

# Exploring Better Teaching Methods by Surveying from Computer Science Students (Case Study for Software Engineering)

Khin Shin Thant, Thet Thet Aung, Hlaing Htake Khaung Tin

*Faculty of Information Science, University of Computer Studies, Hinthada, Myanmar*

**Abstract:** Software engineer is an engineering discipline. It applies to all aspects of software development from the start of the system. Software Engineering is a field of computer science that involves the development and development of computer software and software applications. Computer software is made up of programs that include computer equipment and operating systems. Software testing is a fundamental function of a software engineer. The main difficult of testing is how the appropriate things to check out the software system do. When software testing is taught in the classroom, student interested upon these subject is centered. This paper is learning better teaching techniques by taking surveys from computer science students (case study for software engineering subject).As the student assessment 10%, assignment 10%, quiz 10%, discussion 10% and tutorial 10% and 60% are examined as three hours exam. Especially, concerned with software testing and the faculty determine which part of the assessment is done more according to average mark of all student in each topic.

**Keywords:** Exploring Better Method, Computer Science Students, Surveying, Verification and Validation, Software Testing, Rapid Software Testing.

## I. INTRODUCTION

Rapid Software Testing (RST) provides resources and resources. Provides resources and resources. Experimental design for the world for information and time. This is a mental and professional one. It focuses on making the test work faster and more complete while completing the work of the exam. Software Testing, known as debugging the software, is to find the bugs: errors or other defects in software products. It can be defined as the process of verifying and validating software the actual outcome with the expected outcome. It is a technique to evaluate the capability of a product or program and to test reliability, usability, integrity, security, capability, efficiency, portability, maintainability and compatibility by using testing techniques.

There are two distinct goals. They say that the software is compatible with its requirements and that the software's behavior is incorrect. Developer and client to identify inappropriate or inappropriate errors in the software. Software testing goals (output) are verification and validation, priority coverage, balanced, traceable and deterministic. As the strategies, there are several software testing strategies. They are unit testing, integration testing, system testing and

acceptance testing, etc. As the students, these above facts are taught this academic year.

## II. RELATED WORK

In this section, the various works of software testing are discussed. It is a software application that searches for errors or errors. This document is about software testing, describe the objectives and methodology of the software exam and provide student results.

Software testing is often less formal process. To be testing effective and efficient, members in the team should be familiar with basic software testing goals, techniques and concepts. Software and computer technology are mainly computer science or software engineers. To be sure, a student will prepare different types of degree-level software in this field.

The computer science and software engineer are two different courses of study, each having a similar but important degree. Computer science students learn how to store data on computers and other computer devices. Computer science students study software development and writing. How to handle Databases How to handle databases securely Websites: Learn about applications and program computers.

Software engineering graduates will prepare for careers in software and applications development for large and small businesses. Students who specialize in software engineering specialize in games and entertainment systems. They may also choose to pursue a master's degree in computer science or computer engineering to expand their ability to work with computers at the computer science or hardware design level. [10]

This paper is organized as follows: Section II discusses related work for this research paper. Section III shows the three main topics of software engineering subject for third year computer science students, Section IV identifies the methodology and Section V describes the results and discussion of the paper and Section VI shows the conclusion of the research findings

## III. THE THREE MAIN TOPICS OF SOFTWARE ENGINEERING SUBJECT

In this Academic year first semester there are three main topics in software engineering subject are taught to the

students. They are Rapid Software Development, Verification and Validation and Software Testing. The details of each topic and their weaknesses are further elaborated.

### 3.1. Rapid Software Development

Rapid software development processes are designed to produce useful software quickly. Generally, they are iterative processes. They share some fundamental characteristics although there are many approaches to rapid software development.

- (a) Advantages: Accelerated delivery of customer services and user engagement with the system.
- (b) Disadvantages: Management problems, contractual problems, validation problems and maintenance problems.

In this section, agile method, extreme programming, rapid application development and software prototyping must be learned by the students.

### 3.2. Verification and Validation

Verification and validation are not the same. By verification, our product is building properly. Confirmation: Are we building the right product? Verification and approval is done at every stage of the software process. V&V begins by reviewing needs. Make a review and finally code for the product inspection.

There are two complementary approaches to system monitoring and analysis in the V&V process. They are both software testing or peer review and software testing. This section will provide project approval and approval. Students must learn automatic and silent evaluation and confirmation.

### 3.3. Software Testing

In software testing, there are two fundamental testing activities: component testing- testing the parts of the system and system testing- testing the system as a whole. The component testing's goal is to discover defects by testing individual program component.

The system testing is to integrate components to form subsystem or complete system and focus on establishing that the system meets its functional and non-functional requirements and not to behave in unexpected ways. It is the process of finding errors while executing a program to get a zero defect software. It leads to minimize the error and cut down software costs. Testing can be done automatically or manually and automation software testing is better than other.

#### 3.3.1. System Testing

System testing integrates two or more components, implements system functions or features and then tests those integrated system. It intends testing the end-to-end quality of the entire system and is based on the functional requirement specification and non-functional quality attributes (security, reliability and maintainability). Its purpose is execute the

compliance of the system concerned with the specified requirements.

In system testing, there are two distinct types for complex system: integration testing and release testing.

#### 3.3.1.1. Integration Testing

The combination test builds a system from its component and tests the resulting system for the problems caused by the component connection. The experiment is to build a bigger structure and to really work these parts together. Called correctly, it checks that their data is being moved through their interfaces in a timely manner. This is done immediately after the test has been completed and the test begins immediately.

It can be divided into two parts. Top Down integration and Bottom-Up integration. The disadvantage is finding bugs in this experiment.

#### 3.3.1.2. Release Testing

Release testing, a black-box testing process is known as the functional testing. It is a testing process to release or distribute to the customers. This testing's goal is to make the increasing the supplier's confidence and wants as the system meets its requirements.

#### 3.3.2. Unit Testing

A unit test called a unit test is a test of the components in the system. This is a bug detection process, and its purpose is to identify errors in those components. This is a test developer performing at the lowest or bottom level. The test is the smallest part of each piece or code, which is the unit, checks the parts, etc. At this stage, the individual is acting or acting on the objects or methods in a particular object. Many object classes contain components that integrate attributes and methods. The function of this testing is the output of one module becomes the input of another module.

#### 3.3.3. Structural Testing

This testing is a type of test that tests the structure of a code. It is also known as White Box Testing or Glass Box Testing. This type of test requires knowledge. This is usually done by developers. It is more concerned with how the system works than the performance of the system. It provides more coverage for tests. White-box testing is an approach to test case design by using the knowledge of the software structure and implementation.

#### 3.3.4. Acceptance Testing

This testing, known as the user acceptance testing, is the testing where the user is not very much interested in internal working /coding of the system, but evaluates overall of the system and compares it with the functions requirements they specify. Its purpose is to give the confidence that the system is working as expected behavior as required. It is to determine whether or not it is the most important test before giving the

system a final or final handover to the end user. Methods or methods for testing the software: black field inspection and white tile testing.

3.3.4.1 Black Box Testing

Black box testing is the one in which internal specifics and working don't need to know or accessible to its customer. It supports specifications and output needs. Its purpose is to identify the requirements of the system customer needs. It checks the functionally without any knowledge of the internal implementation. In this stage, the testers only have an understanding of what the software is supposed to do and not how does it. This is done based on customers' aspect. The tester only knows the set of inputs and specific outputs.

- (a) Advantages: Testers don't need to know internal working with the software program. Testing is perform based on user's point of view. Programmer and tester are not associated or independent.
- (b) Disadvantages: Test cases are difficult to design without fair stipulation. As the programmer, the repetition of tests have already been done.

3.3.4.2 White Box Testing

This testing is acutely cost effective in detecting and resolving problems. It is an approach to find error only if the tester has complete data. This isn't used much to debug in large system and networks. This checks the internal structure of the program as the programming skills and the domestic context of the system are used to design test cases. The tester appoints inputs to apply the path through the code and find the appropriate outputs.

- (a) Advantages: It exposes an error hidden in code. It eliminates coding extra line. The programmer gives reasons for implementation.
- (b) Disadvantages: The testers need to know knowledge of internal structure. He must have experience to perform this procedure. Many paths will remain untested.

IV. METHODOLOGY AND ANALYTICAL FRAMEWORK

This research survey was designed to collect on computer students from third year computer science class in academic year 2018-2019 at the University of Computer Studies, Hinthada, Myanmar.

This survey results are based on three sections of the software engineering learning outcomes. In this academic year, as the faculty, the student assessment is classified into five parts: paper exam 60%, assignment 10%, quiz 10%, discussion 10% and tutorial 10%. The student assessment in the four parts according to each topic are described as follow. Their average mark for each topic is found out by using mean equation.

$$Mean = \frac{x_1 + x_2 + x_3 + \dots + x_n}{N} \quad \text{----- (1)}$$

In equation 1,  $x_1$  is the roll no 1,  $x_2$  is the roll no 2 and  $x_n$  is the roll no 93 or the last one. N is the total number of students.

As the subject, three main topics are taught to the students by the faculty. There are 93 students (52 girls and 41 boys) in the classroom. Table 1 shows overview of survey participants.

Table 1: An overview of survey participants

	Participants (n)	Percentage (%)
Male	52	55.9 %
Female	41	44.1%
Total	93	100 %

V. RESULTS AND DISCUSSION

In this paper, as the result, average marks in assignment of all students are full marks in topic 1(Rapid Software Development), 9.0 in topic 2(Verification and Validation) and 7.8 in topic 3 (Software Testing). Average marks in quiz of all students are full marks in topic 1, 9.5 in topic 2 and 9.0 in topic 3. Average marks in discussion of all students are 8.0 in topic 1, 9.0 in topic 2 and 9.0 in topic 3. Average marks in tutorial of all students are full marks in topic 1, 9.0 in topic 2 and 7.9 in topic 3.

Table 6 shows the average number of assignment, quiz, discussion and tutorial for topic 1, topic 2 and topic 3.

Table 6: Average number of Assignment, Quiz, Discussion and Tutorial of survey participants

	Topic 1	Topic 2	Topic 3
Average number of Assignment mark (%)	10	9.0	7.8
Average number Quiz mark (%)	10	9.5	9.0
Average number of Discussion mark (%)	8.0	9.0	9.0
Average number of Tutorial mark (%)	10	9.0	7.9

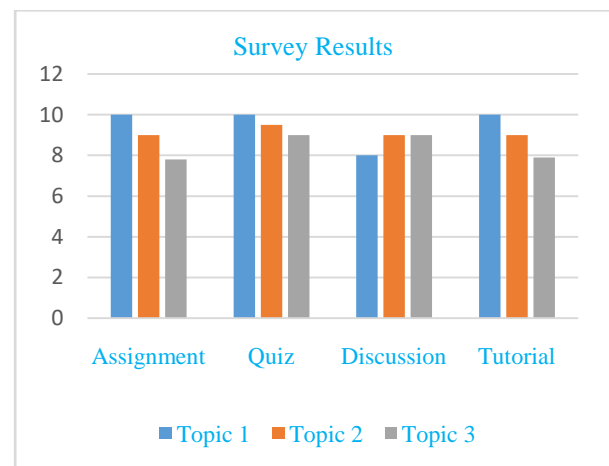


Figure 1: Evaluation of the Survey Results

### 5.1 Analysis of Assignment

In table 2, as assignment, for the topic 1, all students' mark are calculated by equation. The total mark is 930. So the average mark of each student is 10.0. The average marks of each student are 9.0 and 7.8 in topic 2 and 3.

Table 2: An overview of survey participants

	Topic 1	Topic 2	Topic 3
Total	930	837	728
Mean	10.0	9.0	7.8

### 5.2 Analysis of Quiz

In table 3, as quiz, for the topic 1, all students' mark are calculated by equation. The total mark is 930. So the average mark of each student is 10.0. The average marks of each student are 9.5 and 9.0 in topic 2 and 3.

Table 3: An overview of survey participants

	Topic 1	Topic 2	Topic 3
Total	930	884	838
Mean	10.0	9.5	9.0

### 5.3 Analysis of Discussion

In table 4, as discussion, for the topic 1, all students' mark are calculated by equation. The total mark is 744. So the average mark of each student is 8.0. The average marks of each student are 9.0 and 9.0 in topic 2 and 3.

Table 4: An overview of survey participants

	Topic 1	Topic 2	Topic 3
Total	744	838	837
Mean	8.0	9.0	9.0

### 5.4 Analysis of Tutorial

In table 5, as tutorial, for the topic 1, all students' mark are calculated by equation. The total mark is 930. So the average mark of each student is 10.0. The average marks of each student are 9.0 and 7.9 in topic 2 and 3.

Table 5: An overview of survey participants

	Topic 1	Topic 2	Topic 3
Total	930	838	736
Mean	10.0	9.0	7.9

## VI. CONCLUSION

In this paper, Rapid Software Development, Verification and Validation and Software Testing are taught by the faculty in this academic year. To conclude this survey, it is supposed Assignment, quiz and tutorial are fair with average marks in topic 1 but discussion is needed to do more time than present. Secondly, the average marks for topic 2 almost are good. As the faculty, for student assessments are usually done for next academic year. Thirdly, the average mark for topic 3, assignment and tutorial marks are less than quiz and discussion. So it is supposed that the faculty should give more time than usual for assignment and tutorial to the students. As the future plan, do girls learn more interesting than boys in which topic or not that can be made survey. These findings will help to improve the teaching methodology for software engineering subject in the coming academic year.

## REFERENCES

- [1]. D.Bhatt, "A Survey of Effective and Efficient Software Testing Technique and Analysis", (July 2017), IRE JOURNALS, Volume1 Issue 1.
- [2]. Rajendra Kumar, "Software Testing Techniques and Strategies", IJEAST, Vol.2, Issue 2, ISSN No. 2455-2143, Pages 113-117, 2016.
- [3]. Nahid,SusmitaKar, "Review paper on Various Software Testing Techniques and Strategies", GJCST, Vol.19, Issue 2, 2019.
- [4]. HimanshiBabbar, "Software Testing: Techniques and Test Cases", IJRCAR, ISSN, 2320-7345, Vol.5, Issue.3, page: 44-53, March 2017.
- [5]. Sommerville, Software Engineering, 8 edition.
- [6]. Joseph S.Valacich, Joey F.George, Jeffrey A.Hoffer, "Essentials of Systems Analysis and Design".
- [7]. S.McConnell, "Software Project Survival Guide", 1998, Microsoft press.
- [8]. B.Boehm, et al., "Software Cost Estimation with COCOMO II", Prentice Hall, 2000.
- [9]. D.J.Reifer, "Software Management", 6<sup>th</sup> Edition.
- [10]. <https://www.onlineengineeringprograms.com/faq/cs-vs-software-engineering>