Software Testing: A Comparative Study
Model Based Testing VS Test Case Based Testing

Rakesh Reddy Polamreddy
Syed Ali Irtaza

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden
This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 + 20 weeks of full time studies.

Contact Information:

Authors:
1. Rakesh Reddy Polamreddy
   Address: Karlskrona, Sweden.
   E-mail: rakitheone@gmail.com

2. Syed Ali Irtaza
   Address: Karlskrona, Sweden.
   E-mail: s.ali.irtaza@gmail.com

University advisor:
Dr. Richard Torkar
School of Computing. Blekinge Institute of Technology.

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
ABSTRACT

Software testing is considered as one of the key phases in the software-development life cycle (SDLC). The main objective of software testing is to detect the faults either through manual testing or with automated testing approach. The most commonly adopted software testing approach in industries is test case based testing (TCBT) which is usually done manually. TCBT is mainly used by the software testers to formalize and guide their testing activities and set theoretical principals for testing. On the other hand, model based testing (MBT) is widely used automation software testing technique to generate and execute the tests. Both techniques are showing their prominence in real time with some pros and cons. However, there is no formal comparison available between these two techniques.

The main objective of this thesis work is to find out the difference in test cases in TCBT and MBT in terms of providing better test coverage (Statement, Branch and Path), requirement traceability, cost and time. To fulfill the aims of the research we have conducted interviews for static validation, and later we did an experiment for validating those results dynamically. The analysis of experiment results showed that the requirement traceability in MBT generated test cases are very hard to make the test cases traceable to the requirements, particularly with the open-source tool Model J-Unit. However, this can be done by using other commercial tools like Microsoft Spec Explorer or Conformiq Qtronic. Furthermore, we found by conducting experiment, that MBT consumes less time thus it is cost-effective as compared to TCBT and also MBT show better test coverage than TCBT. Moreover, we found that, in our case, requirement traceability is better in traditional TCBT approach as compared to MBT.

Keywords: Test case based testing (TCBT), Model Based Testing (MBT), Requirement traceability, test coverage, cost, time.
ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the people who has helped us in completing this thesis. First of all we are very much thankful to our supervisor Dr. RICHARD TORKAR for his valuable guidance, support and advices from the beginning to the very end. Due to his good supervision, guideline and encouragement we have been able to complete this thesis. In addition, we would also like to convey our gratitude to all the participants of interviews and experiments for their valuable time and inputs. We would also like to thank the department of school and computing for giving this opportunity.

In general, we are very much thankful to our parents for their encouragement and support without them we cannot be able to finish our thesis. Last but not the least; we are very thankful to our friends for their support and motivation.
CONTENTS

CHAPTER # 1 ................................................................................................................................................. 9

1 INTRODUCTION .......................................................................................................................................... 10
  1.1 BACKGROUND ....................................................................................................................................... 10
  1.2 PROBLEM DOMAIN ............................................................................................................................... 11
  1.3 AIMS AND OBJECTIVES ......................................................................................................................... 11
  1.4 RESEARCH QUESTIONS ......................................................................................................................... 11
  1.1 HYPOTHESIS ......................................................................................................................................... 11
  1.5 RESEARCH METHODOLOGY ............................................................................................................... 12
  1.6 RESEARCH DESIGN .............................................................................................................................. 13
  1.7 EXPECTED OUTCOMES ......................................................................................................................... 14
  1.8 THESIS STRUCTURE ............................................................................................................................ 14
  1.9 GOALS .................................................................................................................................................. 15
  1.10 LIMITATIONS ..................................................................................................................................... 15

CHAPTER # 2 ................................................................................................................................................. 16

2 MISCELLANEOUS DEFINITIONS .................................................................................................................. 17

3 UNDERSTANDING MODEL BASED TESTING AND TEST CASE BASED TESTING .................................... 17
  3.1 MODEL BASED TESTING ...................................................................................................................... 17
    3.1.1 Working of MBT ............................................................................................................................... 18
    3.1.2 Related Research ............................................................................................................................ 19
  3.2 TEST CASE BASED TESTING .............................................................................................................. 20
    3.2.1 Working with Test Case Based Testing (ISO / IEC 29119) ............................................................... 21
    3.2.2 Related Research ............................................................................................................................ 22

CHAPTER # 3 ................................................................................................................................................. 24

4 STRENGTHS AND WEAKNESSES OF MBT AND TCBT APPROACHES .............................................. 25
  4.1 INDUSTRIAL INTERVIEWS ..................................................................................................................... 25
    4.1.1 Purpose of Interviews ...................................................................................................................... 25
    4.1.2 Selection of Subjects ......................................................................................................................... 25
    4.1.1 Interview Structure .......................................................................................................................... 25
  4.2 RESULTS AND ANALYSIS .................................................................................................................... 27
    4.2.1 Ratings for TCBT and MBT .............................................................................................................. 28
    4.2.2 Results of TCBT and MBT .............................................................................................................. 30

CHAPTER # 4 ................................................................................................................................................. 32

5 DYNAMIC VALIDATION ............................................................................................................................. 33
  5.1 EXPERIMENT ......................................................................................................................................... 33
    5.1.1 DEFINITION ................................................................................................................................... 33
  5.2 PLANNING .............................................................................................................................................. 34
  5.3 EXPERIMENT DESIGN ........................................................................................................................... 35
    5.3.1 Terminologies .................................................................................................................................. 36
  5.4 EXPERIMENT EXECUTION ...................................................................................................................... 37
  5.5 RESULTS AND ANALYSIS .................................................................................................................... 38
    5.5.1 Test Coverage .................................................................................................................................. 38
LIST OF FIGURES

FIGURE # 1: RESEARCH DESIGN................................................................. 13
FIGURE # 2: MBT PROCESS................................................................. 19
FIGURE # 3: TCBT PROCESS................................................................. 22
FIGURE # 4: INTERVIEWEES (DESIGNATIONS) AND THEIR EXPERIENCE (YEAR)................................. 26
FIGURE # 5: TEST COVERAGE CONSTRUCT........................................... 28
FIGURE # 6: REQUIREMENT TRACEABILITY CONSTRUCT................................. 29
FIGURE # 7: UNDERSTANDABILITY CONSTRUCT........................................... 29
FIGURE # 8: COST AND TIME CONSTRUCT............................................ 30
FIGURE # 9: TEST DESIGN AND PLANNING CONSTRUCT.................................. 30
FIGURE # 10: CROSS PLOT GRAPH FOR TEST COVERAGE AND TEST CASE DETAIL................................. 31
FIGURE # 11: CROSS PLOT GRAPH FOR REQUIREMENT TRACEABILITY, TIME AND COST................................. 31
FIGURE # 12: EXPERIMENT DESIGN.......................................................... 35
FIGURE # 13: TIME GRAPH FOR MBT AND TCBT TEAMS................................. 39
FIGURE # 14: CALCULATED COST FOR THE MBT AND TCBT TEAMS......................... 40
LIST OF TABLES

TABLE # 1: LIST OF DEPENDENT AND INDEPENDENT VARIABLES.................................................. 34
TABLE # 2: TEST COVERAGE IN MBT AND TCBT............................................................................. 38
TABLE # 3: TIME AND COST FOR MBT AND TCBT TEAMS. ............................................................. 39
TABLE # 4: TCBT TEAM -1: REQUIREMENT TRACEABILITY ............................................................. 40
TABLE # 5: TCBT TEAM – 2: REQUIREMENT TRACEABILITY. ......................................................... 40
TABLE # 6: TCBT TEAM – 3: REQUIREMENT TRACEABILITY. ......................................................... 41
TABLE # 7: MANN-WHITNEY OBSERVATION TABLE....................................................................... 42
TABLE # 8: MANN-WHITNEY RANKS TABLE. .................................................................................... 43
TABLE # 7: BACKGROUND INFORMATION TABLE......................................................................... 54
TABLE # 8: TEST CASE TEMPLATE FOR TCBT. ............................................................................... 55
CHAPTER # 1

INTRODUCTION

Introduction
- Background
- Problem Domain
- Aims and Objectives
- Research Questions
- Hypothesis
- Research Methodology
- Research Design
- Expected Outcomes
- Thesis Structure
- Goals
- Limitations
1 INTRODUCTION

1.1. Background

Software-development processes have been studied for decades, and it has been suggested that software-development is considered reliable when it is done through a systematic approach [30]. In addition, the selection of appropriate tools and techniques also helps to assure high-quality [32]. Within the software-development process, the role of testing is of utmost importance as it is regarded as being the primary activity to collectively establish quality assurance indirectly [36]. The traditional approach to software testing is where developers develop the code, which is later tested. Testing is traditionally defined as a process of executing test cases that are designed from test case designing techniques [7, 33]. A testing process can be either manual or automated. In automated software testing, all the software testing activities can be done automatically. Automated software testing is one, gaining its importance in recent times. However, it cannot replace manual testing completely.

Test case based testing (TCBT) is one of the manual testing techniques that has been employed in software industries. Test case based testing strategy is a traditional testing approach which emphasizes on test cases [18]. TCBT helps the software testers to guide and validate their testing tasks and also to set the theoretical principals for testing [18, 20]. In TCBT the test cases are planned and designed before the execution of testing [37].

The primary idea behind TCBT is to design and document the test cases along with the inputs, outputs and the functionality of the system that has to be tested [3, 20]. SWEBOK defined TCBT as designing of test case to validate the correct implementation of functional specification and which can also be mentioned as conformance, correctness or function testing [8].

Testing helps to measure the quality in software systems and according to the industrial experiences, it is said that the testing time usually depends on how big the system really is (i.e. system complexity)? [40]. However, minor changes to the complex system may require a huge amount of time and effort to verify the quality [40]. In order to overcome this, test automation rather than manual testing can be used. Model Based Testing (MBT) is one technique for test automation. MBT can be defined as the process of automating the test design where tests are automatically generated from a model of the System under Test (SUT) [43].

MBT can be seen as a kind of black box testing where test cases are derived from design models, which are projected from system models at a high abstraction level. MBT is tightly associated with commercial tools, which has been evolving in both usability and functionality lately [43]. One reason for selecting MBT among all other testing approaches is because it aims to solve problems like automating the design of functional test cases in order to cut down the design cost, generating a systematic coverage of the model, reducing maintenance costs of test suits since a traceability matrix can be generated automatically from requirements to test cases [43].
1.2. Problem Domain
A number of companies [40, 43] have used MBT for developing software through different tools and approaches. In MBT, the purpose is to test the source code just like in Test Case Based Testing (TCBT), but the actual difference between both approaches comes at the point where you move forward from the user stories (Requirement phase in SDLC) to the next phase. In MBT, models are generated from user stories, then test cases are generated from those models and in TCBT, test cases are generated directly from the user stories. Our research is based on comparing test cases from the MBT process and the TCBT process in order to identify the potential differences in terms of coverage (i.e. path, statement and branch coverage), requirement traceability, cost and time.

1.3. Aims and Objectives
The main aim of this study is to make a fair comparison between test case generation between TCBT and MBT.
Objectives:
• Find out the differences in test cases in MBT and TCBT, which can affect the test coverage (branch, statement and path), requirement traceability, cost and time. To find out which technique is better in terms of providing a better test suite for testing software applications.

1.4. Research questions
RQ.1: What strengths and weaknesses are there in using MBT (in industry)?
RQ.2: What strengths and weaknesses are there in using TCBT (in industry)?
RQ.3: Which technique (MBT or TCBT) is better in terms of providing "better" test cases for software applications?

In research question 3, the term “better” is refer to as test cases which provide better test coverage and can be traceable.

1.1 Hypothesis
One experiment was conducted in academia on two different subjects (see details in “Experiment” section in Research Methodology below). The results obtained after the study will be compared to the following hypotheses:

H0,qual: The comparison between MBT and TCBT will result no significant changes in terms of test coverage (branch, statement and path), requirement traceability, cost and time.

H1: Comparing MBT and TCBT, the results from the experiment will reflect major changes in terms of test coverage (branch, statement and path), requirement traceability, cost and time.
1.5. Research Methodology

Our research will comprise of two parts, i.e. Interviews and experiment, through which we will answer all of our research questions. Our motivation for choosing these two methodologies is because we wanted to include industrial practices in our research. In order to calculate the test coverage, time and cost we had to use experiments instead of SLR. Those two parts are explained in detail below:

1. Interviews

The purpose of this phase is to gather industrial data for TCBT and MBT, and to understand the practices, implementation and plausible problems faced during the implementation of MBT and TCBT in industry. Interviews will be used to gather the information required. The interviewees will comprise of the staff ranging from project managers to software developers and testers. The interview subjects were selected based on convenience sampling. The overall population for the interview is 15 based on their responses we got 3 participants for MBT and 3 for TCBT. The results obtained from interviews will then be analyzed. A semi-structured [13] interview method approach will be adopted, which would include structured and unstructured interview questions [13, 38]. Semi-structured interview method is selected for data gathering because both the closed-ended and open-ended questions will help in fetching the desired data for our research, and this would also keep the interview focused. This part of the case study will answer RQ1 and RQ2.

2. Experiment

The second part of our research will be an experiment in the academia. The participants in the experiment will be two groups (A and B) of students from Software Engineering and Computer Science disciplines. The experiment subjects were selected based on convenience sampling, which is a type of probability sampling. Group A will contain individuals who are trained in TCBT. Group A will develop test cases from a set of user stories. Group B will develop EFSM models (Java programming language) in detail from the user stories, which will be used to generate the test cases. Coverage (branch, statement and path), requirement traceability, cost and time will be measured from the result of comparison of tasks from Group A and B. we will most probably show the result as a bunch of different metrics.

Therefore, the independent and dependent variables [44] selected in the experimentation are:

*Independent variables:*
- Approaches (TCBT & MBT)
- Tools support (for MBT test case generation)
- Personnel experiences

*Dependent variables:*
- Coverage (branch, statement and path).
- Requirement traceability
- Cost
- Time
After getting the results from above experiments, we will analyze the results and compare the analysis report with our hypothesis. This part will answer our RQ3.

### 1.6. Research Design

The research design contains two parts. First part includes qualitative research in which interviews were conducted and the second part was quantitative research in which experiment was conducted.

![Research Design Diagram](image)

**Figure #1: Research design.**
- **Qualitative Research**  
The qualitative research contains interviews through which the data was extracted and answers of RQ.1 and RQ.2 were processed. The reason for doing qualitative research was that; through qualitative research, we got to know the opinions of the industrial professionals in software testing, which is impossible to gather via quantitatively. The answers from the interviews were analyzed, and key points were separated from the general discussion. To validate our findings at this stage, we did a control experiment in the academic setup.

- **Quantitative Research**  
Quantitative research contained an experiment in an academic setup. Experiment was necessary in order to validate our findings during the interview phase. This part of the research was used to answer the RQ.3. Through quantitative measures, we calculated test coverage (branch, statement and path); time consumed, cost and requirement traceability of the test cases. On the basis of quantitative measures, we concluded our research and validated our findings during the interview phase.

**1.7. Expected outcomes**  
The expected outcomes of this research will be:

- Significant changes will be seen in the comparison of TCBT and MBT in terms of various aspects like test coverage (i.e. branch, statement and path), requirement traceability, cost and time.

**1.8. Thesis Structure**  
**Chapter 1** (Introduction): This chapter highlights the selected problem domain, background of topic, research aims, objectives, expected outcomes and adopted research methodologies, which will be utilized in this study.

**Chapter 2** (Background on Industry Practiced MBT and TCBT Approaches): This chapter illustrates some of the most important definitions and also defines the concepts related to MBT and TCBT. Furthermore, this chapter provides the details of currently used test processes related to TCBT and MBT. Furthermore, it describes the related industrial case studies and experiences in working with MBT and TCBT.

**Chapter 3** (Static Validation: Strengths and weaknesses of MBT and TCBT Approaches): This chapter gives details about the strengths and weaknesses associated with the use of both testing approaches i.e. MBT and TCBT as identified by interviewees.

**Chapter 4** (Dynamic Validation): This chapter provides the dynamic validation of the proposed process based on the feedbacks of industry professionals. Further in the chapter, it provides detail information about the experiment design and variables that are necessary to conduct the experiment in order to validate MBT and TCBT process. In the end of this chapter results are analyzed, interpreted and presented.
Chapter 5 (Epilogue): This chapter presents the validation threats in this study and the conclusion.

Chapter 6 (References)

Chapter 7 (Appendix)

1.9. Goals
There are three main goals of this thesis work:
- To point out the potential differences in test coverage (i.e. branch, statement and path); cost, time and requirement traceability between test cases generated in the process of TCBT and MBT.
- To understand the benefits in working with TCBT and MBT.
- To find out if there can be any future extension of MBT approach (if possible).

1.10. Limitations
This thesis has the following limitations:
- Models from the functional requirements in MBT were generated manually by students in java language. Then, those models were given as input to the Model J-Unit tool, which generates the test cases automatically. Test cases from the functional requirements in manual testing were generated manually by students of computer science and software engineering disciplines.
- In this research, we are just focusing on the Finite State Machines (FSM) [28] as it is widely used in the software design phase to model the behavior of the application in different scenarios. (Other modeling languages like State Charts [27], Markov Chains [17] and Unified Modeling Language (UML) [39] will not be the focus of this study).
CHAPTER # 2

BACKGROUND ON INDUSTRY PRACTICED MBT AND TCBT APPROACH

Background and Understanding the two approaches
- Process for TCBT and its related research
- Process for MBT and its related research
2. **MISCELLANEOUS DEFINITIONS**

- **Statement coverage (block coverage)**
  Generally, in software testing, testers are often required to generate test cases to execute every single statement at least once [31]. A test case can be served as input to the program under test, and is executed during testing. Statement coverage is typically defined as, the fraction of total number of blocks (or) statements that are executed by test data [31, 46].

- **Branch coverage (or decision coverage)**
  The branch coverage is defined as, the fraction of total number of decisions (or) branches that are executed by test data [25, 39]. It also helps in validating all the possible branches in the program and makes sure that no branch results to abnormal behavior of application [47].

- **Path coverage**
  In path coverage, test case is executed at least once i.e. all the execution paths of the program from entry to exit are executed during testing [45, 47].

- **Requirement Traceability**
  Requirement traceability is defined as “the ability to describe and follow the requirement in both a forward and backward direction” [4]. by “defining and maintaining relationships to related development artifacts” [29] such as code, configuration files and test cases. Testing is considered as the significant part in the software-development life cycle, which is used by many software-development teams. Having many test cases leads to increase’s effort and cost spent on testing, thus many industrial practitioners like developers, testers and managers giving a lot of importance to traceability [16, 41]. There is availability of some tool support to maintain, retrieve and record to trace information manually [48]. However, this is time consuming, labor-intensive and error-prone [9, 16]. It is more convenient and important to create, maintain and find the links of traceability in testing through an automated process as requirement traceability links goes out-of-date when software evolves.

3. **UNDERSTANDING MODEL BASED TESTING AND TEST CASE BASED TESTING**

3.1. **Model Based Testing**

A good software tester cannot avoid models [14]. In order to test one simplest software solution, testers have to create models in their minds and apply test procedures according to those models
Models are important when quality is in focus, and it is the future of software-development, when quality will be the only distinguishing factor between the vendors [14].

MBT is used to test the application thoroughly by explicitly defining the models in the design phase and then by creating test cases according to those models.

3.1.1. Working of MBT
In short, the steps in working with MBT include the understandability of the system under test (SUT), i.e. the software is developed according to the right requirements. After that design models are generated from those requirements, and then test cases are generated by using a tool from those designs. Finally, after implementing the system, testing is done according to those generated test cases.

These steps are described in detail below.

1. Understanding the SUT
In this step, the requirements are understood and the SUT is analyzed properly with all possible environment behaviors. The prototype creation is a good way to understand the SUT because in model based testing understanding the environment is as important as understanding the SUT itself.

A model of the SUT is created on the basis of requirements, e.g. the software requirement specification document (SRS). It is developed with the understandability of behaviors of SUT under different environments at different abstraction levels [42]. One common abstraction can be each possible input connected to the output.

2. Test Selection Criteria and Test Case Generation
This is one of the most important steps in MBT because the purpose is to define test cases, which should be good enough to detect severe and likely faults at acceptable cost. The test selection criteria are developed in order to choose specific behavioral models of the SUT that would require testing, i.e. indirectly it describes a test suite [42, 43].

After having test selection criteria, test case specifications are developed which transforms the test selection criteria to an operational profile, i.e. test cases are generated at this stage.

3. Test Case Execution
This step can be performed by recalling that the model and the SUT are at different abstraction levels, which should be bridged [42]. It is mainly done by executing the test case by giving a concrete input to the SUT and recording its output [42]. The input is concretized and abstracted by a component known as an “adaptor”. It also takes care of output by making it abstract [42, 43]. After that, output is compared to the expected result. The result can be pass, fail or inconclusive (in the case when decision cannot be yet made).
3.1.2. Related Research

Model Based Testing is not a new concept in the software-development and testing world. Research on MBT started in late 1990s, and with time engineers and researchers began to understand the power of modeling and testing in the software world. Some of the studies, in MBT, are mentioned next.

In one case study [35, 43], the researchers derived test cases from models, which consisted of abstract sequences of input and expected outputs. Those test cases were then converted into an executable form that allowed the researchers to apply all the test cases with conformance to the
model and the implementation. Researchers also applied input parts of each test case to the model and measured model and implementation coverage at the level of the model. Test suites generated from the models detected 11% more requirement faults as compared to test suites generated directly from the requirements. The study [35, 43] shows that making models results in detecting two to six times more errors.

In another case study [43], at Microsoft, a test automation tool for MBT “Test Model Toolkit (TMT)” was developed by the Internet Explorer team in 2001; by the end of 2004, it was being used by 20 teams within Microsoft. They next conducted an experiment in which the BizTalk team spent around eight weeks to generate test suites manually, while with the help of TMT they managed to generate the same test suites in one week (and with more test cases increasing the code coverage by 50%).

At IBM, engineers and researchers found significant cost reductions and good fault finding effectiveness when performing two studies where they used a model based test generator “GOTCHA-TCBeans” [43]. In the first case study [43], they tested the locking feature of the Portable Operating System Interface (POSIX) which was already tested by a manually developed test suits. The effort done for manual testing was 12 person months and the defects found were 18. MBT took 10 person months and detected two additional defects. Also the overall cost for MBT was reduced by 17%.

In the second case study by IBM [43], testing was done in order to validate a part of a java garbage collector. The SUT used in this case study contained 6914 LOC (lines of code). The SUT was initially tested by using specJVM and JCK1.3, through which they found three design errors and 13 implementation errors with statement coverage of 78%. After that they used MBT with FSM (Finite State Machines) models and ran the tests generated from the MBT. They found additional two design faults and two implementation faults. Also, the statement coverage was raised from 78% to 83%. Finally, they mentioned that the time spent on modeling and testing was 3-4 person months and they claimed it as half the time they had spent previously on testing similar systems.

3.2. Test Case Based Testing

Test case based testing is considered as traditional method of testing in which, test cases are defined and planned in conjunction with expected results with respect to requirements. According to IEEE standard 829-1998 [5], TCBT consists of documented tests which comprises of both actual values that are to be used as inputs and as well as pre anticipated outputs [5].

According to ISO/ICE 29119 [49] the structure of test case as follows:

1. Pre-condition for test execution.
2. Set of inputs i.e. values and actions.
3. Expected outputs and post conditions.
4. Compliance with specific requirements.
3.2.1. Working with Test Case Based Testing (ISO /ICE 29119)

However, we are using ISO/IEC 29119 software testing standard [49] to describe Test case Based testing approach. The reason for choosing this process is because it is a new standard which replaces all the other existing software testing standards like IEEE 1008 unit testing and IEEE 829 test documentation. The TCBT process is as follows:

- **Test planning**
  The main intention behind a test plan is to specify the scope, resource allocation, test approaches and to schedule the test activities [5]. Moreover, test plan also gives the detailed information about the product under test, its related risks and mitigations and features to be tested along with the resources that are responsible for accomplishing each testing task.

- **Test Monitoring and Control**
  The purpose of this section is to ensure that the test tasks are accomplished along with the defined test plan and other policies, which also influence the test process [5]. Test monitoring and control phase can be applied to overall project or to any individual test phase.

- **Test completion Criteria**
  This process assures the test assets availability to the respective stakeholders for using afterwards in conjunction with the documented results of testing [5]. Test assets also include test plans, test specification and test environment infrastructure.

- **Test Design and implementation**
  Test design and implementation phase comes prior to test execution phase. This phase is for developing test procedures which will be executed in the test execution phase [5]. Test cases execution requirements and test conditions are derived from the process. Every test case contains related information about the features to be tested, test approaches and P/F criteria (P: pass, F: fail)

- **Test Execution**
  In this phase the test cases that are developed from test design and execution phase are executed for recording the results and outcomes against the test case that are executed [5].

The successful outcomes of test execution process as follows:
- Test case execution
- Recording a test outcome
- Actual vs. expected results comparison
- The test results (pass/fail)
- Maintenance of test execution log

- **Test Incident reporting**
This phase is for reporting the issues and anomalies that are detected in the execution phase which requires further action [5].

Following is the process diagram of traditional Test case based testing.

![Test Case Based Testing Process](image)

*Figure # 3: TCBT process.*

3.2.2. Related Research

In one study [22], several experimental studies have been conducted for comparing different testing techniques. In one of his study, Juristo concluded that available knowledge is limited,
lack of formal foundation and somewhat conflicting [22]. In [25] Kamsties and Lott found that the amount of time taken to find a defect was completely dependent on subject. Basili and Selby rather, found that defect detection rate was not associated with tester experience also found that fault rate is dependent on the software under study [6].

One conclusion from existing studies depicts that more faults are noticed when combining individual testers than techniques [22]. Wood et al. found that more defects are identified in combined pairs than individuals though they use same techniques [45].

However, In industrial setting there are quite a few studies showing that the test case are rigorous, well documented and used [21]. On the other hand, Itkonen [19] reported that most of the time is spent for documenting and writing the test cases rather, in executing them. According to Andersson the use of structured approaches to VαV is sparse. Instead, the selection of test cases is very much based on the experience of the staff [4]. On the other hand, no one reported particular problems that can be traced back to the lack of structured methods specifically [4].
CHAPTER # 3

STATIC VALIDATION

STRENGTHS AND WEAKNESSES OF MBT AND TCBT APPROACHES

Industrial Interviews
- Purpose of interviews
- Selection of subjects
- Interview structure
- Feedback structure
- Results
- Analysis of results collected from Interviews
- Discussion and summary
- Validity threats
4. STRENGTHS AND WEAKNESSES OF MBT AND TCBT APPROACHES

4.1. Industrial Interviews

4.1.1. Purpose of Interviews
The purpose of interviews is to get knowledge about the usage, strengths and weaknesses related to both approaches (TCBT and MBT) in industry. The interviews will help us to answer relevant research questions i.e. RQ.1 and RQ.2. The interview conducted was semi-structured [13] in nature and the interview questions were designed to keep the discussion focused. The interviews were necessary because, we wanted to add industrial experiences in our research and then to compare those with the experiment results to validate everything. Also, we wanted to add personal experiences in our thesis, which cannot be acquired quantitatively.

4.1.2. Selection of Subjects
In order to make the research study authentic and reliable, it is very important to select the right subjects for the interviews. We selected candidates who have worked and are currently working with software testing, specifically MBT and TCBT. Experts working as software developers, software testers, test managers and consultants were used as interview subjects. The selection process was based on two major criteria, work experiences (number of years), their knowledge on either MBT or TCBT. We selected professionals who had more than 5 years of work experience in the field of software testing, specifically MBT or TCBT.

During the interview session with such professionals, it gave us broader understanding of the strengths and weaknesses that are faced by the companies while implementing either one of the approaches.

4.1.1. Interview Structure
The interview method was based on seven points as presented in [26]. They will now be covered in short below, together with details of how we articulated said points:

- **Thematizing**
  The interview questions were based on two reasons. The first reason was, to indentify the potential strengths and weaknesses while using the two approaches (i.e. MBT and TCBT). The second reason was, to look for a possible extension of MBT (if possible). The purpose was to understand how MBT and TCBT are practiced in industry and, furthermore, validates the strengths and weaknesses (preferably statistically) between the two approaches.

- **Designing**
  In the design, we indentified four study instruments on the basis of the literature review. The study instruments were defined as follows:
**Instruments**

The purpose of defining these study instruments one and two was to get knowledge on industry’s view on the strengths of MBT and TCBT as highlighted in literature. The reason for doing this was to validate the strengths mentioned by industrial professionals with the results of experiments. Furthermore, the questions derived from this study instrument were qualitative in nature and were focusing on strengths of MBT and TCBT in industrial practices.

Study instrument three and four was designed in order to get the feedback from practitioners on the weaknesses of MBT and TCBT. These instruments were used to validate the weaknesses mentioned by the industrial practices with the experiment results (in the next chapter). Interview questions derived from this study instrument were also qualitative in nature with a focus on weaknesses of MBT and TCBT. (See appendix 1).

**- Interviewing**

Based on the design, interview questions were identified, which were then used in the actual execution of interviews. We conducted six face to face interviews (three for MBT and three for TCBT) using a video conferencing software application “Skype”. An application “MP3-Skype recorder” was used to record the conference call for the interviews and to make sure that we did not miss anything. In addition a camera was used to take pictures of each step during the execution. The duration of each interview was approximately 40-60 minutes. Both interviewers were taking notes manually as well.

![Figure # 4: Interviewees (designations) and their experience (year).](image)

**- Transcribing**

In this phase, we omitted all irrelevant information and digitized the manual data; that was taken at the time of interviews. In this phase, the general discussion was separated from the key points in the interviews.
Analyzing and Verifying
After transcribing, the output was analyzed to generate results in the form of repertory grid [15]. Repertory grid was selected for the analysis and verification because this method has not often been used in the field of software engineering, and we wanted to test if this method could deliver the expected quality in results from the qualitative data.

There are four steps that are involved in order to make a conclusion on the basis of qualitative data. Following are the steps that were used for the analysis and generating conclusion.

- **Topic**

The topic of the interview was MBT Vs TCBT.

- **Elements Definition**

According to [15], the elements are defined as “the things or events that are abstracted by a construct and are seen as one of the formal aspects of a construct”. In analysis, we used a TCBT and MBT as elements.

- **Construct Definition**

Constructs are the terms that are used to make sense of the elements. In analysis, different constructs were formulated with the help of interviewees in order to present our findings in the interviews. Following are the constructs that were used for analysis.

  o Low test coverage VS High test coverage
  o Low detailed test cases VS Highly detailed test cases
  o Cost efficient VS Time taking process
  o Quality in requirement VS Quality of Testing design and planning.
  o Low requirements traceability VS high requirement traceability

- **Rating Scale**

A rating scale (1-5 Lickert scale) was used in order to position the elements between two extremes of the construct. We used “Web Grid-5” [50] cross plots to illustrate our results.

4.2. **Results and Analysis**

In this section, we analyzed and presented the results of the data that was collected during the interviews. The Strengths and challenges of the MBT and TCBT approaches are discussed in the following section.
4.2.1. Ratings for TCBT and MBT

In this sub-section, we have used 1-5 Lickert Scale to show the average rating of TCBT and MBT approach while considering the constructs that were formulated earlier (in section Interview Design – Analyzing and Verifying – Construct Definition).

Following are results of TCBT and MBT perceived from interview data analysis. The elements were positioned between two extremes of the Likert scales:

- **Test Coverage**

  The test coverage can be seen as strength of TCBT and MBT. According to the interview subjects, the coverage usually depends on the experience of the tester who is writing the test. They also mentioned that in companies the test cases are usually written by the experienced testers, if not, the test cases are approved by experienced testers. Also, in MBT, the test coverage is always higher than TCBT because the test cases are generated by considering the test coverage.

![Test Coverage Construct](image)

*Figure # 5: Test coverage construct.*

The above scale was generated on the basis of average score of TCBT and MBT while considering the test coverage. It shows that TCBT has fairly high test coverage (i.e. branch, path and statement coverage) but MBT has more coverage because of its zero-tolerance towards the test coverage.

- **Requirement Traceability**

  Requirement traceability is one of strengths of TCBT and MBT. According to the interview data, there are several ways to make the requirements traceable through the test cases. According to the interviewees, different applications can be used in order to link the test cases to the software requirements. Also, they mentioned that, there can be a separate column in the test case which shows that this test case belongs to ABC requirement. While in MBT, the traceability is done in a different way. According to the interview subjects; requirement traceability is a challenge in MBT and companies usually find it difficult to track the results back to the system requirements in the MBT approach. Recently, some major studies [1, 2, 10] have been done in order to find out a better way to make requirements traceable in MBT process.

According to the interview data analysis, following Likert chart shows the requirement traceability in TCBT and MBT.
Understandability of Test cases
According to the interview subjects, the understandability of the test cases depends on the experience of the tester who is writing the test cases. It is one of the challenges of TCBT because every tester writes the test cases according to his own knowledge of the system and business. It was also identified that, the more detail test cases have more understandability. In MBT, the automated test cases are not fully understandable by human beings. In MBT, it depends on which tool you are using. For example, Conformiq Qtronic and Microsoft’s Spec Explorer adds reasonable details on the test cases. So that, a human engineer can understand what will be the details and what to be tested.

The following Likert scale shows the rating of MBT and TCBT between two extremes of understandability construct. The scale was rated on the basis of analysis of interview data.

Cost and Time
Cost and time are one the most important attributes of an approach. From the interviews we came to know that it cannot be said that TCBT is more costly or MBT is more costly. It depends on different scenarios, what application is going to be tested, how it will be tested and under what conditions it is being tested. According to the interviewees, in the total cost of the project, there is 50-70% cost for quality assurance and if there is a defect in the release,
the cost is increased even more. It was perceived from the interviews that, MBT is known for decreasing the overall testing cost and it takes less time as compared to traditional manual testing approaches. From the interview data, we managed to rate MBT and TCBT on a Likert scale below:

![Cost and time construct](image)

*Figure # 8: Cost and time construct.*

- **Test Design and Planning**
  Test design and planning depends on the system requirements. According to the interview subjects, understanding the requirements is one of the most important tasks before you actually start test design and planning. If the test plans and designs are made without fully understanding the system requirements there is a high chance that re-work will be required.

Interview data analysis resulted in the following rating of the TCBT and MBT.

![Test Design and Planning construct](image)

*Figure # 9: Test Design and Planning construct.*

4.2.2. Results of TCBT and MBT
The rating of elements on constructs helped in generating the results of the findings in the interviews phase. The results of the interviews phase are mentioned below in the form of cross plot charts.
The above cross plot diagram shows that, TCBT has high coverage but the level of details of test cases are not that much. MBT has higher test coverage and more detailed test cases if compared to the traditional TCBT approach.

According to the interviewees, the coverage depends on how you are testing the application and to which detail you have understood the requirements. Therefore, the coverage depends on the quality of the test cases, which depends on the quality of the system requirements.

The above graph shows that, MBT has comparatively low requirement traceability as compared to TCBT. At the same time, MBT is more cost and time efficient as compared to TCBT.
Dynamic Validation
- Experiment
- Planning
- Experiment Design
- Experiment Execution
- Results and Analysis
5. Dynamic Validation

5.1. Experiment
The goal of this phase is to dynamically validate the findings of interview results. This part is documented just to ensure that, important factors are properly documented and defined before going to the actual execution.

5.1.1. Definition
The purpose of this experiment was to validate the results of interviews dynamically. The time consumed \( t \) to create the test cases, requirement traceability and test coverage (branch, statement and path) were calculated and compared to the data from interviews. The experiment was done with the students of software engineering and computer science disciplines. Another purpose of this experiment was to create a baseline for future work.

Before actual execution of the experiment all the experiments were educated on their respective methodology (i.e. MBT or TCBT). At the beginning of experiment, system requirements were given to the experiment subjects and proper explanation was provided by the authors. In addition, 30 minutes were given to experiment subjects for questioning and to eliminate the confusions they had about the requirements and the methodology.

Some activities of definition phase are mentioned below:

- **Goal Definition**
  The goal was to dynamically validate the results of the interviews and to find out which technique is actually better in providing better test cases. Which means that, time consumption \( t \) to create the test cases, requirement traceability and the test coverage (branch, statement and path) were calculated for both test approaches (MBT and TCBT). The aim was to compare the mentioned measures for both approaches and find out which is better and in providing better test cases.

- **Object of Study**
  Model Based Testing (MBT) and Test Case Based Testing (TCBT) approaches were used as object of study.

- **Purpose**
  The purpose of the experiment was to compare the values of time, cost, requirement traceability and test coverage for both approaches (MBT and TCBT) and, to draw the final conclusion on the basis of those values.

- **Perspective**
  The perspective of this experiment was to create a researcher’s conclusion on MBT and TCBT.

- **Context**
  The experiment was executed under the self-made academic setup.
5.2. Planning
In this part, selection of subjects, hypothesis formation, experimentation context and variable selection was identified and documented.

Following are some areas of planning phase:

- **Selection of Subjects**
The experiment subjects were selected on the basis of interest in software testing, background and experience in the domain of software testing and software-development (see appendix 2). While dividing the teams in MBT group and TCBT group, we tried to divide the experienced students evenly in order to fully control the experiment (for more details on the division of subjects in experiment see the experiment design section below).

Furthermore, the authors gave presentations to the groups on their respective approaches (i.e. either MBT or TCBT).

- **Experimentation Context**
The experiment was conducted in two sessions. One session was dedicated to the MBT group and the other session was dedicated to the TCBT group. Academic setup was maintained during both parts. All teams were provided knowledge of their respective approaches and were given time to ask questions related to the experiments and the approaches.

- **Variable Selection**
There are two types of variables in the controlled experiments [44], i.e. dependent and independent variables.

The dependent and independent variables for the experimentation are given in the table below:

<table>
<thead>
<tr>
<th>Type</th>
<th>Variables</th>
<th>Values</th>
<th>Scale Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent Variables</td>
<td>Testing approach</td>
<td>MBT</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCBT</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Tool Support for MBT</td>
<td>Model J-Unit</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Personal Experiences</td>
<td>Years, Months</td>
<td>Ordinal</td>
</tr>
<tr>
<td>Dependent Variables</td>
<td>Time (t)</td>
<td>{0, 1, 2, 3,..., n}, n ∈ N</td>
<td>Interval</td>
</tr>
<tr>
<td></td>
<td>cost (c)</td>
<td>{0, 1, 2, 3,..., n}, n ∈ N</td>
<td>Interval</td>
</tr>
<tr>
<td></td>
<td>Requirement traceability</td>
<td>{0, 1, 2, 3,..., n}, n ∈ N</td>
<td>Ordinal</td>
</tr>
<tr>
<td></td>
<td>Branch Coverage</td>
<td>{0, 1, 2, 3,..., n}, n ∈ N</td>
<td>Interval</td>
</tr>
<tr>
<td></td>
<td>Statement Coverage</td>
<td>{0, 1, 2, 3,..., n}, n ∈ N</td>
<td>Interval</td>
</tr>
<tr>
<td></td>
<td>Path Coverage</td>
<td>{0, 1, 2, 3,..., n}, n ∈ N</td>
<td>Interval</td>
</tr>
</tbody>
</table>

*Table #1: List of dependent and independent variables.*
- **Hypothesis formation**

One experiment was conducted in academia on two different subjects (see details in “Experiment design” section below). The results obtained after the study will be compared to the following hypotheses:

- H₀-qual: The comparison between MBT and TCBT will result no significant changes in terms of test coverage (branch and path), requirement traceability, cost and time.

- H₁: Comparing MBT and TCBT, the results from the experiment will reflect major changes in terms of coverage (state, branch, path and transition), requirement traceability, cost and time.

### 5.3. Experiment Design

We followed empirical guidelines from Juristo and Moreno [23]. Our experimental design is based on one-factor block design [23, 44]. The experiment was performed in academia, where the students were taken as our experimental subjects. The student subjects were divided into two groups based on their background, knowledge and experience and then further into six sub-groups (see the diagram below). The first group was named as TCBT group, and the second group was named as MBT group. In both groups there were three sub-groups.

![Experiment Design](image-url)

*Figure # 12: Experiment Design.*
The experiment comprises of one phase with two different sessions; one for TCBT and the other for MBT. The TCBT group was asked to design the test cases for the given requirements and in the second session, the MBT group was asked to create the FSM models (Implemented in Java) from the given set of requirements. The FSM models were then given as input to the Model J-Unit tool [11]. The model J-Unit tool was used to generate the test cases automatically.

Note that, systems requirements were kept same for both sessions of the experiment in order to control it. The rational was to measure test coverage (i.e. branch, statement and path), the time consumed by the both teams in generating test cases in terms of efficiency, requirements traceability in terms of effectiveness and cost.

The reason for using one phase was, to reduce learning effect of the subjects and to measure the true effect of experiment. One subject (group of three people) had to carry out the experiment only on time using the given methodology.

5.3.1. Terminologies

5.3.1.1. Experimental Units
The experimental units are the objects on which experiment is done. In this research, the experimental units used were requirement gathering and analysis phase and the design phase of software-development life cycle (SDLC).

5.3.1.2. Experimental Subjects
These are people who actually apply the experiment. In this experiment, there were two main groups. One group was trained on manual testing and the other group was trained on MBT. Both groups contained three sub-groups with three students. These totals of six sub-groups were our experimental units. These groups of students acted as a control group in our experiment.

5.3.1.3. Response Variables
Response variables are the quantitative outcomes of an experiment. In this research, the response variables were efficiency in terms of time consumed for a task, requirement traceability and cost in terms of productivity, cost effectiveness and test coverage (i.e. path, statement and branch).

5.3.1.4. Parameters
The characteristics, which will remain constant throughout the experimentation, are known as parameters. The parameters do not influence the result of the experiment. In this research, following terms were used as parameters:
- Properties of subjects
- Experiment environment
- Software Requirements (as the input to MBT and TCBT.)
- Implementation of MBT and TCBT.

5.3.1.5. Factors and Alternatives
The characteristics, which affect the response variables, are called factors. Factors can have several possible alternatives. In this research, the factor used was applied agile methodology for
test case development. It had two alternatives or treatments: Model Based Testing (MBT) and Test Case Based Testing (TCBT) process. The alternatives were applied according to the arranged sets of experimental subjects (the sub-groups).

5.3.1.6. Blocking Variables
There are undesired differences of variables which can affect different elements of the experiment. These variations are known as blocking variables or undesired variations. These variables cannot be made constant. The experimental subjects, their experiences, ability and time taken to complete a task were considered as blocking variables which can affect the response variables of the experiment.

5.3.1.7. Internal Replication
The repetition of the experiment in the experimental subjects is known as internal replication. It increases the reliability of the results of experimentation [24]. The experiment was replicated among six teams that means there were six internal replications for the experiment.

5.4. Experiment Execution
For experiment execution phase, preparations were made and training was provided to the subjects before the actual execution. While selecting the suitable candidates for experiment, background check on the subjects was done in order to group them in a team of three individuals. Questioning and answering session was also scheduled for removing confusions and clearing ambiguities. Subjects were asked to bring their own laptops for the experiment execution phase.

The experiment was divided in two sessions (one for MBT and the other for TCBT). Before beginning the actual execution, training was provided by the authors with a brief introduction of TCBT and MBT techniques. For Modeling the FSM models Eclipse™ IDE was installed and used on MBT group laptops. Test case template (see appendix 3) was provided to the TCBT group in order to document the test cases.

The experiment was conducted on 13th and 14th January 2012 in a group room at Blekinge Institute of Technology. The training session of MBT lasted 90 minutes and training on TCBT lasted 45 minutes. Time was recorded for each team to complete the tasks.

The system requirements of traffic light control system were used for both sessions of the experiment. There were total six system requirements. We used this system because it is considerably small and understandable by the student subjects.

The first session of the experiment was on MBT and it was conducted on 13th of January 2012. Models were generated by the MBT team on papers first and then discussed. Modeling on paper and discussion took approximately 1 hour. After thorough discussion and one model from each of the three MBT teams was selected for implementing in Eclipse IDE. The implementation was done in approximately 90 minutes. The implemented model was then given as an input to the Model J-Unit tool through which test coverage was measured and test cases were generated. Throughout the first session of the experiment time was recorded.
The second session of the experiment was on TCBT. It was conducted on 14th of January 2012. The requirements provided to the TCBT teams were same as that of provided to MBT teams. A test case template (see appendix 3) was provided to all three teams. The template was used to record the test cases. The test case development phase took approximately 120 minutes because, the TCBT team was working manually and they had to first design the flow charts of the application according to mutual discussion among each team and then they had to write the test cases accordingly. The flow charts were used to calculate the test coverage.

5.5. Results and Analysis

This section of the document provides the analysis and result of the experiment phase. Furthermore, this section highlights the conclusion and validity threats to dynamic validation.

The analysis of the data was done by descriptive statistics i.e. graphs were made in order to statistically analyze the experiment results. After that, hypothesis testing is also done in order to check that whether \( H_{0,\text{qual}} \) can be rejected and conclusion can be drawn from it [44].

5.5.1. Test Coverage

In test coverage; branch, statement and path coverage was calculated for each team for both approaches (MBT and TCBT). Following table contains the results for test coverage.

<table>
<thead>
<tr>
<th>MBT Teams</th>
<th>Model Based Testing</th>
<th>Test Coverage</th>
<th>Test Case Based Testing</th>
<th>TCBT Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>3</td>
<td>Branch Coverage</td>
<td>2</td>
<td>Team 1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Statement Coverage</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Path Coverage</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Team 2</td>
<td>3</td>
<td>Branch Coverage</td>
<td>3</td>
<td>Team 2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Statement Coverage</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Path Coverage</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Team 3</td>
<td>3</td>
<td>Branch Coverage</td>
<td>3</td>
<td>Team 3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Statement Coverage</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Path Coverage</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Table #2: Test Coverage in MBT and TCBT.

The above table shows the comparison of test coverage results of the test cases generated from both approaches (MBT and TCBT). The test cases written by TCBT Team – 1, showed that there are two test cases required to test all conditions which they had sorted out during the analysis and design phase, they also mentioned two test cases are required to cover all the statements (statement coverage) and total of four test cases were required to test all possible paths (path coverage). Similarly, from TCBT Team – 2 test data, the branch, path and statement coverage was 3, 3 and 1 respectively. From test data gathered from TCBT Team – 3, branch, path and statement coverage was 3, 4 and 1 respectively.

We managed to calculate the test coverage of the test data gathered from MBT experiment subjects. The test data from the three MBT teams resulted in branch, path and statement coverage of (3, 3, 2), (3, 4, 2) and (3, 3, 2) respectively.
### 5.5.2. Cost and time

The cost was calculated by giving each team “$x$” value as standard cost per minute. For calculation we have given $x$ a standard value which is 0.5 units / minute i.e. 30 units / hr. The cost was calculated with respect to time consumed by each team. Cost was calculated by:

$$\text{Cost (for a team)} = \text{Job time (in minutes)} \times \text{Wage (x)}$$

The results that were calculated are mentioned in the table below:

<table>
<thead>
<tr>
<th>MBT Teams</th>
<th>Model Based Testing</th>
<th>Time and Cost</th>
<th>Test Case Based Testing</th>
<th>TCBT Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>75 minutes Time</td>
<td>96 minutes</td>
<td></td>
<td>Team 1</td>
</tr>
<tr>
<td></td>
<td>37.5 units Cost</td>
<td>48 units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 2</td>
<td>70 minutes Time</td>
<td>115 minutes</td>
<td></td>
<td>Team 2</td>
</tr>
<tr>
<td></td>
<td>35 units Cost</td>
<td>57.5 units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 3</td>
<td>90 minutes Time</td>
<td>111 minutes</td>
<td></td>
<td>Team 3</td>
</tr>
<tr>
<td></td>
<td>45 units Cost</td>
<td>55.5 units</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table # 3: Time and Cost for MBT and TCBT teams.*

For the analysis, we developed graphs for both teams in order to show the differences between the time and cost for the teams of MBT and TCBT. The following graph shows the differences in time for MBT and TCBT teams. It shows that, time taken by MBT teams is less as compared to time taken by the TCBT team.

*Figure # 13: Time graph for MBT and TCBT teams.*

The following graph shows the difference of calculated cost between MBT and TCBT teams. It shows that MBT team is less costly as compared to TCBT team.
5.5.3. Requirement Traceability

In MBT random test cases were generated by the testing tool (Model J-Unit) and in TCBT, the test cases were written manually by the teams. The test cases were then matched against the system requirements in a metric for both approaches TCBT and MBT.

First three matrices contain the statistical results of requirement traceability for the TCBT teams.

<table>
<thead>
<tr>
<th>Requirements → Test Cases ↓</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC-001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table #4: TCBT Team -1: Requirement Traceability.*

<table>
<thead>
<tr>
<th>Requirements → Test Cases ↓</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC-001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC-005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table #5: TCBT Team -2: Requirement Traceability.*
In Model J-Unit, the requirement traceability is not covered. But requirement traceability in MBT can be done using Qtronic Conformic Tool or Microsoft Spec Explorer [34].

From the experiment results, it is hard to say that which technique is better in providing test cases. So in order to show some significant difference between the both MBT and TCBT, we have performed statistical test on the obtained results. As we have very small samples, performing a t-test is not appropriate here for this reason, we are using a non-parametric test called the Mann-Whitney U test. Basically non-parametric tests are used in order to overcome the fundamental assumptions of normality in parametric tests [51].

5.5.4. Mann-Whitney U test (Two Tailed Test)

The Mann-Whitney U test is used to test whether two independent samples of observations are drawn from identical or same distributions [51]. An advantage with this test there is no necessary to have same number of observations on the both samples [51].

- **Working**
  
  Suppose we have a sample of \(n_1\) number of observations in group A (i.e. from one population) and sample of \(n_2\) number of observations in another group B.

  The Mann – Whitney U test is based on the criteria of comparing the every observation of first group (group A) with the every observation in the other group (group B). The total number of pair-wise comparisons is \(n_1 \times n_2\).

  **Set the null hypothesis H0:** There is no significant difference between the two groups  
  **Set the alternative hypothesis H1:** There is significant difference between the two groups

  - **Procedure to carry out**
    1. Note down all the observations of group A and group B.
    2. Assign ranks to the each observation according to its magnitude.
    3. If at all, we have same magnitude for multiple observations then we average the ranks and then ranked to observations.
    4. Now add up the ranks separately, for group A and group B.
    5. Calculate the U value.

![Table #6: TCBT Team – 3: Requirement Traceability.](image)
1. \( U (A): n_1n_2 + 0.5(n_1)(n_1+1) - R_1 \)
2. \( U (B): n_1n_2 + 0.5(n_2)(n_2+1) - R_2 \)

\( n_1 \) and \( n_2 \) are size of the observations and \( R_1 \) and \( R_2 \) are sum of the ranks of observations of A and B respectively.

6. Calculate \( U \). \( U \) is the smaller of the two values \( U (A) \) and \( U (B) \).
Use a table of critical \( U \) value for the Mann-Whitney \( U \) test. If the obtained \( U \) value is less than the critical \( U \) value of the table then we reject the null hypothesis.

*Note:* The Mann-Whitney \( U \) test follows a \( Z \) distribution when the sample size is greater than 20. We reject the null hypothesis if and only if the obtained \( Z \) value is less than the negative critical \( Z \) value or greater than the positive critical \( Z \) value.

The \( Z \) value is given as:

\[
Z = U - \left( \frac{n_1n_2}{2} \right) \sqrt{\frac{n_1n_2(n_1+n_2+1)}{12}}
\]

**Applying Mann-Whitney \( U \) test for the experiment results**
In our case, we have two set of observations one is for MBT and one is for TCBT. See table 7.

- **Observations**

<table>
<thead>
<tr>
<th>S. No</th>
<th>Observations of MBT</th>
<th>Observations of TCBT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

*Table # 7: Mann-Whitney Observation Table.*

**Null Hypothesis:** There is no difference between the observations taken from MBT and TCBT.

**Alternative Hypothesis:** there is significant difference between the observations taken from MBT and TCBT.

Let \( n_1 \) be the size of first group observations (MBT) and \( n_2 \) is the size of the second group observations (TCBT).

\( n_1 = 9 \) and \( n_2 = 9 \)

- **Ranks of the observations**

Based on the magnitude of the observations the ranks are as follows.
<table>
<thead>
<tr>
<th>S. No</th>
<th>Ranks of MBT</th>
<th>Ranks of TCBT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>11.5</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>11.5</td>
<td>11.5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>11.5</td>
</tr>
<tr>
<td>7</td>
<td>11.5</td>
<td>11.5</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>11.5</td>
<td>17</td>
</tr>
</tbody>
</table>

*Table # 8: Mann-Whitney Ranks Table.*

\[ R_1 = 89.5, \quad R_2 = 81.5 \]

Where, \( R_1 \) and \( R_2 \) is the sum of total ranks of MBT and TCBT.

Now we have to calculate the U values for MBT and TCBT. The \( U (A) \) is taken as the U value for MBT and \( U (B) \) is taken as U value for TCBT.

\[
\begin{align*}
U (A) &= 36.5 \\
U (B) &= 44.5
\end{align*}
\]

By using the table, we have to find out the critical value for the \( U \) static at 5% significant level. i.e. \( U = 17 \)

According to the rule we have to reject the null hypothesis if the smallest value of \( U (A) \) or \( U (B) \) is below the actual value of \( U \) (i.e. 17). In our case, \( U (A) \) got the least value and which is greater than the critical value so; we have to accept the null hypothesis. There is no significant difference between the MBT and TCBT in generating test cases.
CHAPTER # 5

EPILOGUE

Epilogue
- Validity threats
- Conclusion
- Answers of research questions
6. **Epilogue**

6.1. **Validity threats**

In this section, different validity threats related to experiments and interviews are addressed. We followed Creswell [12] to define various validity threats in our study.

6.1.1. Validity Threats for Interviews

- **Internal Validity**
  
The internal validity threats are used in order to verify that, if irrelevant data is omitted from the gathered data from interviews [12]. Because, the questionnaire (*Appendix I*) was designed for a semi-structured interview, a lot of discussion was made with the interviewees on different aspects of both approaches; we omitted a lot of irrelevant data in order to reduce this threat. So that, we could use only the relevant data for analysis and results.

- **External Validity**
  
The external validity threat refers to getting a different outcome in a different setting [12]. There was a threat that candidates will show lack of trust and give faulty data. We selected the candidates who were interested in the topic in order to minimize this threat.

- **Construct Validity**
  
The construct validity threat refers to associate the definition or measurement with the exact variable [12] in the static validation, there was a threat that misunderstanding and wrong interpreting could be done on any statement. We minimized this threat by doing a double check on the data gathered from the interviewees. We compared the data with the literature in order to minimize the risk of wrong interpretation.

- **Conclusion Validity**
  
This validity threat explains the issues that lead to faulty results. It is required because it insures that the results derived from the research data leads to right conclusion and it should be reliable [12].

The only threat associated with conclusion validation was that if we misunderstood something in the interview. We double-checked our interpretations with the literature and since we conducted interviews on Skype, we used a recording application (MP3-Skype call recorder) in order to make sure that we do not miss anything and we understand everything.

As the nature of the interviews was industrial, we found limited resources, which could be used as interview subjects. That is why; the interview data has low statistical power. This can be increased by having more interview subjects in future.

6.1.2. Validity Threats for Experiment

- **Internal Validity**
  
There was a threat that, all teams will be performing their respective testing approaches to generate test cases for the first time. This could affect the results of the experiment. In order to minimize this threat, the teams were made by keeping at least one experienced...
person in each team. Also, we provided training sessions in the beginning of the experiment and proper questioning and answering time was given to solve any confusion related to the test approaches.

- **External Validity**
  The experiment was done with the students of Masters Programs in computer science, software engineering and software quality. It was not done in industry; only one application (traffic system) was used in order to validate the results of the interviews. This is a major threat to external validity of this study.

- **Construct Validity**
  We used a traffic light system with limited functionality in order to test the two approaches (i.e. MBT and TCBT). It is a small application with 7 functionalities in total; the results may vary on the complexity and size of the application. We could not make the experiment with a larger and more complex application because of limitation of resources and availability of experiment subjects. This is a construct validity threat for this study.

- **Conclusion Validity**
  The main threat associated with conclusion validity was the experiment data collected from the experiment subjects. We provided templates for test case templates and training for TCBT and MBT through which we minimized the risk of getting fake and incorrect data.

### 6.2. Conclusion

In this study, we executed one experiment with two sessions in academia. The purpose was to validate the received data from the interviews in the industry. We wanted to highlight the potential difference in requirement traceability, time, cost and test coverage (i.e. branch, path and statement). All of these terms were measured and analyzed based on the experimental data and interviews. The experiment and interviews data showed that, when it comes on requirement traceability in MBT generated test cases it is very hard to make the test cases traceable to the requirements. Requirement traceability varies from different approaches to use MBT i.e. we used Model J-Unit tool for generated automated test cases in MBT but that can be done by using Microsoft Spec Explorer or Conformiq Qtronic. These tools have the functionality of requirement traceability through which test cases can be linked back to requirements. Furthermore, time and cost consumed by all the teams in MBT and TCBT, two out of three pairs of teams have shown that, MBT consumes less time and thus it is cost effective and efficient as compared to TCBT. Moreover, MBT showed better test coverage in experiment as compared to TCBT.

### 6.3. Answers of the research Questions

**RQ.1:** What are the strengths and weaknesses faced in the industries while using Model Based Testing (MBT)?

**Answer:** Section 4.2.2 describes the strengths and weaknesses of MBT. The answer of this question was obtained from the interviews. The strengths of MBT associated with the
comparison of TCBT in this study are time efficiency, cost, and coverage. But the weakness of MBT is requirement traceability. There is no method to make the test cases traceable. There are tools, which can be used to generate test cases (i.e. Qtronic and Spec Explorer). These tools have the functionality which for requirement traceability.

**RQ.2:** What are the strengths and weaknesses faced in the industries while implementing Test Case Based Testing (TCBT)?

**Answer:** Section 4.2.2 describes the strengths of TCBT. The answer of this research question was concluded on the basis of interviews. The strength of TCBT (if compared with MBT) is requirement traceability in which TCBT is better as compared to MBT. Requirements can be made traceable either by manually writing the requirement ID in the test cases or by linking the test cases to requirements using some tool.

**RQ.3:** Which technique is better in terms of providing better test cases for software applications?

**Answer:** This research question is answered on the basis of comparison between the experiment results and results from the interview. Section 5.5 describes the results of this question in detail. When it comes to efficiency and testing the behavior of applications in different environments, MBT is the better than traditional TCBT approach. MBT is better in terms of less time, low cost, better test coverage (i.e. branch, path and statement). According to the experiment results in section 5.5; there is not much difference between test coverage, time and cost of traditional TCBT and MBT. Which means both techniques can be used interchangeably depending on what is being tested and in which environment it’s being tested.
REFERENCES


doi: 10.1109/ESEM.2007.56


doi: 10.1109/ESEM.2009.5314240


doi: 10.1109/ISSRE.1994.341373


[34] Pasupulati, B.V. 2009. TRACEABILITY IN MODEL BASED TESTING. (2009).

c#252;nel, M. Baumgartner, B. Sostawa, R. Z
APPENDIX

1. QUESTIONNAIRE

1.1. Questions related to challenges of Test case based testing (TCBT).
Q1. Do you think designing test cases is time consuming and it affects the cost?
Q2. While doing regression testing, do you think TCBT is suitable?
Q3. In which scenarios do you think TCBT is an exhaustive process?
Q4. Do you follow any specific standard for documenting the test cases in TCBT? How often it is necessary to revise the test cases?
Q5. Do you think that TCBT is the approach to follow when focusing on requirement fulfillment?
Q6. Do you think test cases are not enough and you need to test the system through your own knowledge? (If yes, please explain in what scenarios do you think like that?).
Q7. Do you think TCBT is the best approach to follow while considering stability of the software application?
Q8. Please list down any potential challenges that you have faced while working with TCBT.

1.2. Questions related to Strengths of TCBT.
Q1. How do you trace the requirements through test cases in TCBT? Please explain.
Q2. In what scenarios do you think TCBT is a good approach for having high test coverage?
Q3. Do you think requirements can be better traced, if TCBT approach is used?
Q4. Do you think TCBT approach is good for calculating test coverage?
Q5. Do you think that TCBT improves the overall quality of the test process?
Q6. List down the benefits achieved, while designing and planning the testing process.
Q7. Do you think that TCBT is an effective approach in formulating and guiding the testing tasks?
Q8. Please list down any potential benefits that are related to TCBT. (*In your experience*).

1.3. Questions related to challenges of MBT
Q1. Have you faced any challenges related to any phase in Model Based Testing? (If yes, then please explain about it and explain how you cope with it).
Q2. Have you ever experienced any test coverage issues while performing Model Based Testing? (If yes, please explain about it and how do you cope with it).
Q3. In which scenarios do you think that MBT is not a suitable technique?
Q4. How do you prioritize the tests in MBT?
Q5. Please define quality in context of MBT? How do you measure it?
Q6. Have you ever encountered any requirements traceability challenges while doing MBT? Please explain.
Q7. In which scenarios do you think MBT testing is a time taking process and is not suitable? Please explain.

53
Q8. In which scenarios do you think MBT is not cost effective and other testing techniques are better? Please explain.
Q9. Please list down any challenges related to MBT? And how this approach can be improved?

1.4. Questions related to Strengths of MBT
Q1. In which scenarios do you think MBT is the best technique to use?
Q2. In which scenarios do you think MBT can make significant difference in cost and time?
Q3. Do you think MBT is effective in defect detection? In what scenarios do think MBT is better than other techniques?
Q4. What benefits have you observed after implementation of MBT at your organization?
Q5. To what extent MBT guarantees software quality and stability of the software application? Please explain.
Q6. Do you think MBT is effective in detecting design faults? Please explain.
Q7. To what extent do you think that MBT is “the approach” to perform the functional testing and to find out the behavior of the application in different scenarios?
Q8. Please list down any potential benefits that you might have noticed while working with MBT.

2. BACKGROUND FORM

<table>
<thead>
<tr>
<th>Background Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please encircle the RIGHT option. Attempt question relevant to your testing type.</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Department</td>
</tr>
<tr>
<td>Have you gain any knowledge related to software testing?</td>
</tr>
<tr>
<td>- No</td>
</tr>
<tr>
<td>How would you evaluate your knowledge on software testing?</td>
</tr>
<tr>
<td>- Novice</td>
</tr>
<tr>
<td>- Moderate</td>
</tr>
<tr>
<td>- Skilled</td>
</tr>
<tr>
<td>- Expert</td>
</tr>
<tr>
<td>How long you have been working with software testing?</td>
</tr>
<tr>
<td>- 2-3 years</td>
</tr>
<tr>
<td>- Less than 3 years</td>
</tr>
<tr>
<td>Select your testing type</td>
</tr>
<tr>
<td>- Model Based Testing</td>
</tr>
<tr>
<td>Do you have any knowledge about the selected testing type?</td>
</tr>
<tr>
<td>- No</td>
</tr>
<tr>
<td>How long you have been associated with the selected testing type?</td>
</tr>
</tbody>
</table>

Table # 7: Background Information Table.
### 3. **Test Case Template**

<table>
<thead>
<tr>
<th>Functionality</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case ID</td>
<td></td>
</tr>
<tr>
<td>Team</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Requirement ID</th>
<th>Test Step Description</th>
<th>Input Values</th>
<th>Expected Results</th>
<th>Actual Results</th>
<th>Status(P/F)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table # 8: Test Case Template for TCBT.*