**GLOBAL ACADEMIC RESEARCH INSTITUTE**

COLOMBO, SRI LANKA


**GARI International Journal of Multidisciplinary Research**

ISSN 2659-2193

**Volume: 09 | Issue: 04**

On 31st December 2023

http://www.research.lk

Author: Dr. Rasika Aponsu

IIC University of Technology, Cambodia

GARI Publisher | Software Engineering | Volume: 09 | Issue: 04

Article ID: IN/GARI/JOU/2023/170 | Pages: 27-33 (07)

ISSN 2659-2193 | Edit: GARI Editorial Team

Received: 15.10.2023 | Published: 31.12.2023

# CREATING AN ASSESSMENT STRUCTURE FOR IMMERSIVE LEARNING ENCOUNTERS IN A SOFTWARE ENGINEERING PROJECT COURSE

Dr. Rasika Aponsu

*IIC University of Technology*

*Cambodia*

## *ABSTRACT*

The majority of the ideas and concepts that must be taught in Software Engineering courses are challenging to convey through real-life experiences due to the inherent difficulty in providing students with practical exposure to the underlying insights and processes. There is no effective method for teaching the principles of Agile Scrum and Team Software Process (TSPi) that includes the student, teacher, and business stakeholder. This article will elucidate the notion of a framework for effectively constructing an immersive learning environment for both the student and teacher of the Software Engineering Project course, incorporating the participation of business stakeholders. This offers a chance to shift the focus of learning towards learning design, specifically via the lens of immersive environments, rather than merely gathering knowledge. The online questionnaires were distributed to third-year students enrolled in the Software Engineering Laboratory course, as well as the stakeholders of the projects. The objective of this study is to obtain feedback from both parties about the efficacy and appropriateness of the framework and concept in the instruction and acquisition of the course. This document outlines our expertise in developing, implementing, and refining the course. The assessment involves evaluating the extent to which we have successfully attained the course objectives set by the students and stakeholders.

Keywords: Software Engineering, Experience, Knowledge Management

## *INTRODUCTION*

In the field of Computer Science, particularly in Software Engineering courses, the achievement or lack thereof of students is contingent upon the level of cooperation exhibited among team members. Software Engineering students should get instruction on the theoretical and technical elements of software development, as well as the importance of cooperation and social skills. A crucial element of Software Engineering courses is the construction of a software project. The goal is to create a software solution that caters to users or stakeholders who have the intention of using it on a frequent basis. In addition, students will acquire the skills to effectively address practical challenges encountered in the industrial sector through collaborative teamwork. Software Engineering students should get instruction not just on the theoretical and technical elements of software development, but also on the importance of cooperation and social skills. TME3413 Software Engineering Laboratory is a course offered by the Faculty of Computer Science and Information Technology (FCSIT) at UNIMAS. The course was developed by Jali, Masli, Shiang, Bujang, and others in 2017. This opportunity is available to third-year students enrolled in the Software Engineering degree who

have acquired a solid understanding and proficiency in programming languages, scripting, software modelling, and database management systems (DBMS). The acquisition of these abilities and information enables students to cultivate many types of systems.

**Background**

The Future Ready Curriculum involves a revolutionary approach to teaching and learning, namely through Immersive Learning. This method allows students to get practical experience and actively connect with the business and community, enabling them to find and share their experiential learning. (MOHE, 2018). According to Sharples, Immersive Learning allows individuals to fully engage in a scenario as if they were physically present, utilising their knowledge and resources to address a problem or acquire new abilities. Augmented Reality (AR) or Virtual Reality (VR) are commonly used in virtual environments to enhance the sense of partial immersion during the learning process (2019). Implementing the principles of Software Engineering project development using this learning approach allows students to gain practical experience in the business by working on a genuine case study, engaging with stakeholders and the community. This enables pupils to inquire, acquire problem-solving methods, and acquire knowledge via the process of learning. Immersive Learning Experiences prioritise face-to-face teaching and blended learning approaches to deliver powerful and meaningful learning. This approach actively engages students in work or projects outside the confines of the classroom (Carroll, 2014).

Students will engage in communication with local businesses, as well as participate in the development of a high-quality software product that necessitates a comprehensive grasp. Consequently, students will acquire the skills to do user analysis, discern a value proposition, and scrutinise user experience data. Moreover, the ability to collaborate well in a team and possess exceptional time management abilities are crucial competencies required to achieve a shared objective (Bruegge et al., 2015; Rodriguez et al., 2015). Students' oral communication abilities are used to evaluate active learning. Students are provided with the chances to engage in both discussions and observations of their classmates. Real-world software development project experiences are thought to promote the improvement of students' writing and spoken communication abilities (DiYanni et al., 2020; Offutt, 2013). Students will get support in cultivating problem-solving, critical thinking, and analytic abilities, all of which are vital tools for preparing for improved decision-making, academic excellence, and ultimately, enhanced employability.

Previous courses have provided students with the essential foundations, ideas, and concepts, but they have not been given the opportunity to use these abilities in real-world project environments with genuine stakeholders. Assessments mostly address case studies derived from textbooks or capstone projects, without student involvement in real-world initiatives and actual clients. This course aims to provide students with practical experience in utilising Agile software development approaches through participation in industry experience projects. The industries will consist of small and medium-sized local entrepreneurs or stakeholders. Throughout the semester, the project team will collaborate in completing the whole development cycle, encompassing comprehension of the requirements to the delivery of a fully operational product. The customer and course coordinator will

be provided with a series of presentations and reports detailing the work.

## *METHODOLOGY*

The TME3413 – Software Engineering Lab course implemented the Team Software Process (TSPi) and Agile software methodology to manage software projects in the industry (Jali, Shiang, Masli, & Asmadiah, 2019; Over, 2000; Sussy, Calvo-Manzano, Gonzalo, & Toms, 2008). TSPi offers a set of operational procedures to software engineers, enabling them to better plan software development projects and enhance the quality and productivity of their work. Agile Software development prioritises person and interaction, working software, stakeholder engagement, and responsiveness to change. It is a strategy used in the software development process that involves iterative and incremental steps. Both procedures have similarities in terms of allocating tasks and duties and identifying and addressing risks and difficulties to overcome obstacles (Jovanović et al., 2015).

Three crucial aspects that impact software quality and the effectiveness of a team or organisation are the product, the team or individuals involved, and the technology used. Figure 1.0 depicts the process positioned in the middle of a triangle, connecting three components. The efficiency of the software process is evaluated by considering metrics like as defects, productivity, calendar time, and other relevant measures. Additionally, the quality and performance may be influenced by three environmental factors: the characteristics of the customers/stakeholders in terms of communication, the business circumstances in terms of needs and norms, and the development environment in terms of software tools.

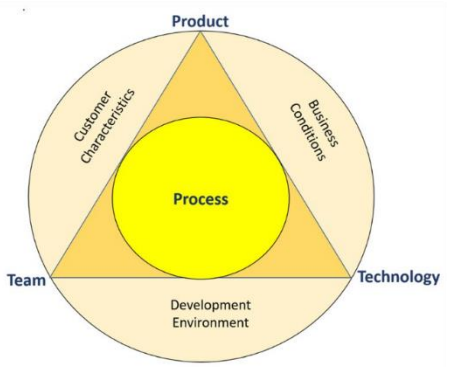


*Figure 1.0. Software Engineering Lab Project Framework (Jali, Masli, Shiang, Mat, et al., 2017)*

### A. Product

• The teams were provided with a roster of practical projects, each including stakeholders from small-scale businesses.
• Additionally, the teams were encouraged to identify their own business stakeholders that align with the project's objectives.

• Established clear objectives for their team and project, as well as identified key milestones for the project.

• Obtained requirements by several approaches including conversations, observations, analysis, and synthesis. The deliverables encompass software requirements and design papers, such as the Software Requirements Specification (SRS), Software Design Specification (SDS), functioning system, concise test report, and user manuals.

### B. Technology

• Enhancing the effectiveness of the software development process by using advanced software tools and methodologies.

• Employ several technologies for project management, such as the TSPi support tool, Trello application, AgroUML, GitHub, Dropbox, and others.

• Programming techniques: Unified Modelling Language (UML), Java, PHP, and other related technologies.

### C. Team

• The composition of each team consists of 5 to 7 individuals, depending on the enrollment of students in the course.

The project team places significant importance on defining and fulfilling certain roles and responsibilities.

• Effective teamwork and communication among team members, teacher, and clients.

### Implementation of the Project

Due to the unique nature of each software project, there is no predetermined set of deliverables that is universally applicable to every project. One aspect of the teams' responsibility is to determine the exact requirements for this project. Common deliverables encompass functional code, comprehensive documentation, instructional resources, and comprehensive test suites. The key factors for a project's success include meeting the client's requirements, ensuring the product's usability, and ensuring its maintainability throughout its lifespan. These projects are collaborative, but, students will also get individual rewards for making additional contributions to the project or if they fail to provide their fair amount of effort. The project comprises two sequential stages of a software product. The initial phase is characterised by exploration and serves as the initial effort to construct the suggested software product. The deliverables for the initial phase consist of two reports, namely Report 1 (updated proposal and Software Requirements Specification) and Demo 1. Subsequently, the instructors will furnish comments, prompting the pupils to reassess and perhaps amend the project objectives prior to proceeding to the second step. Modifying the target over the semester is permissible, particularly when the teams acquire additional knowledge about the project and have a clearer understanding of what they can realistically do within the given period. The deliverables for the second phase consist of report 2, which includes updated Software Requirements Specification (SRS), Software Design Specification (SDS), Test logs, and other related documents, as well as demo 2.

*Table1.Project Deliverables*

| Project Deliverables | | | | | |
|---|---|---|---|---|---|
| Phase 1 | | | Phase 2 | | |
| Proposal Presentation | Initial Report | Demo 1 | Final Report | Demo 2 | e-Archive |
| Week 5 | Week 10 | Week 10 | Week 13 | Week 14 | Week 15 |

This course primarily evaluates the team's end output, such as design, report, and presentation, as well as their group practices, including their ability to meet deadlines, contribute equally, and communicate effectively. Written examinations are not used for assessment in this course. The course evaluations are outlined below;

*Table 2. Course Assessment*

| Course Assesments | | | | | | | |
|---|---|---|---|---|---|---|---|
| Continuous Assessment (70%) | | | | | Final Assessment (30%) | | |
| Project Proposal & Planning | Requirement Specifications | Design and Implementation Specification | Testing & Release | Presentation | Peer Evaluation | Individual Evaluation (Personal log, participation, presentations, etc.) | Team Evaluation (Foundation and functioning) |
| 10 | 15 | 25 | 10 | 10 | 10 | 10 | 10 |

This course offers a significant and valuable learning experience for your students.

i. Enhance students' comprehension of ideas by illustrating their practical implementation in intricate real-life scenarios

ii. Engage students in participatory learning and foster dialogue.

iii. Develop and cultivate interpersonal or communication abilities iii. Instruct the pupils on the efficient management of time.

Student Engagement/Involvement in Academic Activities

The Bloom's Taxonomy (Anderson, Krathwohl, & Bloom, 2001) is utilised to construct the learning activities of the projects, with the aim of facilitating teaching and learning. The system has three hierarchical models that categorise the learning objectives into three domains: Cognitive (knowledge), Psychomotor (skills), and Affective (attitudes). Bloom's taxonomy categorises intellectual talents into six stages within the Cognitive domain. These levels, structured in a hierarchical manner, go from basic to more advanced actions. They are: knowledge, understanding, application, analysis, synthesis, and assessment (Bloom, 1956). The Psychomotor domain encompasses skills that pertain to physical movement, coordination, and the utilisation of motor skills. The Affective domain encompasses five domains that pertain to emotional aspects, which are likewise organised in a hierarchical manner, ranging from basic to more intricate ways of managing emotions: receiving, responding, valuing, organising, and characterising (Bloom, 1956) (Krathwohl, Bloom, & Masia, 1964). Consequently, the learning activities of the projects will facilitate the students' active involvement with team members, teacher, and stakeholders in a purposeful learning experience including cognitive, psychomotor, and emotional domains.

**The Cognitive Domain**

By employing analytical and creative thinking, teams utilise software approaches to solve situations presented by actual customers and stakeholders. The teams then generate and provide various outputs such as reports, logs, and codes. Facilitate the discourse and demonstration with the teams, stakeholders, and the teacher of the course.

b. Psychomotor Domain refers to the area of learning that involves the development of physical skills and coordination.

Conduct requirement elicitation by using appropriate Computer-Aided Software Engineering (CASE) technologies for each step of software development (including software, hardware, and internet tools).

**The Affective Domain**

Effective time management, strong communication skills, professionalism in working with colleagues and stakeholders, exceptional interpersonal skills, providing constructive criticism, and understanding individual and team duties. The above domains activities pertain to the suggested Software Engineering Project Learning Activities, as seen in Figure 2.0. The utilisation of this pedagogical framework has been implemented for the TME3413 Software Engineering laboratory course since 2017.

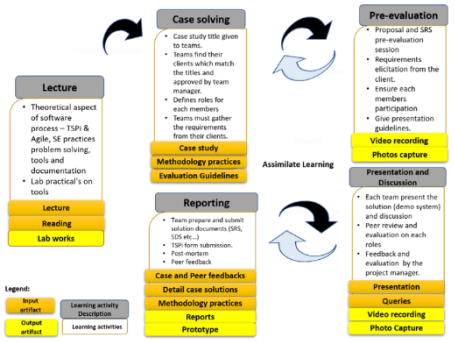


*Figure 2.0. TME3413 – Software Engineering Project Learning Activities*

Given that this study was done and concluded prior to the onset of the COVID-19 pandemic, the framework has been effectively put into practice and the findings have been published in this publication. Nevertheless, because to the ongoing COVID-19 pandemic, certain framework components may not be

appropriate and must be excluded or substituted with alternative methods. For instance, the lockdown has made it difficult to have in-person interactions among students, instructors, and clients due to various constraints such as adhering to the Standard Operating Procedure (SOP), maintaining social distancing, and the restriction on students returning to the university. Therefore, an alternative method to carry out the in-person contacts that will effectively achieve the goals of this framework is through the utilization of online video conferencing platforms such as Skype, Microsoft Teams, Zoom Meeting, Google Meet, and comparable technologies.

**Accomplishments and Evaluation from Students and Stakeholders**

Attainment of Course Learning Outcomes (CLO)

Students participating in the TME3413 Software Engineering Laboratory course are obligated to complete an assessment form assessing the quality of the course material, teaching, and delivery. According to the comments from the students, all the Course Learning Outcomes (CLOs) have been successfully accomplished (see Figure 6.0). The following is a list of the CLOs:

CLO1: Apply Software Engineering principles in a collaborative setting by working on a practical application.

CLO2: Illustrate the benefits of employing several Computer-Aided Software Engineering Tools (CASE Tools) at key stages of software development.

CLO3: Collaboratively develop and implement a comprehensive application to address a specific organizational challenge.

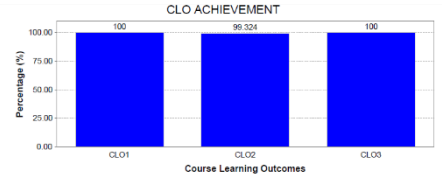


Table: CLO Achievement

| | CLO1 | CLO2 | CLO3 | Number of CLO Achieved |
|---|---|---|---|---|
| G01 | 148 | 147 | 148 | 443 |
| No of student who achieved LO | 148 | 147 | 148 | 443 |

*Figure 6.0. CLO Achievement*

## *CONCLUSION*

This Software Engineering course facilitates the practical application of academic ideas, bridging the gap between theory and practice by transitioning from the classroom to the real working world. The students possess the capability to utilise the information and substance acquired in the course in practical, real-life scenarios. Consequently, they will assess the efficacy of implementing an Agile software methodology and Team Software Process (TSPi), which prioritises team-based software development projects in undergraduate education. The course facilitates successful learning by providing an immersive setting that enables students to cultivate essential skills such as communication, teamwork, and problem-solving. The students will acquire the skills to proficiently communicate and interact with development teams in order to successfully execute projects from inception to finalisation. Consequently, at the conclusion of the course, they comprehended the significance of quality as a fundamental principle that should permeate all aspect of a software development business. Upon meeting the stakeholder's demands and criteria, the project will ultimately enhance the students' motivation to study and instill a sense of confidence in their ability to work

in a real-world setting. As a result, they will be well-prepared for their upcoming internship programmes in the next semester.

**Acknowledgement**

The authors express their gratitude to the reviewers for their valuable and meticulous feedback, which greatly enhanced the paper's quality.

## *REFERENCES*

*Bloom, B. (1956). Taxonomy of educational objectives. Vol. 1: Cognitive domain. New York: McKay.*

*B ruegge, B . , Krusche, S . , & Alperowitz, L. (2015). S oftware engineering project courses with industrial clients. ACM Transactions on Computinig Education. https://doi.org/10.1145/2732155*

*Carroll, J. M. (2014). Innovative practices in teaching information sciences and technology: Experience reports and reflections. Innovative Practices in Teaching Information Sciences and Technology: Experience Reports and Reflections, 9783319036(January 2014), 1–238. https://doi.org/10.1007/978- 3-319-03656-4*

*TSPI ) Practices in the Software Engineering Undergraduate Course. Journal of IT in Asia, 7(1), 1–8.*

*Jali, N., Masli, A. bin B., Shiang, C. W., Mat, A. R., Bujang, Y. R., & Hamdan, N. M. (2017). Software Development in Software Engineering Course-Looking into Project Planning and Estimation using Team Software Process (TSPi) and Scrum. SoTL Bulletin Vol 1. 2017 . Universiti Malaysia Sarawak, 51.*

*Jali, N., Shiang, C. W., Masli, A. B., & Asmadiah, N. (2019). Team Software Process ( TSPi ) Web-Based Support Tool. International Journal of Recent Technology and Engineering.*