Situation Based Framework for Testing Methods and Metrics

Fiza Siyal, Rafia Naz Memon and Irfana Memon

Abstract - **There are many methods to test the software; the problem is to select an appropriate method from wide range of testing methods as well as to select measurement methods for assessing the testing effort. The purpose of this work is to create a situational method from method base of testing methods and testing metrics. Situational method engineering focuses on construction of new advanced development methods from parts of existing methods, called method fragments. Our proposed framework is based on testing methods and metrics to adapt the testing phase according to situation. Situations are created out of project characteristics and testing levels. We have collected all the testing methods / techniques and test metrics into SME repository, this repository will help developers for selecting a suitable method(s) by taking into account the project characteristics, and that selected method is called the situational method. The proposed framework can be automated to facilitate the new (novice) software developer for selecting a suitable method from variety of testing methods and testing metrics.**

Index Terms- SME

I. INTRODUCTION

Software testing is the process of operating a system or component under specified conditions, observing or recording the results and making an evaluation of some aspect of the system or component [1]. Software testing is the more difficult because of the vast array of programming languages, operating systems and hardware platforms that have evolved in the intervening decades. If we look in 1970's, there were very less number of computer users, but now it has become difficult to survive without a computer, so, not only do computers exist on your desk but software is present in almost every device we use, which raises the value of testing it [1]. There are many methods to test the software, but mainly there are two ways to test:

- Black box testing, and
- White box testing.

Further these methods encompass a wide range of techniques for testing a program /product.

There is a need to use suitable method or methods for program / product testing and on the same time selecting specific metric from the various test metrics to measure the efficiency and effectiveness of test which is performed at a particular testing level, so that programmer or tester can assess the quality of work done in a timely manner and cost effective way.

There is a variety of testing methods available and the problem is to select an appropriate method from wide range of testing methods and also measurement methods for assessing the testing phase. Situational method engineering is an innovative, flexible and dynamic way, where the different methods are combined to construct a situational method, which is then used by the development team to produce quality product. Most research in this area is based on assembly of software development methods [2, 3][4] to create a situational method where the focus is on analysis phase, so by keeping that idea in mind, the purpose of this work is to create a situational method from method base of testing methods and testing metrics.

Rest of the paper is organized as follows: Section 2 presents an idea of situational method engineering, section 3 presents the proposed framework based on testing methods and metrics, and section 4 summarizes and concludes the paper.

II. SITUATIONAL METHOD ENGINEERING

Method engineering (ME) and situational method engineering (SME) focuses on formalizing the use of methods for systems development. Method engineering discipline provides a way to design, construct and adapt methods for software development [3]. ME incorporates every aspect to create a development method for a specific situation [3]. This approach enables a developer to feel an owner of a methodology that is exactly adjusted according to individual requirements [3]. SME uses derived methods created from smaller pieces called fragments or chunks are stored in a method base [4-9], the structural aspects of which are discussed by e.g. [10]. In order to construct a situational method, project characteristics are focused, these characteristics assist a developer to decide about which fragments are (or are not) suitable to current product requirements [3]. Previous research on SME focuses on construction of new advanced development methods from parts of existing methods, called method fragments. Research reveals that how to model the existing methods and how to assemble method fragments into new project-specific methods, called situational methods.

Fiza Siyal, . Rafia Naz Memon, Department of Information Technology, Quaid e Awam University of Engineering Science and technology, Pakistan Irfana Memon, Department of Computer Systems Engineering, Quaid-e-Awam University of Engineering Science and Technology, Pakistan. Email: engr_fizza2001@quest.edu.pk

III. PROPOSED FRAMEWORK

Our proposed framework is based on repository of testing methods and metrics that is used to adapt the testing phase according to situation. Situations are created out of project characteristics and testing levels. A situational method is then constructed by selecting a method or methods from SME repository. Instead of taking or selecting random methods or techniques for testing a product it is important to choose a suitable method or adapt the testing phase by selecting methods from repository of methods. Situational method engineering provide a way of selecting a desired method from method pool that best fits the needs of the project, the same idea we have applied on just testing methods because it is testing that ensures whether quality exists in product or not. The proposed framework is presented in fig. 1. The components of proposed work are explained below.

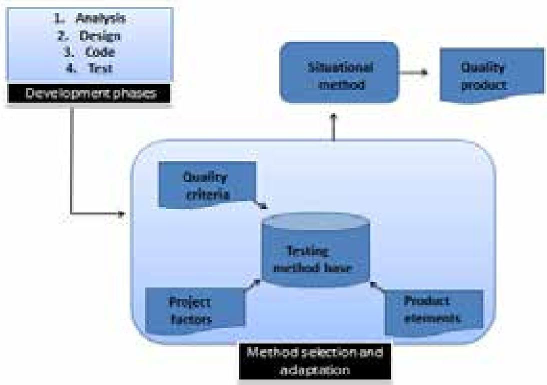

Fig. 1. Proposed Framework

A. Development phases

Software development can be divided into 4 phases, phase 1 is analysis which performs requirements collection, phase 2 is design that translates requirements into graphical representations or software architecture, phase 3 is implementation where the work from previous phases is converted into code, and final phase is testing that may be performed in parallel to code phase. So in our proposed framework development phases include analysis, design, implementation and testing.

B. Testing phase and levels

Testing phase in software development has broad aspects, where the testing is broken into four levels as: unit, integration, system, and acceptance. In normal sequence of testing, unit test is performed first, then integration testing is done, after these levels system test and acceptance test are conducted respectively. There are various types of testing at each test level, a particular type or types at a particular level may be selected by taking into account project characteristics and objectives of test [11], for example at unit test level only white box techniques are used, but it is important to know which type from white box techniques is required to perform a unit test.

C. Testing Method Base

Method base is a collection of methods used for software development. In the same manner method base of our proposed framework is a collection of various testing methods, it includes two main entities:

1) Testing Methods Entity

Testing methods entity includes a wide range of white box and blackbox testing techniques such as: control flow testing, data flow testing, use case testing, equivalence class partitioning, boundary value analysis, decision table testing etc. [12].

2) Test Metrics Entity

This entity presents a set of test metrics. In order to measure the performance of a particular test at a particular test level these metrics are used. Evaluation of testing may be done as: test coverage, test effort percentage, time to fix a defect etc., so there are many formulas (metrics) used for assessment are part of repository [13]. Metric is a unit used to describe or measure an attribute of something. A testing metric uses various units to measure different aspects of testing, for example LOC (lines of code) is a very simple metric for measuring the size attribute of software and in order to measure defect discovery rate attribute the metric used is defects found per thousand lines of source code. A metrics may be used to measure the process or a product. A process metrics find the effort or effectiveness of a particular test, whereas a product metrics measure some attribute of a product. For example, you may have a set of metrics that you use to evaluate the output of your testing team, one such metric may be the number of defects found. On the other hand some metrics may measure the efficiency of test cases written or the number of tests executed in a given period of time [14].
Some test metrics can be categorized as:

- Metrics for assessing the testability of the software
- Metrics for evaluating test cases
- Metrics for calculating test costs
- Metrics for measuring test coverage
- Metrics for assessing test effectiveness

The intension behind collecting various metrics is to give awareness to programmers so that they get motivated to participate in various measurement activities.

D. Method selection

This part of framework involves criteria for selection of a particular method from repository of methods. Project factors, Product elements and Quality criteria are the factors that affect the selection of a particular method for testing as well as test metrics to measure the test effectiveness.

1) Project factors

Project factors comprises of basic information related with project that includes: cutomers, information, team, equipment and tools, schedules and test items etc. Customers are the persons whose opinion about the system matters the most and the one who can get benefit or suffers from the work done. To a tester it is important to know about the product and project characteristics. Product information such as data, functions and behavior makes the task of testing easy. Team includes the persons who support the activities of testing; they may be writers, users or programmers. Equipment and tools covers the technical aspects of testing, where different hardware, software or documents including matrices and checklists are needed to accomplish the task of testing. Schedule information consists of various durations like, the total estimated time for testing, time taken to develop a software increment and time taken to perform a review on test related documentation. Test item is the product that is to be tested or it can be any component of the product. In order to test a particular item it is necessary to confirm the availability, volatility and testability characteristics of the item.

2) Product elements

A software product is much more than a code. It consists of many independent elements; it involves a purpose, platform and user. The product elements include Structures, Functions, Data, platforms and operations. Structures comprise of executable code and non-executable files, hardware that is integral part of product, and interfaces that establishes the link among various subsystems. Functions can be defined as anything that the software product does. Various things are part of functions such as user interfaces, system interfaces, error handling, and multimedia and log files. Data is a thing that is manipulated by the software product. It can be input data, output data, preset data (e.g the information stored in databases), persistent data shows the status of the product such as “options settings, view modes, contents of documents etc”. A platform is a thing on which product relies, it consists of external hardware and external software. External hardware and software are the components and configurations, required in order for a product to work; it includes CPU, memory, keyboards, and peripheral boards, operating systems, drivers etc. operations specify that how a product will be used, in physical environment (light, noise and distractions) by users.

3) *Quality criteria*

This is further divided into two categories as:

- *Operational criteria:* includes some attributes that determine whether product has some quality or not? These attributes are capability, reliability, usability, performance, scalability, install ability, compatibility, maintainability, portability etc.
- *Development criteria*: specify some standards that must be present in a product if it claims to have high quality, such as: coding standards, regulatory standards and industry standards.

IV. CONCLUSION AND FUTURE WORK

Software testing has great significance, and test techniques too, because the aim is to improve the quality of product and increase the customer satisfaction. Measurements not only improve the performance, but also give an insight into improvement efforts to development team. The intent behind collecting various metrics should be shared with development team so that they get motivated to participate in various measurement activities. By Testing metrics tester can evaluate the maturity and readiness for release of product. We have collected all the testing methods / techniques and test metrics into SME repository, this repository will help developers for selecting a suitable method(s) by taking into account the quality criteria, product elements and project environment.

The proposed framework can be automated to facilitate the new (novice) software developers with variety of testing methods and testing metrics. This will be a complete guideline especially for those developers and testers who have less experience of development and testing. Testers and developers can easily select an appropriate method from the SME repository by just giving the required information.

REFERENCES

[1] G. J. Myers, C. Sandler, and T. Badgett, *The art of software testing*: John Wiley & Sons, 2011.

[2] S. Brinkkemper, "Method engineering: engineering of information systems development methods and tools," Information and software technology, vol. 38, pp. 275-280, 1996.

[3] B. Henderson-Sellers and J. Ralyté, "Situational Method Engineering: State-of-the-Art Review," J. UCS, vol. 16, pp. 424-478, 2010.

[4] A. F. Harmsen, J. Brinkkemper, and J. H. Oei, Situational method engineering for information system project approaches: University of Twente, Department of Computer Science, 1994.

[5] M. Saeki, "Embedding metrics into information systems development methods: An application of method engineering technique," in Advanced information systems engineering, 2003, pp. 374-389.

[6] C. Rolland, V. Plihon, and J. Ralyté, "Specifying the reuse context of scenario method chunks," in Advanced Information Systems Engineering, 1998, pp. 191-218.

[7] J. Ralyté, "Reusing scenario based approaches in requirement engineering methods: CREWS method base," in Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on, 1999, pp. 305-309.

[8] J. Ralyté and C. Rolland, "An assembly process model for method engineering," in Advanced information systems engineering, 2001, pp. 267-283.

[9] J. Ralyté and C. Rolland, "An approach for method reengineering," in Conceptual Modeling—ER 2001, ed: Springer, 2001, pp. 471-484.

[10] I. Mirbel and J. Ralyté, "Situational method engineering: combining assembly-based and roadmap-driven approaches," Requirements Engineering, vol. 11, pp. 58-78, 2006.

[11] P. Oladimeji, M. Roggenbach, and H. Schlingloff, "Levels of testing," Advance Topics in Computer Science, 2007.

[12] I. Jovanović, "Software Testing Methods and Techniques," The IPSI BgD Transactions on Internet Research, vol. 30.

[13] H. M. Sneed, "Measuring the Effectiveness of Software Testing," SOQUA/TECOS, vol. 58, p. 109, 2004.

[14] M. walia, "Realizing efficiency and effectiveness in software testing through comprehensive metrics model.",2014