

Selection and Application of Software Testing Techniques to Specific Conditions of Software Projects

Othmar Othmar Mwambe
Lecturer, LIS Department

Tumaini University Makumira Dar es salaam
college, Dar es salaam, Tanzania

Oleksandr Lutsaievskiy (PhD)
Project Management Consultant
EPAM Systems, Ukraine

ABSTRACT

Software is becoming a key ingredient in most of the modern systems and devices that support important business processes in our modern society. It is an essential component of many embedded applications that control various sensitive applications such as air traffic control systems, rockets, automated banking systems (ATM), security systems (radar) and so many other applications. The failure of those systems can result into severe damage [7]. It is obvious that software testing technologies are essential for software testers. Even though there are several software testing methodologies and techniques to support the quality of software but to find relevant parameters for their applicability conditions remains an open question. This paper attempts to shed light on decision criteria, by describing various software testing techniques and their distinctions. The finding shows that these methodologies and techniques do not have direct influence on the quality of software but the choices we make on selecting relevant methodology/or technique for the application.

Keywords

Software testing, software quality and standards, agile methodologies, software development.

1. INTRODUCTION

In recent years software development has become a great field of interest due to its role in the technological innovation. Software development industries are very good at adopting the latest technology trends and development practices so as to cope with rapid growing technology. All large organizations are now integrating their systems and as a result the need of real-time systems and regular software improvement is growing dramatically. The main challenge remains to software development industries is to find a proper software development methodology and testing techniques that can sustain the existing situation in market. The traditional software development methodologies (e.g. V-model, figure 3) seemed to slow down software development processes due to existing time constraint situation. This study attempts to tackle the existing problem in the software development market (“Selection of relevant software development methodology and testing technique”) and propose some solutions. We believe that choice of software development methodology has significant implications for software testing [20].

2. Background study, related work, literature review and state of the Art

2.1 The Importance of testing, Software Quality and Standards

The first Ariane 5 Flight 501 exploded 40 seconds after liftoff on June 4th 1996. The launch failure was due to unhandled floating point conversion exception in an internal guidance system function. The lack of system tests (that would have identified the problem earlier) resulted into great loss [7].

In this section we are going to discuss about important subjects (figure 1) that can eventually result in to the improvement of software quality and standards.

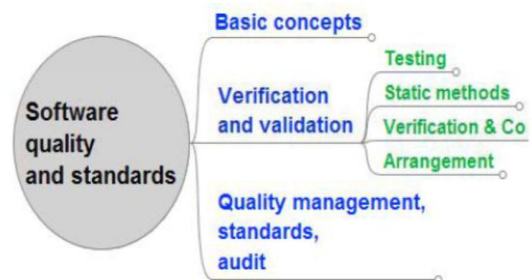


Figure 1: Software Quality and standards

Error - A discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition / A human action that produces an incorrect result [ISO/IEC 2382 / ISTQB]

Fault - an abnormal condition that may cause a reduction in or loss of, the capability of a functional unit to perform a required function.

Defect: A flaw in a component or system that can cause the component or system to fail to perform its required function [ISO/IEC 2382 / ISTQB].

Failure - The termination of the ability of a functional unit to perform a required function / Deviation of the component or system from its expected delivery, service or result [ISO/IEC 2382 / ISTQB]

Verification-is an activity trying to demonstrate that the result of the next stage of development complies with specification given in the previous stage [8]. Through verification we are able to make the software product to behave the way we want it to.

Validation is an activity trying to demonstrate that what was required has been achieved.

Software Quality-is a degree to which a system, component or process meets specified requirements (customer/or user needs and expectations) [6].

Software testing is an essential tool that plays a great role in the improvement software development to a degree into which a system, component or process meets specified requirements (customer/user needs and expectations) based on certain standards [6]. Therefore a quality product is the result of well performed tests that fall into particular standards.

Software testing is crucial because it helps us to identify risks that propagate in the system throughout software/system development cycle. When the risk has been identified then a proper action (*Treat, Transfer, Tolerate, and Terminate*) can be taken into consideration with respect to the assigned priority (figure 2)

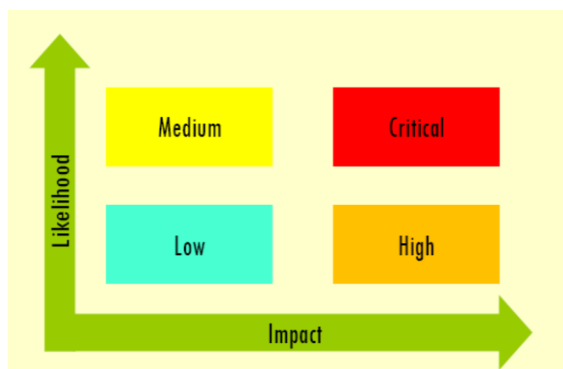


Figure 2: Risk possible matrix-Priority [8]

2.2 Testing psychology and Design

In most cases, many testers/or programmers rely on following facts [8] during test performance:

- A good test has high probability of detecting errors
- If errors were found, then the test was successful
- Testing does not prove that there are no errors in the program but it can only show that there are errors in the program

However in any tests design, the following problems[Error! Reference source not found.] must be solved:

- How to choose inputs?
- How to value the outputs?
- When to finish testing?
- Who are the testers? and
- When, whether and how to perform regression testing?

3. Software testing techniques

There are two basic classes of traditional software development methods (based on V-model (figure 3) and water fall model), functional testing (Black Box) and White Box testing. Obviously, functional testing derives tests from external descriptions of the software such as specifications, requirements, and design while White Box testing derives tests from source code internals of the software such as branches, statements and conditions.

Research problem

Even though there are several software testing techniques to support software test cases but to find relevant parameters for their applicability conditions remains an open question. In this study we are going to describe various software testing techniques and their distinctions, including modern family of Agile software development methods.

Unit testing- is the testing of individual hardware or software units or groups of related units [6]; it belongs to White Box class whereby the tester examines Low-Level Design of the code, verifying if the code does exactly what it is made for. It assesses software with respect to implementation.

Integration testing- is testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them [6]; it assesses software with respect to subsystem design. It belongs to both Black Box testing and White Box testing whereby the test cases based on High Level Design and Low-Level Design verifying if the units are properly working together particularly when they are integrated into a very large code.

System testing - belongs to Black Box class, whereby the test is conducted with full system implementation, including non-functional properties of the system. It is conducted on a complete, integrated system to evaluate the system compliance with its specified requirements [6]; it also responsible for compatibility issues when embedding a program in different environments (Operating systems/applications).

Functional testing - verifies if functionally the system works with respect to requirements specification. It belongs to Black Box class under high level design.

Regression testing- is selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements [6]; it belongs to both classes Black Box testing and White Box testing under high level design. It belongs to both classes due to fact that regression testing runs throughout software development cycle (figure 3) whereby a tester uses it to examine whether the new code works properly and has not caused unintended side effects in previously-working functionality.

Beta testing- is commonly used through trial version software packages whereby a software/system development organization gives out either partial or full version of its software products for free. Users (or testers) install and use the software package but in turn they give back the feedback report to the development organization. There are several advantages of using beta testing including wide population search for errors in variety environments and low cost but it also involves serious disadvantages such as lack of quality error report because users sometimes give back report which are not detailed, and much effort is needed to examine all the error reports as the reports come from multiple users.

Acceptance testing- is formal testing conducted to determine whether or not a system satisfies its acceptance criteria (the criteria the system must satisfy to be accepted by a customer) and to enable the customer to determine whether or not to accept the system [6]; It belongs to a Black Box testing class based on requirements specification. These tests are customer dependent in such a way that customers are involved in some pre specification which later given to the testing team to run before the delivery of the product.

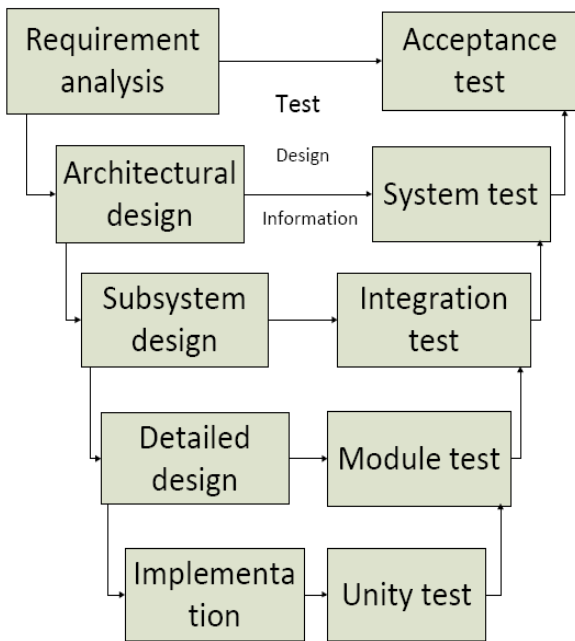


Figure 3: Software development activities and testing levels—the “V Model”

3.1 Functional Testing

Functional testing is also known as Black Box testing and behavioral testing. It is a software testing methods that examines functionality of an application based on its functional requirements and not internal structure (figure 4).

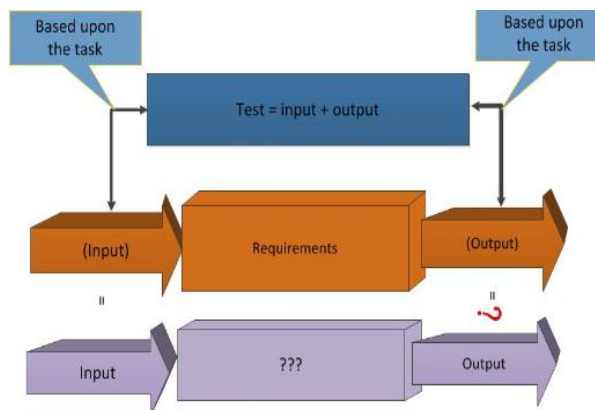


Figure 4: Functional testing

Black Box method is applicable to almost all levels of software testing process (Table 1) .It considers only input and output of the software without taking into account its internal structure. It is tackling errors base on the following categories:

- I. Incorrect or missing functions
- II. Interface errors
- III. Errors in data structures or external database access
- IV. Behavior or performance errors and
- V. Initialization and termination errors.

Black Box testing techniques

Equivalence Partitioning

This technique based on division of input values. It divides the input values into two classes, valid and invalid partition then it selects some values from each partition to be representative values of the rest and uses the representative values as test data (figure 5).And finally decision can be made based on test data.

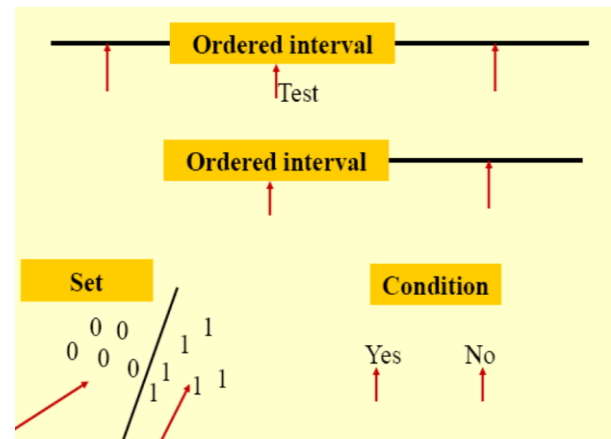


Figure 5: Equivalence Classes [Error! Reference source not found.]

Boundary Value Analysis (BVA)

This technique based on determination of boundaries for the input values. It selects test data from the input values that lie at the boundaries and just input and outside the boundaries (figure 6).The boundary value is defines as “data value that corresponds to a minimum or maximum input, internal, or output value specified for a system or component” [6].

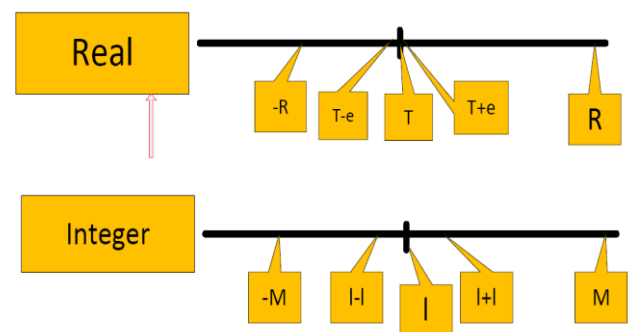


Figure 6: Boundary Value Analysis

Let us create BVA test case with respect to figure 5 above based on integer values.

Assuming that our input conditions have range from $-M$ to M . We have set a boundary I at “ I ”. Our test cases will be:

- Immediately below the boundary($I-1$)
- At the boundary(I)
- Immediately above the boundary($I+1$)

Decision table testing

This technique based on records resulted from some business rules with some conditions which are implemented in a certain program so as to be tested (Table 2). Actions are events that should start upon the set of input conditions. Each rule can result into a test case.

Usage of equivalence classes/boundary values

- Distinguish the input and output data equivalence classes, if necessary also input and output combinations
- Select data in each equivalence class and in boundary values
- Combine those data into tests
- Detect the corresponding outputs
- Identify test and compose a testing plan
- Test, estimate

Advantages Black Box testing

- Efficient when it is used on large systems
- Is not necessarily for a tester to be technical person
- Test cases can be designed immediately after the functional specifications are done.
- Usually tests base on user's perspective

3.2 Agile Software Testing Approaches

Various Agile software methodologies (such as XP, SCRUM, DSDM, FDD and Crystal) appeared in the early 1990's as a response to the issues that traditional software development methodologies (v-model and waterfall, figure 3) had with cost of changing the requirements, long time to market and high risk of failing the expectations of end-users in a changing business environment. Then in 2001, at a summit of CTOs, academics and the founders of these methodologies came to consensus and developed the "Agile Manifesto" that summarized their views on a modern way of building software (figure 7).



Figure 7: The agile software development manifesto [10]

Agile methodologies support step-by-step increase in feature addition (incremental development) and feature upgrade (iterative development) whereby the work is carried effectively within a development team(s) closely working with business representative. The software development is carried out by time-boxed iterations with length of 1-4 weeks, during which the development team code, test and deploy the functionality with the highest priority. At the end of each

iteration the team ships the part of working software product. Since working increments of the software are released very often in agile software development there is also a need to test often.

Agile Testing - is an approach in software testing based on agile manifesto principles and it is taking into consideration client requirements as early as possible. The testing is performed from the beginning of the project and continually testing throughout software project lifecycle. [Figure 8] "Testing is not a phase in Agile teams, testing is a way of life. Agile teams test continuously. It is the only way to ensure that the features implemented during a given iteration/or sprint are actually done" (Elisabeth Hendrickson [16]).

One of the practices in Agile testing is driving the development by writing the automated test cases (unit tests) that define a desired implementation improvement or new functionality before writing the functionality itself [22]. After the unit tests are written, the function is coded to pass the written tests. This approach, called test-driven development (TDD), significantly reduces the number of defects found in production code, as any changes in the code are tested immediately versus unit tests.

Agile testing also involves automation of regression testing to minimize time and human errors that can be resulted from manual work.

Therefore, Agile software automation testing formation framework based on the following criteria [11]

- Test is repetitive
- The test evaluates high risk conditions
- The test is costly or not impossible to perform manually
- The test requires multiple data values to perform the same action
- The test is a baseline test run on several different conditions.

Agile software development methodologies – two widely used agile software development methodologies are extreme programming (XP) and Scrum. They are slightly different but sharing the same iterative approach (figure 8).

Extreme programming – begins with the release of a planning stage then other iterations are following, each iteration is ending up with user acceptance testing. It based on development processes and not managerial aspect. It follows some principles/or concepts such as **continuous integration** – whereby teams must integrate all the changes they introduced in the program daily, **pair programming** -tells that two coders can satisfy user stories more efficiently when they are working together than working alone, **project velocity** which is the unit for work done in the project, **User stories**-these are statements used to present the problem to be solved.

Scrum-includes both aspects, managerial and development processes. Instead of creating a complete plan for developing software, a team spends minimal time creating a sketch plan to understand timelines and scope of the work. The team has daily status meetings (called Scrums) where each team member describes the task to be done on that day and progressive report from the previous day. Then first part of working software is released as soon as possible so as for customer to review and give feedback on improvement of the software. It includes some concepts such as **product backlog** – is a prioritized list of high-level requirements, **Scrum Master** – a team member, responsible for using the development

process effectively, **burndown chart** – a chart that shows progressive updates on completed functionality, **sprint backlog** - is a partial list of backlog items assigned to an iteration (sprint).

Advantages of Agile software methodologies

Reduced time to market and costs – due to fact that this approach focuses more on the working product rather than support documentation and complex process, therefore it includes less documentation and **very transparent** since end-users and stakeholders get opportunity to provide regular feedback often. Its iterative feature supports quick accomplishment of the projects.

Flexibility and low cost of change – Agile software development features accept any request for change to be implemented (e.g. regular feedback from the end user) for software project improvement at any stage without budget constraint.

Higher quality – Agile software development methodology produces high quality software due to test driven approach, high alignment of development team to business needs, close communication between members of development team and

business, and fast risk mitigation on a daily status meetings [1].

4. Conclusion

Choice of software development methodology has significant implications for software testing [20]; therefore the study attempts to assess the influence of software development methodologies, software testing techniques, and relevant parameters for their applicability conditions so as to help software developers and testers making a proper choice. Our findings show that due to dramatic growth of real-time systems with high demand of regular updates, therefore if the Agile software methodologies are well implemented in the software development project with the support of some testing techniques (such as regression testing, table 1) from traditional software development methods (e.g. V-model, figure 3) can make a perfect match in case of large systems development. Finally, the study shows that the software development methodology doesn't real matter in the software development but its relativity with statement of problem.

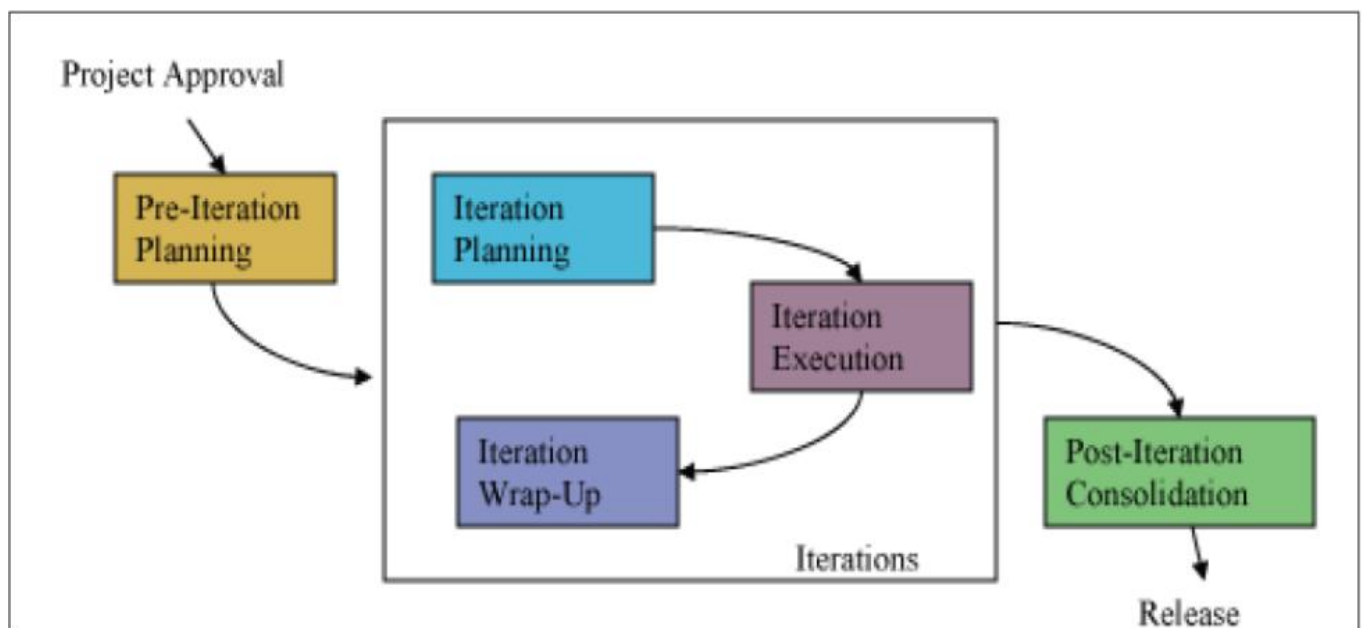


Figure 8: Agile development process features initial planning stage, rapid repeats of the iteration stage, and some form of consolidation before release [9]

Table 1: Software techniques levels

Testing technique	Class	Specification	Coverage	Tester
Unit	White Box	Low-Level Design	Small unit of code	Code writer (Programmer)
Integration	White Box Black Box	High level and Low-Level Design	Large code with many classes	Code writer (Programmer)

Functional	Black Box	High Level Design	Whole Product	External tester
System	Black Box	Requirements Analysis	Whole Product	External tester
Regression	Black Box White Box	High Level Design	Throughout software development cycle	Programmer/or External tester
Acceptance	Black Box	Requirements Analysis	Whole Product	User/or Client
Beta	Black Box	Situational	Whole Product	User/or Client

Table 2: Decision Table

		Rules							
Conditions	Printer does not print	Y	Y	Y	Y	N	N	N	N
	A red light is flashing	Y	Y	N	N	Y	Y	N	N
	Printer is unrecognized	Y	N	Y	N	Y	N	Y	N
Actions	Check the power cable			X					
	Check printer-computer cable	X		X					
	Ensure printer software is installed	X		X		X		X	
	Check/replace ink	X	X			X	X		
	Check for paper jam		X		X				

5. REFERENCES

- [1] Myers, G., "The Art of Software Testing". Hoboken, New York: John Wiley & Sons, 2004.
- [2] Beizer, B., "Software Testing Techniques", 2nd. Edition, Van Nostrand Reinhold, USA 1990.
- [3] Srinivasan D., Gopaldaswamy R., "Software Testing: Principles and Practices", 1st Edition, Pearson Education, New Delhi India, 2006.
- [4] Sommerville, "Software Engineering", 7th Edition, Pearson Education, 2005.
- [5] Bertolino, A., "Guide to the Knowledge Area of Software Testing". Software Engineering Body of Knowledge. IEEE Computer Society, February 2004, <http://www.swebok.org>.
- [6] IEEE, "IEEE Standard 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology," 1990.
- [7] Prof. J. L. Lions. Ariane 5 flight 501 failure: Report by the inquiry board, July 1996, <http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html>
- [8] Prof. JaakTepandi, "Software Quality and standards 2011 IDY0204 v12.09.2011", <http://www.ttu.ee/users/tepani/wi-tk-k.html>.
- [9] Serena, "An Introduction to Agile Software Development", June 2007
- [10] uTest, White Paper, "Agile software Testing-Ten Tips for understanding Agile Development", 2012
- [11] Hashir R, Madhu B K and Loksha V, "A Sophisticated Study on Best Practices of Agile software Testing", ISSN 2249 -071X ,2012
- [12] Erickson, J., Lyytinen K., and Siau, K. "Agile modeling, agile software development, and extreme
- [13] Programming: The state of research," Journal of Database Management"(16:4), 2005, pp. 88-100.
- [14] Crispin, L. and Gregory, J. "Agile testing: A practical guide for testers and agile teams", Addison Wesley, Boston, Massachusetts, 2009.
- [15] Madhu B K, Megha J and Loksha V: A Study on Agile Software Testing – Emergence and Techniques African Journal of Mathematics and Computer Science Research Vol. 3(9), November 2010
- [16] Elisabeth Hendrickson.: 2005: Agile Testing
- [17] L. Rising and N. S. Janoff, The Scrum software development process for small teams, IEEE Software, Issue 17, pp. 26-32 , 2000

- [18] L. Williams and A. Cockburn, "Agile Software Development: It's about Feedback and Change," *IEEE Computer*, June 2003, pp. 39-43
- [19] Laurie Williams, "Testing Overview and Black-Box Testing Techniques", 2006
- [20] Henderson-Sellers, B. and Serour, M. K. "Creating a dual-agility method: The value of method engineering," *Journal of Database Management* (16:4), 2005, pp. 1-23.
- [21] Amman and Offutt, "Introduction to software testing", chapter 1, pp.6-7
- [22] Beck, K. *Test-Driven Development by Example*, Addison Wesley - Vaseem, 2003