

Research Article

Effectiveness of Model Based Testing: An Evidence from Pakistan

Faisal Nawaz, Faisal Shafique Butt, Wasif Nisar and Irfan Sharif

Department of Computer Sciences, COMSATS Institute of Information Technology,
Wah Cantt., 47040, Pakistan

Abstract: This research study investigates the relationship between the use of Model Base Testing (MBT) and organizational performance. MBT is the submission of Model Based Design for scheming and optimally performing the essential objects to achieve software testing. Models can be custom to characterize the anticipated conduct of the System under Test, or to characterize the anticipated testing policies and testing situation. Use of MBT is explored in Pakistani context through discussion of different approaches of MBT. Data has collected through online questionnaires and then frequencies bar charts have been created and used to represent the responses which indicate that the use of MBT increases reliability, effectiveness and organizational performance. But cost reduction is not applicable in Pakistani context due to its introductory stage.

Keywords: Model, testing, Pakistan, performance, System under Test (SUT)

INTRODUCTION

Every time the chief determination of software testing is to catch flaws and raise the sureness in the system under test (Malik *et al.*, 2010a) by inspection whether the application imitates to the stipulations (Malik *et al.*, 2010b). The customers of software industries are facing numerous challenges for getting their increased business value. The customers are demanding supple, consistent and well-organized solutions for their software. Moreover, the customers want their software to be used within different environments. Today's research came to a point that it takes about 60% of the development effort for testing (Abhors *et al.*, 2010). Every tester distinguishes that interpretation and sympathetic code is greatly tougher as compare to interpretation and sympathetic high level description of the system (Liu *et al.*, 2007). Active test coverage needs not only a well prearranged Testing exertion, but also needs testable necessities. Necessities are frequently described as not appropriate for testing, because they are, for example, imperfect or too unclear (Hasling *et al.*, 2008).

At the early stages manual testing was adopted which was a time and cost consuming practice. Due to time consumption, the delivery of the software was delayed and hence increasing the cost of software as well which makes software over budgeted. With the passage of time a new methodology was adopted which named as "Modeling". Modeling is the graphical representation of an object and the testing evolution has also been made model based testing (Schieferdecker,

2012). During modern centuries the development of software in the automotive industry controlled to a development process originated on model based skills which have much compensation for automotive developments (Bringmann and Kramer, 2008).

The overall awareness of model-based testing (of deterministic systems) is a clear conduct model encodes the proposed conduct of an application called system under test (Pretschner *et al.*, 2005). Thus we can say that automation of the test design is recognized as model based testing or automation generation of the test from models of system under test SUT. The process of MBT starts from the generation of test cases from any specific models then it moves towards implementation of the model and finally it compares output behavior to check its conformance between implementation and its respective specification (Schieferdecker, 2012) but the testing results depends upon the models, methods and valid only if the model accurately represents the behavior of the units (Stenbakken, 1996) because In model-based development, numerous subjects are inter-related. Any change in one artifact can influence other related artifacts; therefore, it is essential to provide for these relations and dependencies for successful testing (Farooq *et al.*, 2010). For example Model Based Testing automates black box testing based on the model of the system under analysis (Aydal *et al.*, 2009). So Model Based Testing makes utilization of models as a foundation for generating tests for the System Under Test (SUT). When appropriate models are accessible, this can be a influential approach providing automated generation of test cases for the system under test SUT

Corresponding Author: Faisal Shafique Butt, Department of Computer Sciences, COMSATS Institute of Information Technology, The Mall, Wah Cantt., 47040, Pakistan, Tel.: +92-321-5029899

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

(Kanstrén, 2009). In earlier stages, following questions are identified:

- What are the challenges which occur when we apply MBT on different testing phases?
- For different phases of testing can we use same modeling language?
- At different phases which type of benefits does MBT delivers (Dalal *et al.*, 1999)?

But later on, When MBT entered into the industry, even though its possible gains were massive. Its potential benefits include:

- Enhanced documents of test cases, robust visualization and improved communication among developers and testers.
- Facilitation of mechanically producing determined tests and evaluation of optimized test report.
- The capability to guess and select regression test bundles.
- Enhanced test quality by model based quality study.
- Less complicated changes of test models and suites with regard to changed requisites and models and with regard to new product or service versions and sophisticated preservation of test conditions.
- Trimmed timetables and costs
- Before time, precise specification, modeling and evaluation of system actions and in the early hour's detection of specification mistakes (Schieferdecker, 2012).

The advantages of MBT are high coverage, clear traceability and less manual effort (Mlynarski, 2010). Software testing procedure composes of different type of testing phase s which comprises of component testing, integration testing, system testing and installation testing. During these testing stages, different defects are normally found and removed. Quality of any software depends upon which software development process being followed. The component or module testing is the mainly time serious portion in software testing so all modules' testing can be completed in limited time and such type of activates consumes 40-50% of the whole resources of software development process (Htoon and Thein, 2005).

Another study has overviewed of the MBT methods were overviewed and practiced in another study. First-generation MBT comprised test making of system models. Actual test behavior and test generation were reformed by extending main ideas and algorithms

but the range of limitations such as using same model for generating test and code still remained. The detection of errors was nearly impossible because errors were transmitted to code and tests. The models which were used in second generation were scenario models, usage models, environmental models etc. Test models grant us the skill of testing fears in devoted model. This helps us in eliminating the duality of system and test system in model stage. This approach also helps us to gain access to conditions level, which differentiates between system requirements and test requirements and the models. During high level analysis; model driven software engineering and documentation is properly used to design tests by methods and tools established as industrial practice. The quality of tests including rightness and reporting can be determined earlier at model level (Schieferdecker, 2012).

In another study, it has been discussed that hybrid method of testing for making test cases which presented a technique which draws benefits of model-based testing and software architecture very distinctively. They used an easy Client/Server system for showing the realism regarding their testing method (Reza and Lande, 2010).

Another study has reviewed relevant literature and found many MBT situations that vary and this happens especially in test models area and these situations affect the ability to detect error. They suggested tabular examination for MBT situations from the viewpoint of a test manager. Thus, they studied the attempts initiated by several organizational features and testing activities. In order to draw the comparisons, they established criteria and stipulated metrics to differentiate the situations under consideration. For the purpose of defining metrics in a systematic way, they applied Goal-Question-Metric (GQM) method. This method allows objective comparisons because of using systematically derived metrics with the help of GQM technique. GQM method permits expansion of metrics, in case of emergence of additional comparison criteria in the future (Guldali *et al.*, 2010).

It has been developed a uniform method for designing specifications of model-based testing. It started from the practical approach of action-word. In which the researchers employed formal semantics in the arena of specification language for increasing the precision of action-words based testing. Furthermore, the notion of coverage had also been added in the original action-word technique. The need to specify the test automata is neutralized by the sequential logic approach. Employing LTL based modalities and the use of predicates has added to the advancements in the use of several test specifications (Krishnan, 2004).

While another research study has examined an automotive network controller for reviewing various test suites regarding error detection, coverage of model and implementation coverage. This eventually found that the actual number of programming errors detected had no direct relation with the models used in the testing experiment. The test suites that were based on automatically generated models detected the same number of errors as to those testing suites that were based on hand-crafted models. They further argued that an increase of six times in the model-based test numbers would lead to an 11% increase in the number of errors to be detected (Pretschner, 2005).

In another case study has conducted a survey of MBT users to create a profile of existing users of MBT to offer details about why, how, who and what. They were checked all the previous reports related to profit and cost, assumptions about success and failure factors and how software process aspects and MBT are associated to each other. The result specifies that MBT is an efficient and realistic software development technology (Binder, 2011).

Hence we adopted that survey items for this study to check the obstacles, reimbursement and expenses, statements regarding achievement or be unsuccessful issues, satisfaction and reliability of the use of MBT in Pakistani scenario and then check the relationship of MBT and organizational effectiveness.

METHODOLOGY

We operationalized the MBT in terms of cost, approach, reliability and effectiveness which are taken as the independent variables. We adopted seventeen items from an already developed survey paper (2011, Model-based Testing User Survey: Results and Analysis) by Robert V. Binder containing 40 items for this study to check the obstacles, reimbursement and expenses, statements regarding achievement or be unsuccessful issues, satisfaction and reliability for the use of MBT in Pakistani scenario for which data were collected from the following software houses in the premises of Lahore and Islamabad:

- Talented Earth Organization
- Iengineering PVT Ltd
- Ikonomi, Landmark recourses Pakistan
- Qzlogix
- System limited Lahore,
- InfoTech Private limited Lahore

Then SPSS version 17 will be used for the data analysis in which descriptive statistics, frequencies tables and bar charts will be developed for data representation.

THEORETICAL FRAMEWORK

Variable explanation: There are one independent variable “use of MBT” which is operationalized in terms of cost, approach, reliability and effectiveness and one dependent variable “organizational performance (Fig. 1).

Model based testing: We can define model based testing as a real life application of “Model based design” used to design and execute the essential procedures for carrying out “software testing”. Such models can be applied for representing the preferred performance of the System under Test (SUT), or to show the preferred testing environment and testing strategies. The Fig. 2 shows this approach.

For this study, we have operationalized the use of MBT in terms of cost, time, quality and reliability and employee’s satisfaction. So we have adopted user survey items developed by Binder (2011)

Organizational effectiveness: We have developed two items for the measurement of organizational effectiveness i.e:

- Overall, how effective for the organization do you think MBT has been?

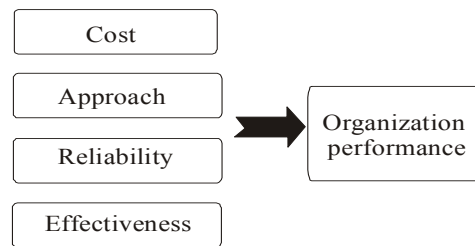


Fig. 1: Relationship between use of MBT and organizational performance

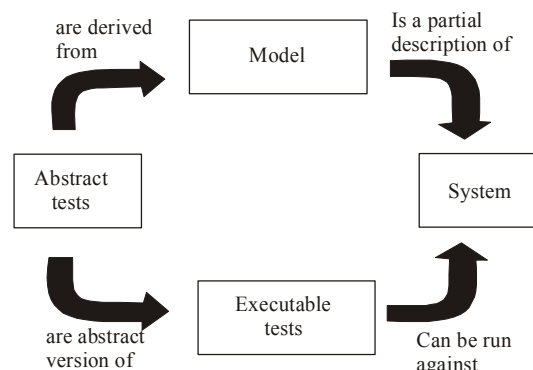


Fig. 2: General model-based testing setting

- Does organization performance increase?

Hypothesis:

- H1:** Use of MBT is effective for the organization for cost reduction
- H2:** Use of MBT increase reliability
- H3:** Use of MBT is an effective method
- H4:** Use of MBT increase organizational performance

SAMPLING AND DATA COLLECTION

Sample size and sampling technique: The sample size is 50 and convenience sampling technique is used for this study because convenience sampling is non-probability sampling techniques where respondents are choose because of their convenient accessibility and closeness to the researcher.

Data collection: Data were collected for this study through online questionnaires. Responding men and women were from IT firms. Total fifty questionnaires were sent and thirty seven were returned. So, the response rate is 74% (Table 1 and 2).

Data analysis and interpretation: Data were examined using SPSS (version-17). Reliability test, descriptive statistics, correlation and linear regression were applied on the data for the purpose of data analysis and interpretation. We used bar charts for the results representations.

Measures of variables: Dependent: Organizational performance was measured using two items, which were developed for this study. Independent: Use of MBT was measured using fifteen items which was adopted by Robert V. Binder (2011). Responses were on a different number of point scales for each item.

Descriptive statics: Descriptive statics are used for the frequencies bar charts for presentation of the results of each item.

Approach: Figure 3 shows the types of MBT approaches which are coded as:

- Agile
- Waterfall
- Ad hoc
- CMMI level 2+
- Incremental/Rational unified process
- Team software process
- Extreme programming/Test driven
- Spiral
- What were the generally accepted process/ approach used for the MBT project?

Table 1: Shows 29% people belongs to business organization

Types	Frequency
Business	29
Educational/ research	6
Government	2
Total	37

Table 2: Shows 27 respondent organization's size is 11-1100

No of employees	Frequency
1-10	2
11-100	27
101-500	5
501-1000	2
1001-10000	1
Total	37

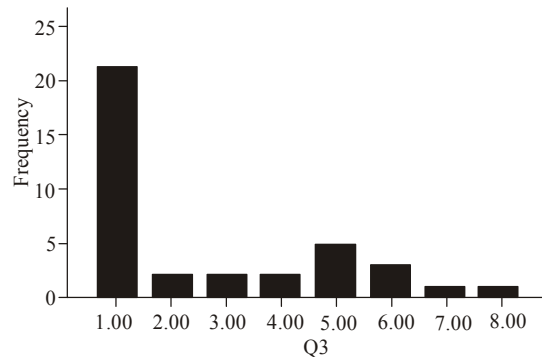


Fig. 3: User software process

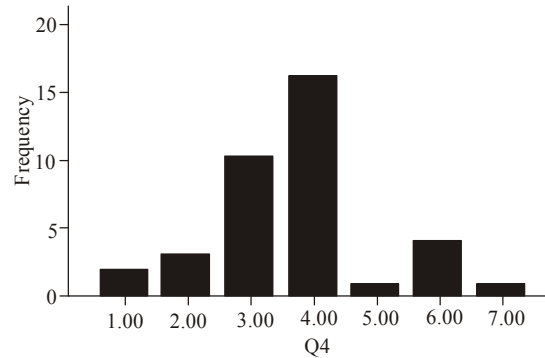


Fig. 4: User MBT process

Results showed that the 60% respondents used agile approach for the MBT project.

Figure 4 shows general application domain of the System under Test which is coded as:

- Embedded Controller
- Transaction processing
- Communications
- Software Infrastructure
- Simulation/supercomputing
- Social Media

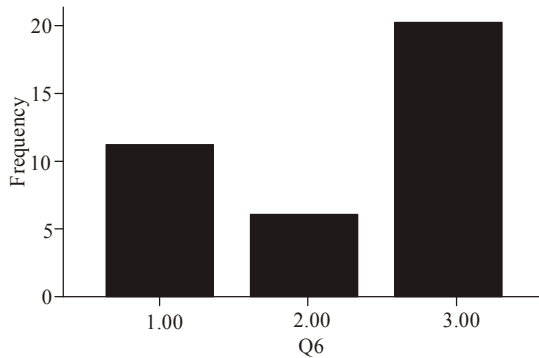


Fig. 5: MBT provider type

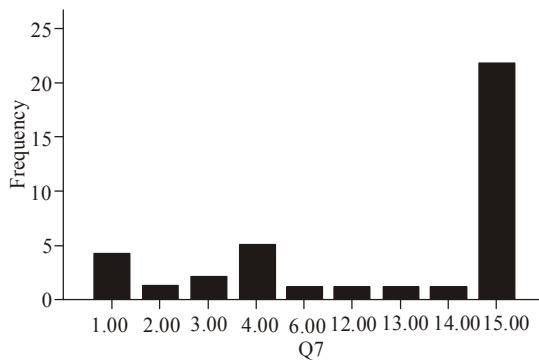


Fig. 6: Commercial MBT product

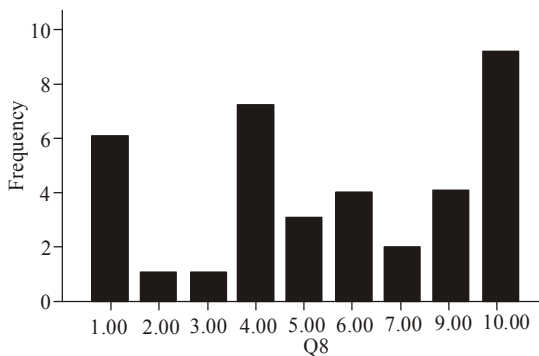


Fig. 7: The common approach for linking MBT tool output, SUT control interfaces and SUT observation interfaces

- Gaming
- What is the common application area of the System under Test?

About 50% respondents reported that they used software infrastructure and 23.33% used communication as domain for system under test. Figure 5 shows what type of MBT apparatus were used in the development which are coded as:

- Commercial
- Custom Developed
- Open Source
- What type of MBT tools were used in the project? More than 50% respondents reported that they used open source MBT tools in the project and rest of them used other kinds of tools almost equally.

Figure 6 shows which commercial tool was used mostly which are coded as:

- Spec Explorer (Microsoft) Matelo (All4tec)
- Conformiq Tool Suite (Conformiq)
- MBTsuite (sepp.medgetmore)
- T-Vec/RAVE (T-Vec Technologies)
- Rhapsody Automatic Test Generation (IBM/Rational)
- Reactis (Reactive Systems)
- No Magic
- NI TestStand
- Enterprise Architect
- ElviorTestCast MBT
- CertifyIt (Smartesting)
- ATD-Automated Test Design
- NONE
- If you used a commercial tool, which one?

More than 60% respondents reported that they do not use any commercial tool.

Figure 7 shows the universal method for linking MBT apparatus output, system under test control interfaces and system under test surveillance interfaces which are coded as:

- The tool/method produces code/data for an adapter
- Which abstracts the SUT interface?
- The tool produces code/data for the SUT Interface.
- The tool produces code/data which is more Processed to make executable test cases.
- Manual programming/scripting is necessary to Take effect produced test cases to the SUT Interfaces.
- Test cases are manually entered and observed.
- What is the all-purpose method for linking MBT tool output, SUT control interfaces and SUT observation interfaces?

23.33% respondents reported that they used the approach in which Test cases are manually entered and observed and 30% are used interface approach and 16.67% used the approach in which the tool produces code/data for an adapter.

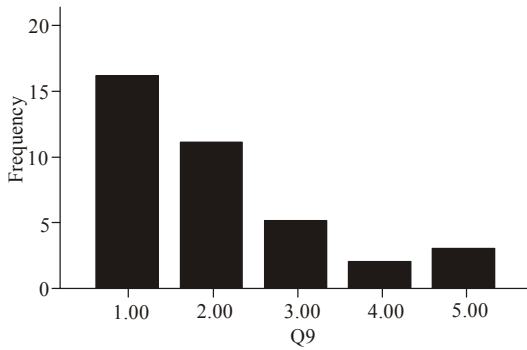


Fig. 8: Aspect of the SUT

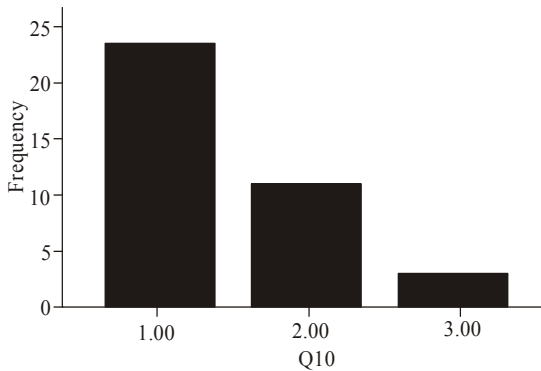


Fig. 9: Scope of MBT

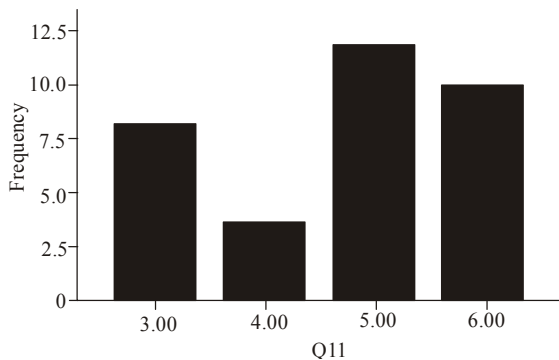


Fig. 10: Percent increase or decrease in escaped bugs

Figure 8 shows features of the SUT do you test with MBT which are coded as:

- Functional
- Reliability
- Performance
- Load/Stress
- Security
- Which features of the SUT do you test with MBT?

More than 45% respondents reported that they test functional aspects of the SUT with MBT and

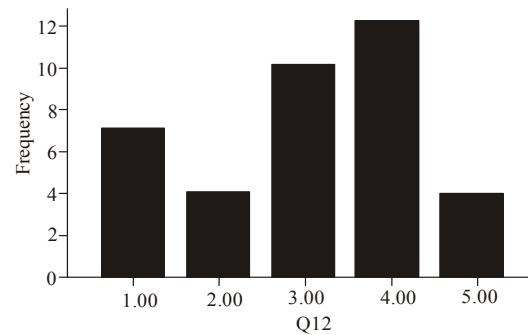


Fig. 11: MBT problems

23.33% used for reliability and 13.33% used for performance.

Figure 9 shows the scope using MBT which are coded as:

- System
- Component
- Unit
- At what scope do you use MBT?

MBT is employed at all levels of testing: above all at system scope more than 50%, at module scope 32.14% and at unit scope for 10.71% respondents reported.

• **Reliability:** Figure 10 shows that the top estimation of the percent amplify or decline in runaway bugs after testing with MBT which are coded as:

- -500%
- -400%
- -300%
- -200%
- -100%
- 0

What is your most excellent approximation of the percent boosts or reduces in runaway bugs following testing with MBT?

32.14% Respondents reported that on normal testing duration reduced by 100%, with the median upgrading at 18%. 35.71% respondents reported that there is no difference and 30% reported median reduction in runaway bugs after testing with MBT.

Figure 11 shows the responses' about the some common MBT problems which are coded as:

- Not Applicable
- Not a Problem
- Better than expected
- As Expected

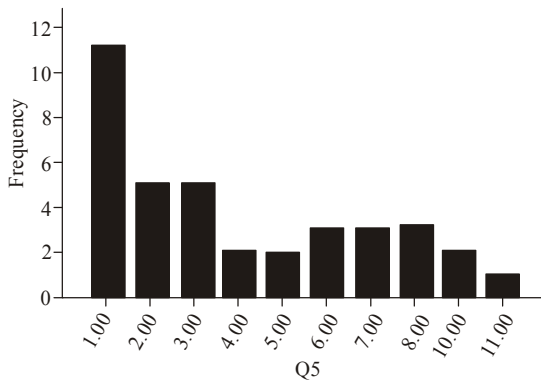


Fig. 12: impetus for using MBT

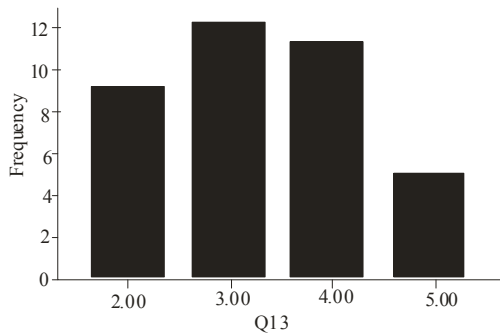


Fig.13: Testing cost of MBT

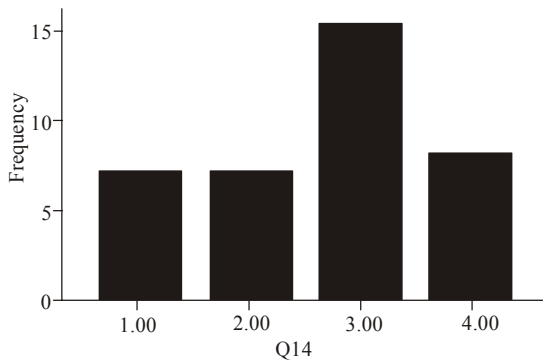


Fig.14: Effectiveness of MBT

- Worse than expected
- What has your knowledge been with some common MBT problem

More than 25% respondents reported that they found MBT as reliable as better than expected and 37% reported that they found errors as expected.

Cost: Figure 12 shows the factors led to your project using MBT which are coded as:

- likelihood of reduced testing costs and/or work
- Option of less bugs
- Prospect of reduced testing time
- Evaluation of MBT technology/trade study
- Academic research project
- Time, cost, or schedule pressures
- Personal curiosity
- Organization-wide adoption
- Compliance with external standards
- Better coverage of tests
- Boost of Test Case quality Test automation
- maintenance
- Separation of test design and test development
- What features guided to your project using MBT?

As the result showed that 30% respondents reported that they used MBT for the option of reduced testing costs and/or work and the likelihood of less bugs and possibility of reduced testing time are also important factors for using MBT.

Figure 13 shows the estimation of the percent amplifies or diminishes in testing costs with MBT which is coded as:

- -100%
- -50%
- 0%
- 50%
- 100%
- “What is your best estimate of the percent increase or decrease in testing costs with MBT? “

Only 17.24% respondents reported that 50% cost is decreased by using MBT and 37% reported that no effect on cost and more than 30% reported that cost is increased.

Effectiveness: Figure 14 shows that the responses about the effectiveness of the use of MBT which are coded as:

- Extremely effective
- Moderately effective
- Slightly effective
- No effect
- Slightly ineffective
- Moderately ineffective
- Extremely ineffective
- On the whole, how successful do you think MBT has been?

24% respondents reported that the use of MBT is extremely effective and 20 and 34% reported as moderately effective and slightly effective respectively

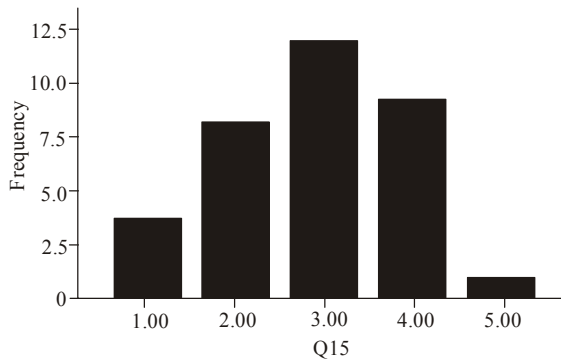


Fig. 15: Use of MBT

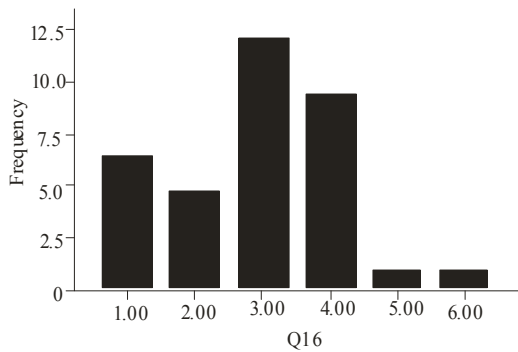


Fig. 16: Effective for organizations

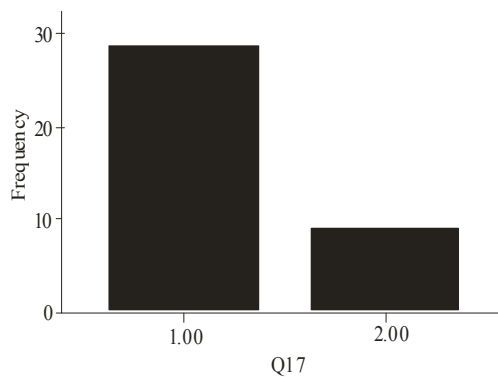


Fig. 17: Organizational performance

as well as 20% respondents said that use of MBT is not effective at all.

Figure 15 shows that how respondents like to continue to use MBT which are coded as:

- Very likely
- Extremely likely
- Fairly likely
- A little likely
- Not at all likely

- How probable are you to carry on utilizing MBT?

As the results show that all the respondents want to continue to use of MBT.

Figure 16 show the Overall effectiveness FOR THE organization by using the MBT which are coded as:

- Extremely effective
- Moderately effective
- Slightly effective
- No effect
- Slightly ineffective
- Overall, how effective for the organization do you think MBT has been?

As the results show that use of MBT is effective for the organizations as 24% respondents reported that MBT is extremely effective but on the other hand 34% said that use of MBT has no effect for the organization.

Organization performance: Figure 17 shows the increase or decrease of organizational performance which is coded as:

- YES
- NO
- Does organization performance increase?

More than 70% respondents reported that the organizational performance is increased with the use of MBT.

CONCLUSION

As this study proposed the four hypotheses:

- H1:** Use of MBT is effective for the organization for cost reduction
- H2:** Use of MBT increase reliability
- H3:** Use of MBT is an effective method
- H4:** Use of MBT increase organizational performance

On the basis of the respondents results, H2, H3 and H4 are supported but H1 is rejected. H1 is rejected which is not expected but there is certain reasons especially in Pakistani context. As the MBT is a type of technique which is not commonly use in Pakistan that's why organizations that use this technique, they need to train their employees which increase their implementation cost of MBT tools. H2 is supported as the reliability is increased by using MBT. This is evident that if we adopt MBT tools instead of manual testing then it will reduce the bugs but it is obvious that bugs could never be fully removed. As no one experience any worse problem with the use of MBT. H3 is also supported which means use of MBT is highly

effective for the organizations. It argues that if the organizations being aware of their employees with the MBT tools then it will reduce testing time which is very crucial in software development and it will be very helpful for the customer and market growth. So, the use of MBT is being critical as the world transform into automation tools that's why the organizations are willing to use the MBT tools in future.

H4 is also supported as the organizational performance depends heavily on three factors i.e. time, cost and quality. If we use MBT tools then it will reduced testing time, cost as there is no need to hire a third party for the testing processes. Quality is also increased as the number of bugs decrease. In short, use of MBT is highly effective for the organizational performance.

In Pakistan, agile is mostly used approach for the MBT project and Software infrastructure and communications used as domain for system under test. Open sources MBT tools are being used in software projects and mostly organizations do not use any commercial tool as it is obvious that if the open source tools are available in the market then there is no need to purchase a commercial tool. Organizations are used the different types of approaches i.e., mostly Test cases are used when the data is manually entered and observed and then some are used interface approach and very few used the approach in which the tool generates code/data for an adapter. Mostly software houses test functional aspects of the SUT with MBT and some software houses are also used for reliability and performance. MBT is employed at every scope of testing but mostly for system scope and then for component scope and unit scope.

ACKNOWLEDGMENT

First of all, we all thank to almighty God who gave us the knowledge, strength and resources to make it possible to us. Then we especially thank to Mr. Muhammad Sharif (Assistant Professor COMSATS institute of information technology, Wah, Pakistan), Mr. Mudassar Raza (Lecturer COMSATS institute of information technology, Wah, Pakistan) for their correspondence and support throughout this study. We are also thankful to all employees of software houses who provided us data for this research.

REFERENCES

Abbors, F., A. Backlund and D. Truscan, 2010. MATERA-an integrated framework for model-based testing. Proceeding of 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS), Oxford, pp: 321-328.

Aydal, E.G., R. F. Paige, M. Utting and J. Woodcock, 2009. Putting formal specifications under the magnifying glass: Model-based testing for validation. Proceeding of 2009 International Conference on Software Testing (ICST), Verification and Validation, pp: 131-140.

Binder, R.V., 2011. 2011 model-based testing user survey: Results and analysis. System Verification Associates, Vol. 33.

Bringmann, E. and A. Kramer, 2008. Model-based testing of automotive systems. Proceeding of International Conference on Software Testing, Verification and Validation, DOI: 10.1109/ICST.2008.45.

Dalal, S.R., A. Jain, N. Karunanithi, J.M. Leaton, C.M. Lott, G.C. Patton, *et al.*, 1999. Model-based testing in practice. Proceeding of International Conference on Software Engineering (ICSE '99), Los Angeles, CA, USA, pp: 285-294.

Farooq, Q., M. Iqbal, Z.I. Malik and M. Riebisch, 2010. A model-based regression testing approach for evolving software systems with flexible tool support. Proceeding of 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS), Oxford, pp: 41-49.

Guldali, B., M. Mlynarski and Y. Sancar, 2010. Effort comparison for model-based testing scenarios. Proceeding of 3rd International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Paris, pp: 28 -36.

Hasling, B., H. Goetz and K. Beetz, 2008. Model based testing of system requirements using UML use case models. Proceeding of 1st International Conference on Software Testing, Verification and Validation, Lillehammer, pp: 367-376.

Htoon, C.Y. and N.L. Thein, 2005. Model-based testing considering cost, reliability and software quality. Proceeding of 6th Asia-Pacific Symposium on Information and Telecommunication Technologies, APSITT 2005, Yangon, pp: 160-164.

Kanstrén, T., 2009. Behaviour pattern-based model generation for model-based testing. Proceeding of Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns (Computationworld '09), Athens, pp: 233-241.

Krishnan, P., 2004. Uniform descriptions for model based testing. Proceedings of the Australian Software Engineering Conference (ASWEC ' 04), pp: 96-105.

Liu, G., Q. Liu and P. Xie, 2007. Model-based testing and validation on knowledge-based systems. Proceeding of IEEE International Conference on Grey Systems and Intelligent Services (GSIS 2007), Nanjing, pp: 1242-1245.

- Malik, Q.A., A. Jaaskelainen, H. Virtanen, M. Katara, F. Abbors, D. Truscan, *et al.*, 2010a. Model-based testing using system vs. test models-What is the difference? Proceeding of 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS), Oxford, pp: 291-299.
- Malik, Q.A., D. Truscan and J. Lilius, 2010b. Using UML models and formal verification in model-based testing. Proceedings of the 17th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS '10), Washington, DC, USA, pp: 50-56.
- Mlynarski, M., 2010. Holistic model-based testing for business information systems. Proceeding of 3rd International Conference on Software Testing, Verification and Validation (ICST), Paris, pp: 327-330.
- Pretschner, A., 2005. Model-Based Testing.
- Pretschner, A., W. Prenninger, S. Wagner, C. Kühnel, M. Baumgartner, B. Sostawa, *et al.*, 2005. One evaluation of model-based testing and its automation. Proceedings of the 27th International Conference on Software Engineering (ICSE '05), pp: 392-401.
- Reza, H. and S. Lande, 2010. Model based testing using software architecture. ITNG '10 Proceedings of the 7th International Conference on Information Technology: New Generations, pp: 188-193.
- Schieferdecker, I., 2012. Model-based testing. *Software IEEE*, 29(1): 14-18.
- Stenbakken, G.N., 1996. Effects of nonmodel errors on model-based testing. *IEEE T. Instrum. Meas.*, 45(2): 384-388.