing processes such as the ISTQB and the 29119-2. AQ1
It uses a new standard, a defect taxonomy SW91 which is identified as a consensus standard by the US Food and Drug Administration (FDA). This paper presents steps to be followed in implementing taxonomy based testing at a medical device software organisation which follows the software development process explained in IEC 62304:2006+A1:2015. Finally, the future work section AQ2
explains how this framework can be tailored to testing techniques and how its efficiency and benefits will be evaluated via expert reviews and implementation at medical device software organisations.

Keywords: Taxonomy based testing · Defect taxonomy · Medical device software · IEC 62304:2006+A1:2015 · ISTQB · 29119-2

1 Introduction

A defect taxonomy is a system of hierarchical categories designed to be a useful aid for reproducibly classifying defects in the software development lifecycle [1]. There are many defect taxonomies available such as the orthogonal defect classification scheme, IEEE's defect classification scheme and HP's defect classification scheme. Researchers have found many benefits of applying defect taxonomies in different domains such as the safety critical domain, the business domain and the telecommunications domain [2, 3]. Defect taxonomies have been successfully used in software testing and benefits such as early detection of defects, brainstorming test ideas, maximising test coverage and creating goal oriented test cases were identified [4–6].

There was no common language to discuss defects in a broader context to improve medical device software (MDS) quality. Existing defect classification methods focus on defect attributes such as priority, severity and probability of recurrence [7]. Adopting generic defect classifications into safety critical software domains is not straightforward [8]. Since MDS is often safety critical, a domain specific defect classification scheme was needed [9]. To address this, the Association for the Advancement of Medical

A. Walker et al. (Eds.): EuroSPI 2019, CCIS 1060, pp. 1–13, 2019.
https://doi.org/10.1007/978-3-030-28005-5_47

Instrumentation (AAMI) has developed and published a standard, a new defect classification scheme called classification of defects in health software-SW91 in cooperation with the US Food and Drug Administration (FDA). It has been recognised as a consensus standard by the US FDA [10]. SW91 contains a set of defect categories for health care software. It includes defect categories from the requirements gathering to the maintenance phase of software development. This research has contributed to the development of SW91 via validating its defect coverage [9, 11].

Considering both the development of SW91 and the benefits of defect taxonomies in software testing, taxonomy based testing using SW91 was proposed for improving MDS quality [4–6, 12]. In taxonomy based testing, the requirements will be mapped into potential SW91 defect categories, and test cases will be written based on the requirements and the mapped defect categories. Test cases will be executed to verify whether the software complies with the relevant requirements and does not contain the mapped defect categories. An empirical retrospective study of taxonomy based testing was conducted using data from a MDS development organisation, company A. This study has helped to assess the applicability and the benefits of taxonomy based testing. The process followed in this study and benefits including defect reporting at the testing phase, defect minimisation and risk minimisation were explained in our previous publications [9, 11].

After conducting the retrospective study using data from company A, another MDS development organisation, company B from Ireland which provides products and services for connected health was contacted to implement taxonomy based testing. Company B did not agree to share their data due to concerns about data confidentiality. This necessitated the development of a taxonomy based testing framework to implement taxonomy based testing at company B while respecting their data confidentiality concerns. As a result of this meeting with company B, the alpha version of the taxonomy based testing (α-TBT) framework was written and it was sent to the product development manager in company B for his feedback.

Feedback on the α-TBT framework indicated that, while the approach was recognised to be beneficial, implementation of the α-TBT framework would require a lot of resources. Due to resource constraints, it was not possible to implement the framework at company B. This comment was taken into consideration and a beta version of the TBT (β-TBT) framework was written. The β-TBT framework enables the implementation of taxonomy based testing on a staged basis so that it can be adopted at any phase of MDS development such as implementation, testing or maintenance.

This paper details the β-TBT framework which enables implementation and investigation of taxonomy based testing at any MDS development organisation which follows the MDS development processes detailed in IEC 62304:2006+A1:2015. This paper is structured as follows: Sect. 2 explains the interconnection between taxonomy based testing using SW91 and the standard for MDS development processes, IEC 62304:2006+A1:2015. Section 3 explains the steps to implement taxonomy based testing. Section 4 explains the standard test processes and how taxonomy based testing fits into the standard test processes from the BS ISO/IEC/IEEE 29119-2 and the ISTQB. Section 5 explains the future work of this research and it details how this framework will be validated and how this framework will be tailored to testing techniques. Section 6 presents the summary and conclusion.

2 Taxonomy Based Testing Using SW91 and IEC 62304:2006 +A1:2015

In this research, we use SW91 as the defect taxonomy. In order to implement taxonomy based testing using SW91 at a MDS development organisation, it is essential to consider the MDS development processes. Successful MDS development processes are required to demonstrate compliance with a set of medical device standards and regulations [13]. IEC 62304:2006+A1:2015 is a standard for MDS development processes [14]. It describes the processes and the corresponding activities and tasks that are required to demonstrate compliance. SW91 follows the MDS lifecycle processes stated in IEC 62304:2006+A1:2015 such as software development planning, software requirements analysis, software architectural design and software detailed design. Table 1 shows how the MDS development lifecycle processes from IEC 62304:2006 +A1:2015 can be merged with SW91 parent level defect categories. By considering the linkages detailed in Table 1, it is possible to implement taxonomy based testing using SW91 in any MDS development organisation which is following the development processes detailed in 62304:2006+A1:2015.

Table 1. IEC 62304 and SW91 [15]

IEC 62304 MDS – software lifecycle processes	SW91, classification of defects in health software
Software development planning	Planning (1)
Software requirements analysis	Requirements (2)
Software architectural design	Architecture and Design (3)
Software detailed design	Architecture and Design (3)
Software unit implementation	Implementation (4)
Software integration and integration testing	Implementation (4), Test (5)
Software system testing	Test (5)
Software release for utilisation at a system level	Release defects (6)
Software maintenance process	Maintenance (7)

Annex C.4 of IEC 62304:2006+A1:2015 [14] explains a V-model for programmable electrical medical system requirements. This V model includes the MDS lifecycle processes detailed in IEC 62304:2006+A1:2015. Since SW91 has defect categories for each process of MDS development stated in IEC 62304:2006+A1:2015, it is possible to implement taxonomy based testing at any process of MDS development such as software requirement analysis or software implementation. Test cases can be created for each process of MDS development and they can be executed in each process's validation step. How taxonomy based testing fits with the V model development process from Annex C.4 of IEC 62304:2006+A1:2015 was explained in our previous papers [9, 11]. This section has discussed the interconnection between the MDS development processes from IEC 62304:2006+A1:2015 and taxonomy based testing using SW91 defect categories. The next section explains the steps to be followed in the implementation of taxonomy based testing. These steps were formulated based on the literature review [4, 5].

3 Steps for Taxonomy Based Testing

This section details the steps needed to implement taxonomy based testing at any MDS development organisation which follows the IEC 62304:2006+A1:2015 process. As explained in Sect. 1, these steps are part of the beta version of the taxonomy based testing framework which addresses data confidentiality of MDS development organisations. These steps enable the implementation of taxonomy based testing without the researcher's direct involvement with the organisation's data. It has the following six steps and the remainder of this section explains these six steps: [1] Review SW91 defect categories, [2] Requirements analysis and test phase selection, [3] Requirement mapping, [4] Test case writing, [5] Test execution and [6] Results analysis.

3.1 Step 1: Review SW91 Defect Categories

Since taxonomy based testing is a new testing approach and SW91is a new defect taxonomy, it is essential to get an understanding of SW91 and its defect categories. The selected MDS development organisation and all relevant participants who will participate in taxonomy based testing need to familiarise themselves with SW91 defect categories.

SW91 includes defect categories from the planning of a system to the maintenance of a system. SW91 contains multi-level defect categories such as parent level and child level. The parent level defect categories are as follows: Planning (1), Requirements (2), Architecture and Design (3), Implementation (4), Test (5), Release (6) and Maintenance (7) [7]. Each parent level defect category includes several child defect categories. Each defect category has a defect code with a unique number. The numbering uses a hierarchical system. Each parent level category is numbered and each child level category is represented by appending a full stop and its own number. Each defect category has annotations and some of the defect categories have an example as well. For example, Data accessibility (4.1.5.1) is one of the child level defect categories from Implementation (4) defects. It has the following annotation: "An object declared at an inner scope hides an object declared at an outer scope" [7].

Annex F of SW91 details all defect categories included in SW91. Participants can go through Annex F to get an overall picture of the defect categories and then they can go through the SW91 document to see defect category definitions. Further details on SW91 were explained in our previous paper [9].

3.2 Step 2: Requirements Analysis and Test Phase Selection

Taxonomy based testing can be implemented in three different ways. The organisation can select an appropriate way that is aligned with their business goals and their ongoing projects. The requirements need to be analysed to select an appropriate option to implement taxonomy based testing. The remainder of this section explains the following three options:

1. Implement taxonomy based testing for all requirements.
2. Implement taxonomy based testing for higher risk requirements.
3. Implement taxonomy based testing for selected requirements at appropriate test phases.

1. **Implement taxonomy based testing for all requirements**
   The organisation can implement taxonomy based testing for all relevant requirements. All requirements will be used to conduct taxonomy based testing. It will include all of the test phases such as unit testing, software integration testing, system testing, acceptance testing and maintenance level testing. The requirements will be mapped into potential SW91 defect categories which is explained in Sect. 3.3, Step 3: Requirements mapping.
2. **Implement taxonomy based testing for higher risk requirements**
   The organisation can select requirements with higher risk using their risk management process to conduct taxonomy based testing. The selected requirements will be mapped into potential SW91 defect categories which is explained in Sect. 3.3, Step 3: Requirements mapping and will go through the entire test phase such as unit testing, software integration testing, system testing, acceptance testing and maintenance level testing.
3. **Implement taxonomy based testing for selected requirements with appropriate test phases**
   The organisation can select suitable test phases by analysing the requirements. When stakeholders are analysing the requirements, they can identify and decide suitable test phases. The test phases include unit testing, software integration testing, system testing, acceptance testing or maintenance level testing. Table 2 shows sample requirements from company A and their appropriate test phases.

Table 2. Sample requirements and test phases

Sample requirements	Appropriate test phase
The application must be available for Apple IOS devices	Acceptance testing
The application must have a news feed that will appear on the applications home page	Software integration testing
The application must have up to date and accurate calculations and data	System testing
The application must show data outputs in a number of visual graphs	Unit testing

After selecting the appropriate test phases, the selected requirements will be mapped into potential defects. For example, if an organisation decided to implement taxonomy based testing on unit testing, they can select Implementation (4) defects. Table 3 shows the test phases and relevant SW91 defect categories.

Table 3. SW91 defect categories and tests phases

Test phase	SW91 defect categories
Acceptance testing	Planning (1), Requirements (2) Test (5), Release defects (6)
Software integration testing	Architecture and design (3), Test (5)
System testing	Architecture and design (3), Test (5)
Unit testing	Implementation (4), Test (5)
Maintenance level testing	Maintenance (7), Test (5)

3.3 Step 3: Requirements Mapping

The requirements mapping step is designed to map the selected requirements into potential SW91 defect categories. At this step, we must have a clear idea about the selected requirements, the test phases and SW91 defect categories. It is expected that participants have gone through the SW91 document at least once as explained in Sect. 3.1 before starting this step. When participants have the requirements, they can predict the potential defects which may occur in the future development of the requirements. Participants can get an idea of related keywords to those requirements. Keywords can be from the requirement itself or participants can use their experience to find related keywords. Once they identify the keywords, they will use those keywords to search the SW91 document for relevant defect categories. Requirements will be mapped to the identified defect categories. After this initial mapping, participants can go through Annex F of SW91 to get any missed potential defect categories for their requirements. Annex F of SW91 details the entire defect categories included in SW91. If participants cannot understand the defect categories by the name itself; they can go through the SW91 document for the defect category definitions.

One or many people can do this mapping. When many people are doing this mapping, the final version of the mapping can be created by comparing and merging the individual mappings into one mapping. The time taken to conduct the mapping can vary between individuals. It depends on individual expertise and their familiarity with SW91 defect categories. The time taken to conduct the mapping should be recorded by the organisation as it is essential to evaluate the efficiency of taxonomy based testing.

Table 4 shows a sample mapping of the potential SW91 defect categories for a requirement, *R1: Provide data validation, including the use of appropriate user interface (field) controls as well as back end data validation* [12]. From the requirement R1, the following keywords were identified: data, validation, controls, interface, fields and back end/database. These keywords were used to search the SW91 document for relevant defect categories. After the initial mapping, Annex F was searched for any missing defect categories and the mapping was concluded.

This mapping was done with the intention of implementing taxonomy based testing for all of the test phases such as unit testing, software integration testing, system testing, acceptance testing and maintenance level testing. Defect categories from different phases of the software development life cycle were included here.

Table 4. Requirements mapping example

Requirement and ID	Phases from SW91	Defect categories from SW91
R1: Provide data validation, including the use of appropriate user interface (field) controls as well as back end data validation	Requirements (2)	Requirement clarity (2.1.2)
	Architecture and design (3)	Transactional integrity (3.4.4)
		Devices and drivers (3.11.2)
		I/O timing or throughput (3.11.3)
		External data validity (3.11.4)
	Implementation (4)	Data definition (4.1)
		Data use and processing (4.2)
		Software interfaces (4.4)
		Interface parameter value (4.4.2.4)
		Naming, data definition, declarations (4.11.3)
		Control flow (4.3)
	Test (5)	Initialisation of test data (5.1.6)
	Release defects (7)	Database upgrades (7.5)

This requirement did not map to any defect categories from the Planning (1), Release Defects (6), or Maintenance (7) phases. This mapping can differ when selecting the defect categories for particular tests phases. If the stakeholders decide to implement taxonomy based testing at any particular test phase, they can map into the appropriate defect category from SW91 as shown in Table 3. For example, if they want to conduct unit testing using taxonomy based testing, they can map into the Implementation (4) defect categories. In this requirement mapping, more details such as use case diagrams or control flow diagrams can be used. The following stakeholders can be included in this step: business analyst, developers, testers and domain experts.

3.4 Step 4: Writing Test Cases

Test cases will be created based on the requirements and mapped defects after finishing the mapping of the requirements into SW91 defect categories. When quality assurance engineers are writing the test cases, they can use the mapped defect categories as a basis and it is expected that the test cases will cover the mapped defect categories. This type of test case writing will allow creating goal oriented test cases and it will help in increasing the test efficiency [4, 5].

3.5 Step 5: Test Execution

The test cases will be executed and failures will be identified and recorded. If a particular test case failed then the possible mapped defect categories can be the causes for that failed test case. Quality assurance engineers will be able to see the mapped defect categories for failed test cases. They can prepare a report with the possible defect categories which caused the failure in the MDS. This step does not require any more effort than normal test execution.

3.6 Step 6: Results Analysis

If the defect data and systems components will be retained by the organisation and not provided directly to the researcher, this is an essential step from a research point of view. The data from each step of taxonomy based testing can be collected through questionnaires in a focus group or at short interviews. This data will be used for research purposes only. If there are any publications, then approval will be sought from the organisation and only anonymised data will be published once approval has been obtained. This step will be used to evaluate the benefits and efficiency of taxonomy based testing. Annex E.2 of IEEE Std 1012-2016 - Standard for System, Software, and Hardware Verification and Validation explains the measures for evaluating defect density such as requirement defect density, design defect density, implementation defect density and test defect density [16]. These measures will be used to evaluate the efficiency of taxonomy based testing.

At this stage of the document, taxonomy based testing using SW91 and its steps have been explained. Taxonomy based testing can be implemented at any MDS development organisation which follows development processes detailed in IEC 62304:2006+A1:2015. The next section explains the relationship between taxonomy based testing and standard test processes from the ISTQB and the BS ISO/IEC/IEEE 29119-2.

4 Integration of Taxonomy Based Testing into the ISTQB and BS ISO/IEC/IEEE 29119-2 Test Processes

If an organisation wants to adopt taxonomy based testing, then it should be aligned with general software test processes. The literature review revealed two main test processes, the ISTQB [4, 17] and the ISO/IEC/IEEE 29119-2 [18]. This section will explain how taxonomy based testing is adaptable at an organisation which follows standard test processes such as the ISTQB and BS ISO/IEC/IEEE 29119-2 test processes.

The BS ISO/IEC/IEEE 29119-2: test process defines the following three-layer process and their sub processes [18]: organisational test process, test management process and dynamic test process. The organisational test process is used to develop and manage test specifications at the organisational level. These specifications can be applied to testing across the whole organisation. The test management process has the following three sub-processes: test planning, test monitoring and control and test completion. The dynamic test process has the following four sub processes: test design and implementation, test environment set-up and maintenance, test execution and test incident reporting. The ISTQB is not divided into different levels and the organisational test process is not discussed in it. However, implementation of taxonomy based testing is focused on the project level and it will not make any changes to the organisational test process in any organisation which follows the BS ISO/IEC/IEEE 29119-2: test process. The ISTQB has the following five processes: Test planning and control, test analysis and design, test implementation and execution, test evaluation and reporting, and test closure activities [17]. These five processes can be linked with the BS ISO/IEC/IEEE 29119-2: test processes except for the organisational test process. Table 5 shows the ISTQB test process and the equivalent stages from the BS ISO/IEC/IEEE 29119-2 test process.

Table 5. Test processes from the ISTQB and the BS ISO/IEC/IEEE 29119-2 test processes

ISTQB test process	BS ISO/IEC/IEEE 29119-2 test process
Planning and control	Test planning Test monitoring and control
Analysis and design	Test design and implementation
Implementation and execution	Test design and implementation Test execution Test environment set-up and maintenance Test incident reporting
Evaluating exit criteria and reporting	Test monitoring and control
Test closure activities	Test completion

Figure 1 details the test processes from the BS ISO/IEC/IEEE 29119-2 and the green coloured rectangles show the ISTQB test processes. It is clear that the ISTQB test process has equivalent stages in the BS ISO/IEC/IEEE 29119-2 test process. When it comes to the test environment setup and maintenance, the BS ISO/IEC/IEEE 29119-2 has a separate process and the ISTQB does not have a separate process for the test environment setup and maintenance, however the implementation and execution process from the ISTQB test process includes test environment set up and maintenance [17]. This section investigated the similarity between the ISTQB and the ISO/IEC/IEEE 29119-2 test processes. The next section outlines the integration of taxonomy based testing into standard test processes.

4.1 Integration of Taxonomy Based Testing Using SW91 into the Standard Test Processes

The main difference between the standard test processes and the taxonomy based test process is the requirements mapping. The standard test processes do not have a requirements mapping step. In taxonomy based testing, requirements will be mapped into their potential defect categories from SW91. Test cases will be created based on the requirements and the mapped defect categories.

Steps 1 and 2 of taxonomy based testing (review SW91 defect categories, requirements analysis and test phase selection) are integrated into the ISTQB test process 1, test planning and control. These first two steps of taxonomy based testing are integrated into test planning, test monitoring and control from the BS ISO/IEC/IEEE 29119-2:2013 test process. Steps 3 and 4 (requirement mapping, test case writing) of taxonomy based testing are integrated into the ISTQB test process 2, test analysis and design. This step is integrated into test design and implementation from the BS ISO/IEC/IEEE 29119-2:2013 test process. Step 5 (test execution) of taxonomy based testing is integrated into the ISTQB test process 3, test implementation and test execution. This step is integrated into test design and implementation, test execution, test environment set up and test incident reporting from the BS ISO/IEC/IEEE 29119-2:2013 test process. Step 6 (results analysis) of taxonomy based testing is integrated into the ISTQB test process 4, test evaluation and reporting. This step is integrated into test monitoring and control from the BS ISO/IEC/IEEE 29119-2:2013 test process. Taxonomy based testing does not have any specific test closure activities. It can include test closure activities as explained in the ISTQB test process or the BS ISO/IEC/IEEE 29119-2:2013 test process. Figure 2 details how the steps of taxonomy based testing can be integrated into the standard test processes from the ISTQB and the BS ISO/IEC/IEEE 29119-2:2013.

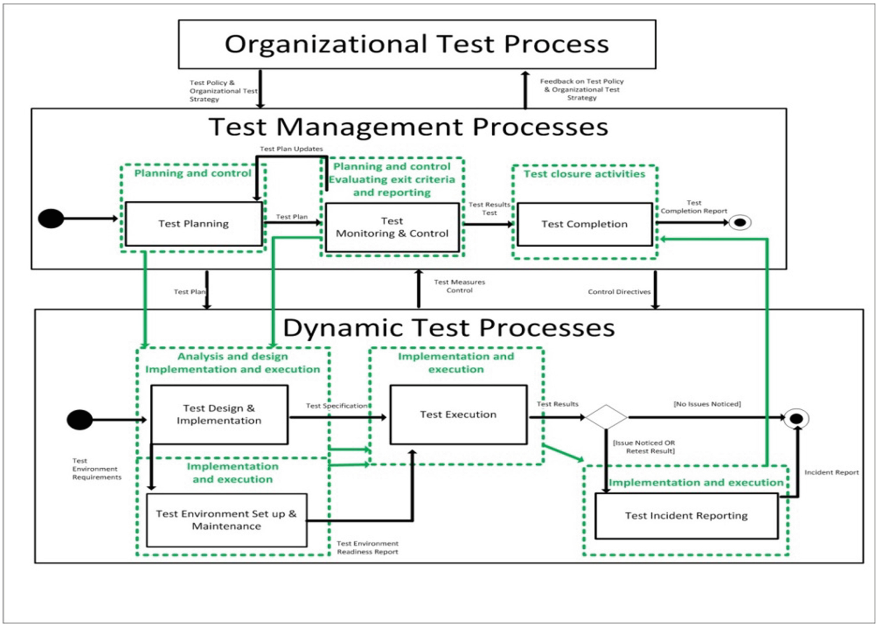

Fig. 1. Test processes from ISTQB and BS ISO/IEC/IEEE 29119-2 (Color figure online)

5 Future Work

This framework will be reviewed by two academic experts who are established researchers in the field of software testing and software testing using defect taxonomies. The framework will then be reviewed by experts from MDS development organisations, company B and company C. Feedback will be collected and it will be used to create the final version of the framework. Parallel to the expert reviews, this framework will be implemented at MDS development organisations C and D to assess the benefits and efficiency of taxonomy based testing. The efficiency will be calculated using the formulas explained in the Annex E.2 of IEEE Std 1012-2016 - Standard for System, Software, and Hardware Verification and Validation standard [16]. This research is currently investigating how taxonomy based testing using SW91 can be customised to the testing technique used by a MDS development organisation [19].

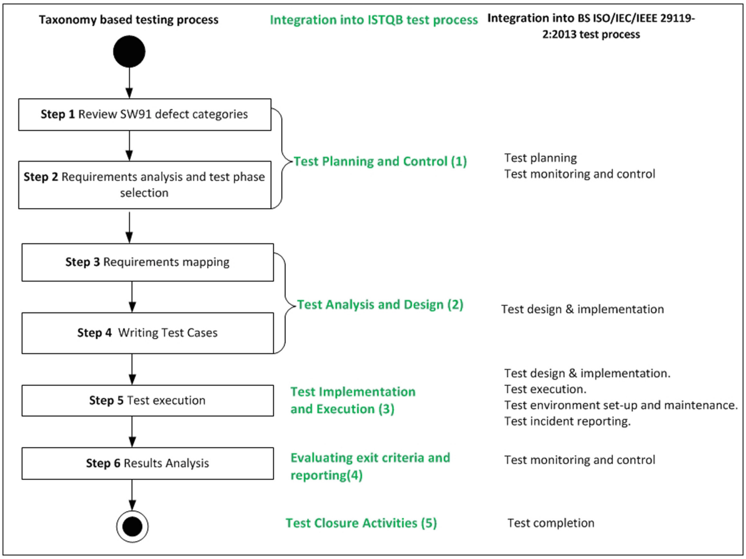

Fig. 2. Standard test processes and taxonomy based testing

6 Summary and Conclusion

Taxonomy based testing is a new testing approach. It will use a new standard, SW91, Classification of Defects in Health Software. A retrospective study of taxonomy based testing at company A revealed the following benefits: risk minimisation, defect minimisation and root cause analysis [9, 11]. MDS organisations can achieve the same benefits by implementing taxonomy based testing. Taxonomy based testing will allow MDS organisations to learn about defects across different projects and will improve the software process [20]. A framework is developed to implement taxonomy based testing at any MDS organisation which follows the processes detailed in 62304:2006

+A1:2015. This framework enables the implementation of taxonomy based testing without the researcher's direct involvement with the organisation's data. The framework has six steps which can be aligned with the ISTQB and the BS ISO/IEC/IEEE 29119-2 test processes. In our future work, MDS development organisations, companies C and D have agreed to implement taxonomy based testing using the framework. Future work will also investigate how taxonomy based testing using SW91 can be customised to the testing technique used by a MDS development organisation.

Acknowledgements. This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero—the Irish Software Research Centre (www.lero.ie).

References

1. ISTQB: Standard glossary of terms used in Software Testing (2017). http://glossary.istqb.org/search/defecttaxonomy
2. Freimut, B.: Developing and Using Defect Classification Schemes (2001)
3. Vallespir, D., Grazioli, F., Herbert, J.: A framework to evaluate defect taxonomies. In: XV Argentine Congress of Computer Science (2009)
4. Felderer, M., Beer, A.: Using defect taxonomies to improve the maturity of the system test process: results from an industrial case study. In: Winkler, D., Biffl, S., Bergsmann, J. (eds.) SWQD 2013. LNBIP, vol. 133, pp. 125–146. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35702-2_9
5. Felderer, M., Beer, A.: Using defect taxonomies for requirements validation in industrial projects. In: RE (2013)
6. Vijayaraghavan, G., Kaner, C.: Bug taxonomies: use them to generate better tests. In: STAR EAST 2003, pp. 1–40 (2003)
7. AAMI: American National Standard ANSI/AAMI SW91:2018 (2018)
8. Silva, N., Vieira, M.: Experience report: orthogonal classification of safety critical issues. In: Proceedings of the International Symposium on Software Reliability Engineering, ISSRE (2014)
9. Rajaram, H.K., Loane, J., MacMahon, S.T., McCaffery, F.: Taxonomy-based testing and validation of a new defect classification for health software. J. Softw. Evol. Process **31**(1), 1–13 (2018)
10. AAMI: Consensus Standards (2019). http://www.aami.org/newsviews/newsdetail.aspx?ItemNumber=7367. Accessed 19 Feb 2019
11. Rajaram, H.K., Loane, J., MacMahon, S.T., McCaffery, F.: Benefits of defect taxonomies and validation of a new defect classification for health software. In: EuroAsiaSPI^2, p. 15 (2017)
12. Black, R.: Advanced Software Testing - Vol. 1, 2nd Edition, vol. 1. Rocky Nook Inc., Santa Barbara (2008)
13. Rust, P., Flood, D., McCaffery, F.: Creation of an IEC 62304 compliant software development plan. J. Softw. Evol. Process **28**(11), 1005–1010 (2016)
14. IEC: Medical device software—Software life-cycle processes. BS EN 62304:2006 +A1:2015, vol. 3, no. November 2008. p. 88p (2015)
15. Simone, L.K.: AAMI SW91 CDV2 Classification of Defects in Health Software - Overview Public (2017)

16. IEEE: IEEE Std 1012-2016 - Standard for System, Software, and Hardware Verification and Validation (2017)
17. Graham, D., van Veenendaal, E., Evans, I., Black, R.: Foundations of Software Testing; ISTQB Certification (2006) AQ3
18. ISO/IEC/IEEE: BSI Standards Publication Software and systems engineering—Software testing Part 2: Test processes (2013)
19. ISO/IEC/IEEE: BSI Standards Publication Software and systems engineering—Software testing Part 4: Test techniques (2015)
20. Pries-Heje, J., Johansen, J., Messnarz, R.: SPI Manifesto. Version A, vol. 1, p. 2010 (2010)

Author Proof

Author Query Form

Book ID :	**488218_1_En**
Chapter No :	**47**

Please ensure you fill out your response to the queries raised below and return this form along with your corrections.

Dear Author,
During the process of typesetting your chapter, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query Refs.	Details Required	Author's Response
AQ1	This is to inform you that corresponding author and email address have been identified as per the information available in the Copyright form.	
AQ2	Per Springer style, both city and country names must be present in the affiliations. Accordingly, we have inserted the city name "Dundalk" in affiliation. Please check and confirm if the inserted city name is correct. If not, please provide us with the correct city name.	
AQ3	Kindly provide publisher name and location for Ref. [17].	Publisher: Cengage Learning EMEA Location: London