OPEN ACCESS

Middle East Research Journal of Engineering and Technology
ISSN: 2789-7737 (Print) & ISSN: 2958-2059 (Online)
Frequency: Bi-Monthly
DOI: 10.36348/merjet.2023.v03i03.001

KUWAIT SCHOLARS PUBLISHER
Website: http://www.kspublisher.com/
Email: office@kspublisher.com

Mapping the Course Objectives and Program Learning Outcomes of Software Engineering Course by Bloom's Revised Taxonomy

Khin Shin Thant[1], Myat Mon Khaing[1], Hlaing Htake Khaung Tin[2*]
[1]Faculty of Information Science, University of Computer Studies, Hinthada
[2]University of Information Technology, Yangon, Myanmar

Abstract: The outcomes of software engineering can have a significant impact on the success of a software project and the satisfaction of its stakeholders. The results of the study provide a clear understanding of how success can be achieved in a program. Bloom's Revised Taxonomy is a framework used to classify learning outcomes in terms of different levels of complexity and cognitive ability. It can be used to assess the active learning outcomes of a software engineering course. Bloom's Revised Taxonomy can be applied to assess the active learning outcomes of a software engineering course. Assessing the active learning outcomes of a software engineering course using Bloom's Revised Taxonomy, educators can ensure that students are gaining a deep understanding of the material and are able to apply what they have learned in new and innovative ways. The main goal of the survey was to enable computer science students to use real-world software to help them make better decisions. Other objectives of this research include: (1) students having good communication with teachers gives them the opportunity to feel motivated and involved in the software engineering learning process and (2) Improve teaching and learning methods for Software Engineering. This research design is for undergraduate computer science students.

Keywords: Mapping, Course Objectives, Learning Outcomes, Software Engineering, Bloom's Revised Taxonomy.

Research Paper

***Corresponding Author:**
Hlaing Htake Khaung Tin
University of Information Technology, Yangon, Myanmar

How to cite this paper:
Khin Shin Thant *et al* (2023). Mapping the Course Objectives and Program Learning Outcomes of Software Engineering Course by Bloom's Revised Taxonomy. *Middle East Res J. Eng. Technol,* *3*(3): 33-38.

Article History:
| Submit: 17.04.2023 |
| Accepted: 25.05.2023 |
| Published: 29.05.2023 |

1. INTRODUCTION

Students' learning outcomes will find full overview of the information systems used by the business environment in the age of information technology. Assessing the outcomes of a program is an essential part of the educational process, as it allows educators and administrators to evaluate the effectiveness of the program and make improvements where necessary. There are several steps involved in assessing program outcomes: develop clear and measurable learning objectives, select appropriate assessment methods, collect and analyze data, interpret and communicate results, use results to improve the program, repeat the assessment cycle.

Bloom's revised taxonomy is an updated version of the original bloom's taxonomy, which was created by educational psychologist Benjamin Bloom in 1956. The revised version was published in 2001 by a group of educators led by Lorin Anderson, and it reflects the changes in education theory and practice that have occurred since the original taxonomy was developed.

The revised taxonomy maintains the same basic structure as the original, with six hierarchical levels of cognitive complexity: Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating. However, there are some key differences between the two versions. The revised taxonomy provides a more flexible and nuanced framework for understanding and assessing cognitive complexity in learning. It acknowledges that learning is a dynamic and multifaceted process, and encourages educators to focus on developing a range of cognitive skills and strategies rather than simply aiming for higher levels of abstraction or complexity.

Assessing program outcomes is an ongoing process that involves careful planning, data collection, analysis and communication. The results of assessment should be used to improve the program and ensure that students are achieving the intended learning outcomes.

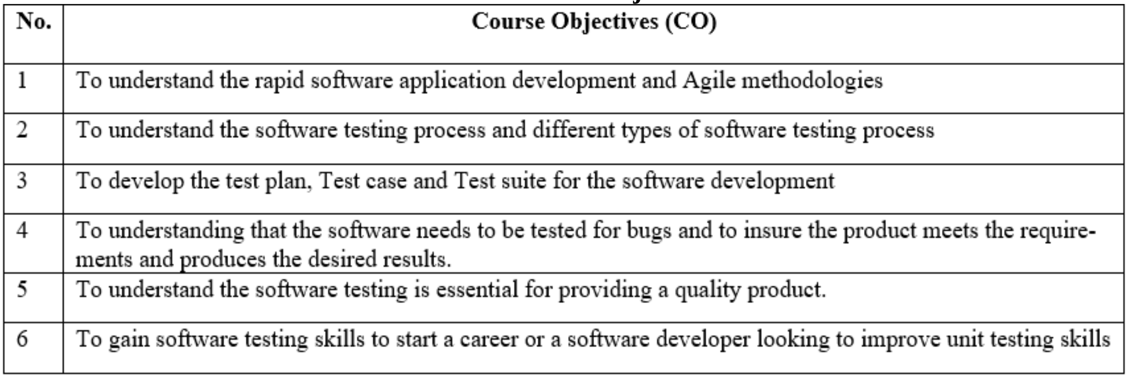
2. THE ROLE OF SOFTWARE ENGINEERING

This module is intended to provide in-depth coverage of software testing further to develop the introductory material covered in Software Engineering. The goals of the course are too rapid software application development and introduce the students with the skill to select and apply a testing strategy and testing techniques that are appropriate to a particular software system or component. In addition, the student will become a capable user of test tools; will be able to assess the effectiveness of their testing activity; and will be able provide evidence to justify their evaluation. The course will be supported by two practical exercises involving the development of appropriate tests and the application of a range of testing tools.

2.1. Course Objective

Course objectives (CO) are specific and measurable statements that describe what students should be able to do by the end of a course. They provide a roadmap for the course and guide the selection of content, teaching strategies, and assessment methods. The following table 1 shows the course objectives for a software engineering course.

Table 1: Course Objectives



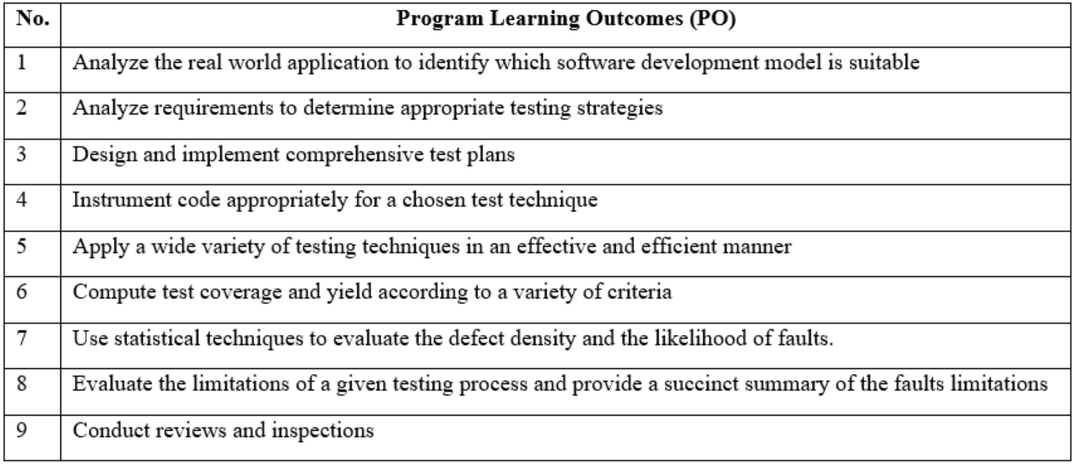
No.	Course Objectives (CO)
1	To understand the rapid software application development and Agile methodologies
2	To understand the software testing process and different types of software testing process
3	To develop the test plan, Test case and Test suite for the software development
4	To understanding that the software needs to be tested for bugs and to insure the product meets the requirements and produces the desired results.
5	To understand the software testing is essential for providing a quality product.
6	To gain software testing skills to start a career or a software developer looking to improve unit testing skills

These course objectives align with the levels of Bloom's Revised Taxonomy and provide a clear direction for the course content and activities. By assessing students' achievement of these objectives, instructors can evaluate the effectiveness of their teaching strategies and make adjustments to improve student learning outcomes.

2.2. Learning Outcomes

Learning outcomes are specific and measurable statements that describe what students should be able to do as a result of their learning experience. They focus on the knowledge, skills, and attitudes that students are expected to develop during a course or program. The following table 2 shows the learning outcomes for a software engineering course.

Table 2: Program Learning Outcomes



No.	Program Learning Outcomes (PO)
1	Analyze the real world application to identify which software development model is suitable
2	Analyze requirements to determine appropriate testing strategies
3	Design and implement comprehensive test plans
4	Instrument code appropriately for a chosen test technique
5	Apply a wide variety of testing techniques in an effective and efficient manner
6	Compute test coverage and yield according to a variety of criteria
7	Use statistical techniques to evaluate the defect density and the likelihood of faults.
8	Evaluate the limitations of a given testing process and provide a succinct summary of the faults limitations
9	Conduct reviews and inspections

Learning outcomes provide a clear understanding of what students should be able to do at the end of the course, and help to guide the selection of teaching methods and assessment strategies. By assessing students' achievement of these learning outcomes, instructors can evaluate the effectiveness of their teaching methods and make adjustments to improve student learning outcomes.

3. METHODOLOGY

3.1. Overview of Learning Activities

Learning activities figure 1 refer to a wide range of actions and strategies that facilitate the acquisition and development of knowledge, skills, and attitudes. These activities may be formal or informal, and can take place in a variety of settings, including classrooms, workplaces, and online environments. Learning activities are designed to be engaging, meaningful, and relevant to learners' needs and interests, and to support their ongoing growth and development.

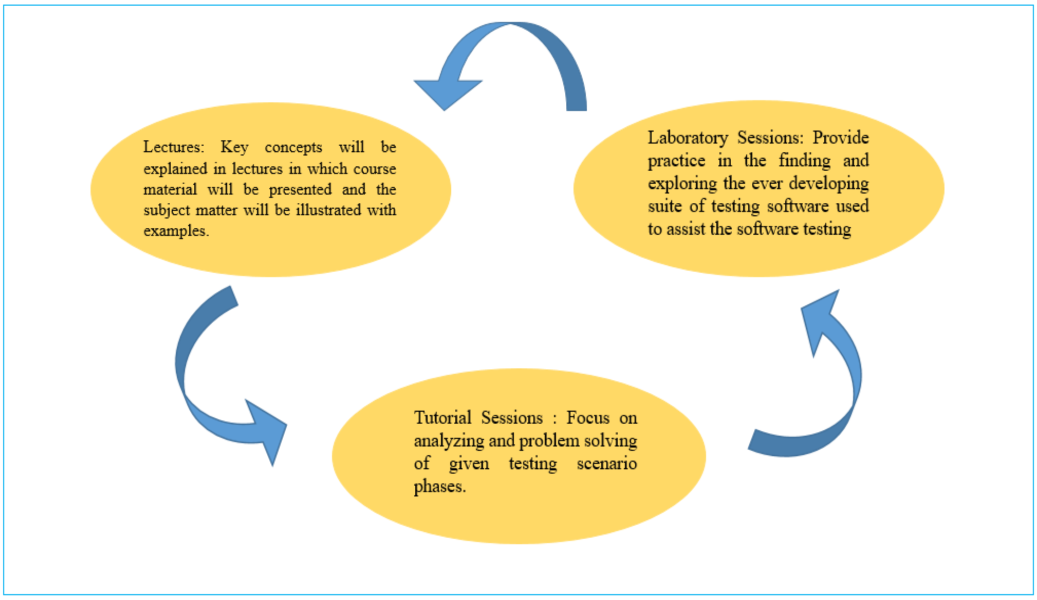

Lectures: Key concepts will be explained in lectures in which course material will be presented and the subject matter will be illustrated with examples.
Laboratory Sessions: Provide practice in the finding and exploring the ever developing suite of testing software used to assist the software testing
Tutorial Sessions : Focus on analyzing and problem solving of given testing scenario phases.

Figure 1: Overview of Learning Activities

3.2. Course Policy

A course policy typically refers to a set of guidelines, rules, and expectations that outline the procedures and requirements for students enrolled in a particular course. These policies may cover a wide range of topics, such as attendance, grading, assignments, communication with the instructor, academic integrity, and accommodations for students with disabilities.

Course policies tables 3 are usually included in the course syllabus, which is typically distributed to students at the beginning of the course. It is important for students to carefully review and understand the course policies, as they can affect their grades and overall success in the course.

If you are a student enrolled in a course, it is important to adhere to the course policies and communicate with the instructor if you have any questions or concerns. If you are an instructor, it is important to create clear and fair course policies and communicate them effectively to your students.

Table 3: Course Policies

1	Attendance is not mandatory but highly recommended.
2	Coming to class fully prepared helps push the conversation much further.
3	Make-up exams, tutorial and assignments are not available.
4	Individual deliverables are to be submitted individually and group work is collaborative.
5	All exams and assignments are to be completed by the student alone with no help from any other person.
6	Students are allowed to discuss about homework and project problems with others.
7	Students are not allowed to copy the contents of a white-board after a group meeting with other students.
8	Students are not allowed to copy the solutions from another colleague.
9	If so, the all students (with same solution) must be deducted 0.5% of their marks (assignment).
10	If you hand in a late assignment, you must identify (1) how late this assignment is and (2) how many total slip days you have left.
11	After you have used up your slip time, any assignment handed in late will be marked off 0.5% per day. That is, after 3 days, the mark will be zero.
12	In extreme circumstances (e.g., medical emergencies), we will grant no-penalty extensions. Please be prepared to provide written documentation (e.g., doctor's note).

3.3. Revised Bloom's Taxonomy

The revised bloom's taxonomy is a classification of the different goals for their students, which are also called learning objectives. There are 6 major categories: Remember, Understand, Apply, Analyze, Evaluate and Create. The following six levels of learning are shown in table 4.

Table 4: Six levels of Revised Bloom's Taxonomy

1.	Remember	Recall facts and basic concepts (Action verbs include: define, duplicate, list, memorize, repeat, state)
2.	Understand	Explain ideas or concepts (Action verbs include: classify, describe, discuss, explain, locate, recognize, report, select, translate)
3.	Apply	Explain ideas or concepts (Action verbs include: execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch)
4.	Analyze	Explain ideas or concepts (Action verbs include: differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test)
5.	Evaluate	Explain ideas or concepts (Action verbs include: appraise, argue, defend, judge, select, support, value, critique, weigh)
6.	Create	Explain ideas or concepts (Action verbs include: design, assemble, construct, conjecture, develop, formulate, author investigate)

3.4. Mapping Learning Outcomes to Course Outcomes

Mapping learning outcomes to course outcomes is an important step in ensuring that a course is aligned with the overall learning objectives of a program or institution. Learning outcomes are the specific knowledge, skills, or abilities that a student is expected to gain from a course or program, while course outcomes are the specific goals and objectives of a particular course.

The mapping process involves identifying the learning outcomes that are addressed by each course outcome and ensuring that there is a clear alignment between the two. This can be done by reviewing the course syllabus and identifying the specific activities, assignments, and assessments that are designed to achieve each course outcome.

By mapping learning outcomes to course outcomes, instructors can ensure that their course is focused on the most important knowledge and skills that students need to master, and that there is a clear connection between what students are learning in the course and the broader goals of the program or institution. This can also help instructors to design more effective assessments and evaluate student learning more accurately. Mapping learning outcomes to course outcomes table 5 and figure 2 show an important tool for ensuring that a course is well-designed and aligned with broader educational goals.

Table 5: Mapping of CO and LO

Course Objectives	Learning Outcomes								
	LO1	LO2	LO3	LO4	LO5	LO6	LO7	LO8	LO9
CO1	√								
CO2		√	√	√	√	√		√	
CO3		√	√	√	√	√		√	
CO4		√					√		
CO5					√	√			
CO6					√		√		√

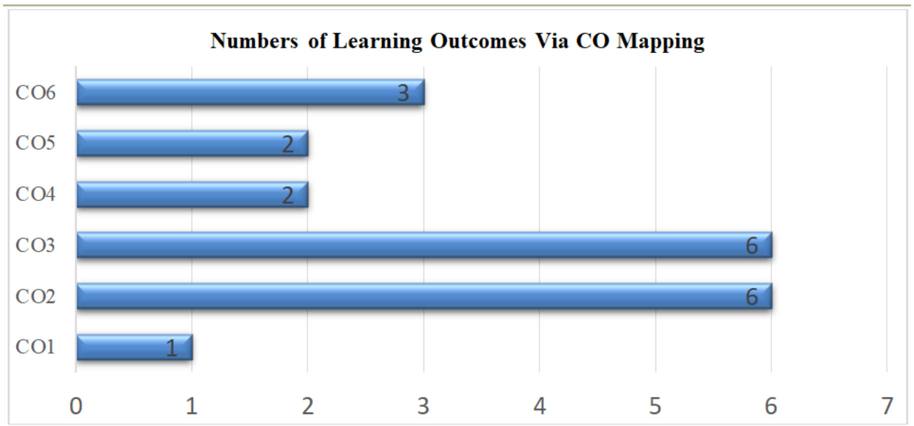
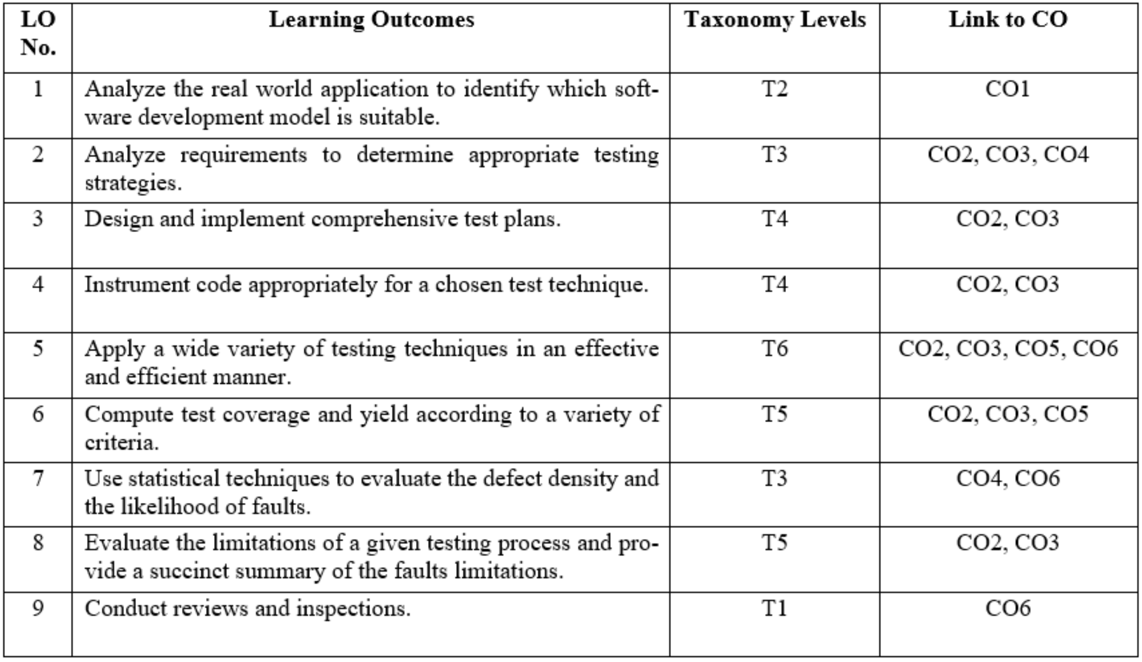


Figure 2: Mapping of Learning Outcomes and CO

5. FINDINGS AND DISCUSSIONS

Linking course outcomes to taxonomy levels table 6 can help to ensure that course goals are clear, meaningful, and measurable, and that students are appropriately challenged and supported in their learning. Taxonomy levels are a framework for categorizing and organizing educational goals and objectives according to their complexity and cognitive demand.

Table 6: Learning Outcomes with Taxonomy levels and Linking of CO



LO No.	Learning Outcomes	Taxonomy Levels	Link to CO
1	Analyze the real world application to identify which software development model is suitable.	T2	CO1
2	Analyze requirements to determine appropriate testing strategies.	T3	CO2, CO3, CO4
3	Design and implement comprehensive test plans.	T4	CO2, CO3
4	Instrument code appropriately for a chosen test technique.	T4	CO2, CO3
5	Apply a wide variety of testing techniques in an effective and efficient manner.	T6	CO2, CO3, CO5, CO6
6	Compute test coverage and yield according to a variety of criteria.	T5	CO2, CO3, CO5
7	Use statistical techniques to evaluate the defect density and the likelihood of faults.	T3	CO4, CO6
8	Evaluate the limitations of a given testing process and provide a succinct summary of the faults limitations.	T5	CO2, CO3
9	Conduct reviews and inspections.	T1	CO6

By linking course outcomes to taxonomy levels, instructors can ensure that their teaching strategies and assessments are aligned with their desired learning outcomes. For example, if the course outcome is to have students analyze a particular concept, then assessments and learning activities should be designed to challenge students to break down the information and identify its component parts.

The outcomes of software engineering can vary depending on the specific project and its goals, but some common outcomes include:

1. High-quality software products: Through the use of software engineering principles and practices, software products can be developed that are reliable, efficient, and maintainable.
2. Timely delivery of software products: The use of project management techniques and software development methodologies can help ensure that software products are delivered on time.
3. Cost-effective software development: By following software engineering best practices, unnecessary rework and delays can be avoided, which can reduce the cost of software development.
4. Improved collaboration and communication: Software engineering emphasizes collaboration and communication between developers, testers, and other stakeholders, which can help ensure that everyone is working towards the same goals.
5. Increased customer satisfaction: Software engineering can help ensure that software products meet customer requirements and expectations, which can lead to increased satisfaction and repeat business.
6. Continuous improvement: Software engineering encourages continuous improvement through the use of feedback mechanisms and iterative development processes, which can help ensure that software products remain relevant and effective over time.

6. CONCLUSIONS AND RECOMMENDATIONS

Mapping learning outcomes (LOs) to course objectives (COs) is an essential process that ensures alignment between the learning goals of a course and the specific objectives that students are expected to achieve. This mapping process helps instructors to design effective teaching and learning activities that are aligned with the learning outcomes and course objectives. The following are some recommendations that can be drawn from LO and CO mapping:

1. Clear alignment: When LOs and COs are well-mapped, it is clear that the course content, teaching strategies, and assessment methods are aligned with the intended learning outcomes. This ensures that the course is focused on the most important concepts and skills, and that students have a clear understanding of what they are expected to achieve.
2. Effective assessment: Mapping LOs to COs helps instructors to design effective assessment methods that measure the achievement of specific learning

outcomes. This ensures that the assessment is valid and reliable, and that it accurately measures student learning outcomes.

3. Improved student learning: When LOs and COs are aligned, students are more likely to achieve the intended learning outcomes. This is because the course content, teaching strategies, and assessment methods are all designed to support the same learning objectives. This can lead to improved student engagement, motivation, and overall learning outcomes.
4. Continuous improvement: Mapping LOs to COs also provides a framework for continuous improvement. By assessing student achievement of specific learning outcomes, instructors can identify areas of the course that may need improvement and make adjustments to improve student learning outcomes over time.

LO and CO mapping is an important process that ensures alignment between the intended learning outcomes of a course and the specific objectives that students are expected to achieve. This mapping process helps to ensure that the course is focused on the most important concepts and skills, and that students have a clear understanding of what they are expected to achieve.

REFERENCES

1. Bloom, B., Engelhart, M., Furst, E., Hill, W., & Krathwohl, D. (1984). Taxonomy of Educational Objectives: The Classification of Educational goals: Cognitive Domain. New York, NY, USA: Longman.
2. Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., ... & Steece, B. (2009). *Software cost estimation with COCOMO II*. Prentice Hall Press.
3. Reifer, D.J. (2008). “Software Management”, 6th Edition.
4. Tin, H.H.H. (2019). “Analysis the Learning Outcomes Survey of the Software Project Management. *International Journal of Future Trends in Engineering and Science, 02*(02).
5. Sommerville, I. (2016). “Software Engineering”, Global Edition, 10th Edition.
6. Anderson, L. (2001). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives. New York, NY, USA: Addison-Wesley.
7. Syeed, M. M., Shihavuddin, A. S. M., Uddin, M. F., Hasan, M., & Khan, R. H. (2022). Outcome Based Education (OBE): Defining the Process and Practice for Engineering Education. *IEEE Access*, *10*, 119170-119192.
8. OBE Committee. (2010). OBE Implementation Guidebook, Faculty of Electrical & Electronic Engineering.
9. Mall, R. (2014). “Fundamentals of Software Engineering”, 4th Edition.
10. Pressman, R.S. (2015). “Software Engineering”, A practitioner’s Approch, 9th Edition.
11. https://www.onlineengineeringprograms.com