International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I9P140

Volume 72 Issue 9, 428-435, September 2024
© 2024 Seventh Sense Research Group®

Original Article

The Interaction Learning Models of Skill Mapping for Software Tester

Davit Sanpote

School of Information and Communication Technology, University of Phayao, Thailand.

Corresponding Author: davit.sa@up.ac.th

Received: 16 May 2024 Revised: 08 August 2024 Accepted: 17 September 2024 Published: 28 September 2024

***Abstract** - Several countries are working to improve education in the current period by implementing collaborative thinking and cooperative methods in 21st century learning. They use technology as a key component to make learning sessions interactive and engaging. Students can use what they have learned fast and precisely. The goal of the new teaching and learning methodology is to promote innovation in postsecondary education. Inspired by the "Hooked Model," a framework emphasizing fast transition, the teaching model in the course "Software Testing for Second-Year Students in Software Engineering program, the University of Phayao" The ADDIE Model serves as the foundation for the development of the novel teaching strategy. It includes five stages of development: analysis, design, development, execution, and assessment. Because each step is meant to be adaptable, instruments that enhance instruction can be made. During the design stage, the researchers integrate the concepts of the ADDIE Model and the Hooked Model. The anticipated result is the creation of a blended learning model to improve teaching strategies. The objective is to develop a creative and adaptable teaching methodology appropriate for the changing nature of education in the twenty-first century and beyond.*

***Keywords** - ADDIE Model, Interaction Learning, Hook Model, Skill Mapping, 21st century community change agent.*

1. Introduction

At the moment, a lot of countries are committed to improving education by emphasizing interactive models over theoretical ones, going beyond only stimulating learning. This method is intended to adapt to the shifting patterns of global development and modern education in the 21st century. The focus of education in the 21st century is on learner-centred learning management. It encourages and facilitates educators and students to work together to produce new ideas and knowledge [13]. This is accomplished by using technology as a key component of interactive and participatory learning experiences, along with collaborative thinking and teamwork. It is encouraged for students to communicate, take an active role in their education, identify pertinent material, and apply knowledge effectively. The role that educators play in the learning paradigm is changing. Teachers become facilitators of easy learning management instead of just being knowledge providers [8]. Their job is to support students in becoming autonomous creators of information and inventions by providing a variety of learning activities and evaluations that are in line with each student's abilities. The "Hook Model" serves as the inspiration for the teaching methodology in the course "Software Testing for Second-Year Students" of the University of Phayao's School of Technology and Communication's Software Engineering program. This strategy incorporates behavioral learning and focuses on being responsive to learners [2][3]. Observing how learners behave helps educators create curricula that can change students' behavior to better fit the goals of the course [4]. Trigger, Action, Reward, and Investment are the four stages of development that make up the Hook Model [9] [7].

This model is originally derived from product design, similar to platforms like Facebook and Twitter, which cater to users' needs through the designers' products. In the context of education, the Hook Model is comparable to a course or subject that instructors must analyze and design to be consistently engaging and stimulating for learners. The innovation in this teaching management model is still in its foundational framework. The ADDIE Model is used in the design phase, and the anticipated outcome is the development of an innovative model suitable for evolving education [10] in the 21st century and beyond.

2. Literature Review

2.1. Hook Model for Teaching Software Testing

Digital services such as Instagram and Uber leverage the Hook model to encourage continuous usage, which increases user engagement and loyalty [9]. Uber needs fewer triggers since it caters to specific customer demands, whereas Instagram uses a lot of them to draw users into and between interactions. Both platforms place a strong emphasis on the necessity of investment in the above rewards to create user habits and future Hook cycles.

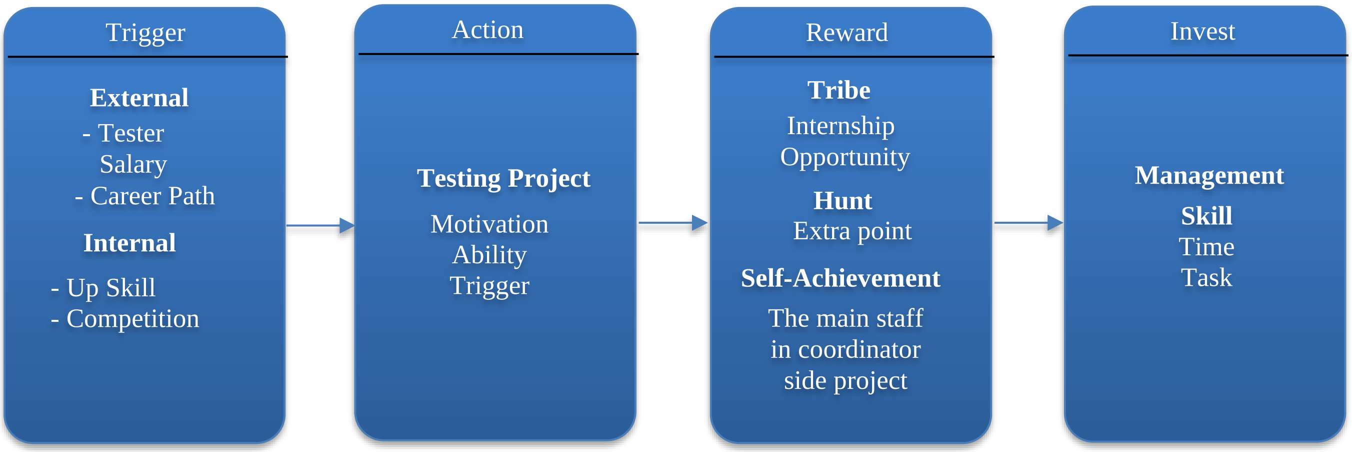

Fig. 1 Hook model process cooperation with software engineering curriculum

Uber needs to keep giving out rewards to keep users happy, whereas Instagram can utilize reward failures as internal triggers to get people back on board. When it comes to external transitions, Uber places more emphasis on user return experiences driven by profit than Instagram, which emphasizes frequent transitions incorporating rewards and triggers. Since the primary focus of applications such as Uber is on service delivery, user satisfaction, and usability, fewer internal triggers may be required in their design [7]. The Hook model approach, depicted in Figure 1, was the basis for Software Engineering (SE) education. The trigger state pointed at career paths in the external aspect and the internal focused on re-skill up-skill level. In action, the state SE curriculum used a software testing project within the classroom to track the student's progress. The reward state is another significant point to motivate and gather students on track. Extra points and student opportunities have been located in this state. Management skills such as time and task management are indicated in the last state.

2.2. Designing learning structure using the ADDIE Model

The confluence of the Waterfall model and the ADDIE paradigm in developing learning applications emphasizes. The importance of system development and instructional design in meeting user and system needs [2]. By strengthening the application's technological aspects and design solutions, this integration can guarantee the production of excellent instructional learning materials [10]. The study also emphasizes how crucial it is to conduct thorough analyses during the design, development, testing, and implementation phases of creating effective educational systems. User Acceptance Testing (UAT) emerges as a critical step in verifying and validating educational material before its delivery to end users. Software Testing is one of the important subjects to include as a major skill of software testers. Figure 2 in this research uses the software testing subject as a case study to track student project progress based on the ADDIE model from the begin the Analyze phase until the Evaluate phase. Hook models have been included along with the ADDIE model during the development and implementation phase. Moreover, the expected result of this process is to find significant factors to improve and update the software testing skill set further.

2.3. Skill Mapping in Software Tester

Software engineering major at the University of Phayao pointed to five careers that are the most related to software engineering abilities noted in the red umbrella model in Figure 3 such as UX/UI Designer, Frontend Developer, Backend Developer, Project Manager/Coordinator, and QA/Software Tester [1]. These careers gain basic knowledge in software engineering courses [12]. After the student completes a mid-year course, they can find their proficiency to decide to improve their skills within the five careers provided.

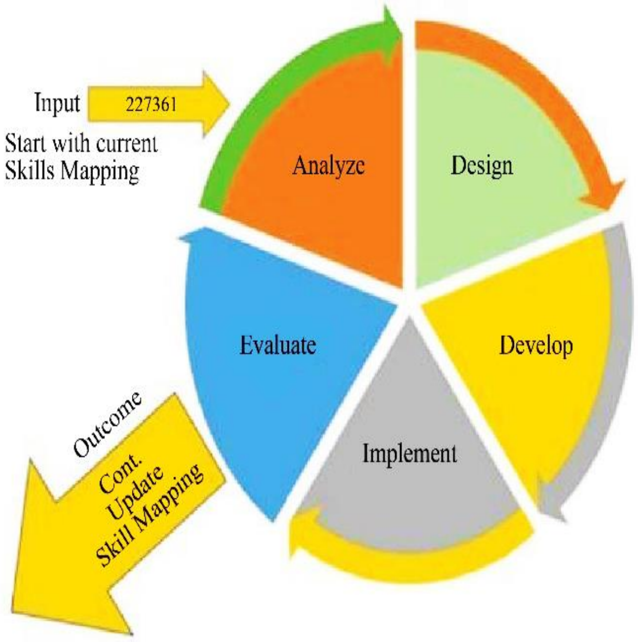

Fig. 2 ADDIE model process cooperation with software engineering curriculum

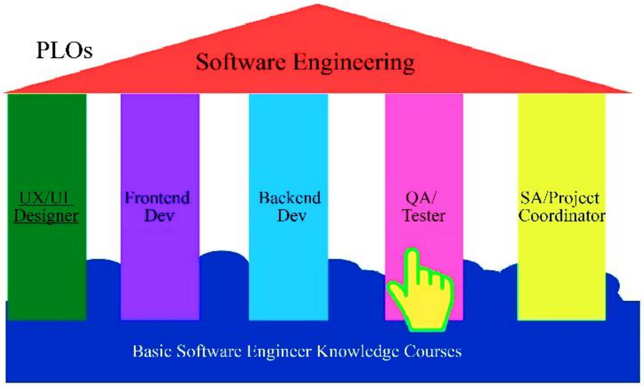

Fig. 3 Program learning outcome for software engineering student

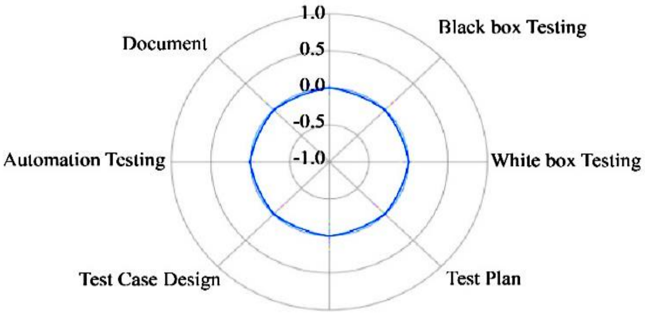

Fig. 4 Software tester skill set in an ideal 2024

According to an analysis of job advertisements, novice software testers need a blend of testing, technical, and soft skills [5, 6]. The software industry highly values these skills, with nearly all job postings requiring testing expertise, the majority demanding technical know-how, and a large portion emphasizing soft skills. To adequately prepare computing graduates for such roles, it is crucial for software testing courses to integrate these identified skills. Employers are looking for a diverse skill set in novice software testers [11], as evidenced by the range of abilities sought in job postings. This underscores the significance of comprehending industry expectations for entry-level software testing positions [11]. Figure 4 represents the 6 significant software testing skills. The result and discussion, part of this research will explain the up-to-date software testing skill set.

3. Research and Methodology

This research divided software engineering students into 2 groups, the control group and the experimental group in the software testing class. These two groups will be compared from the software testing skill set mentioned in Figure 5.

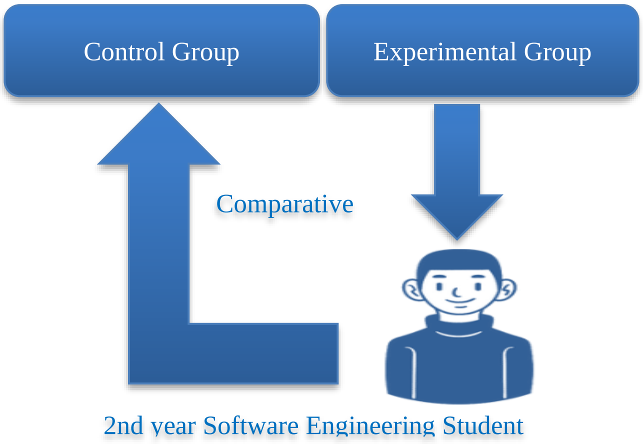

Fig. 5 The control group and experimental group [SEUP 41 Students]

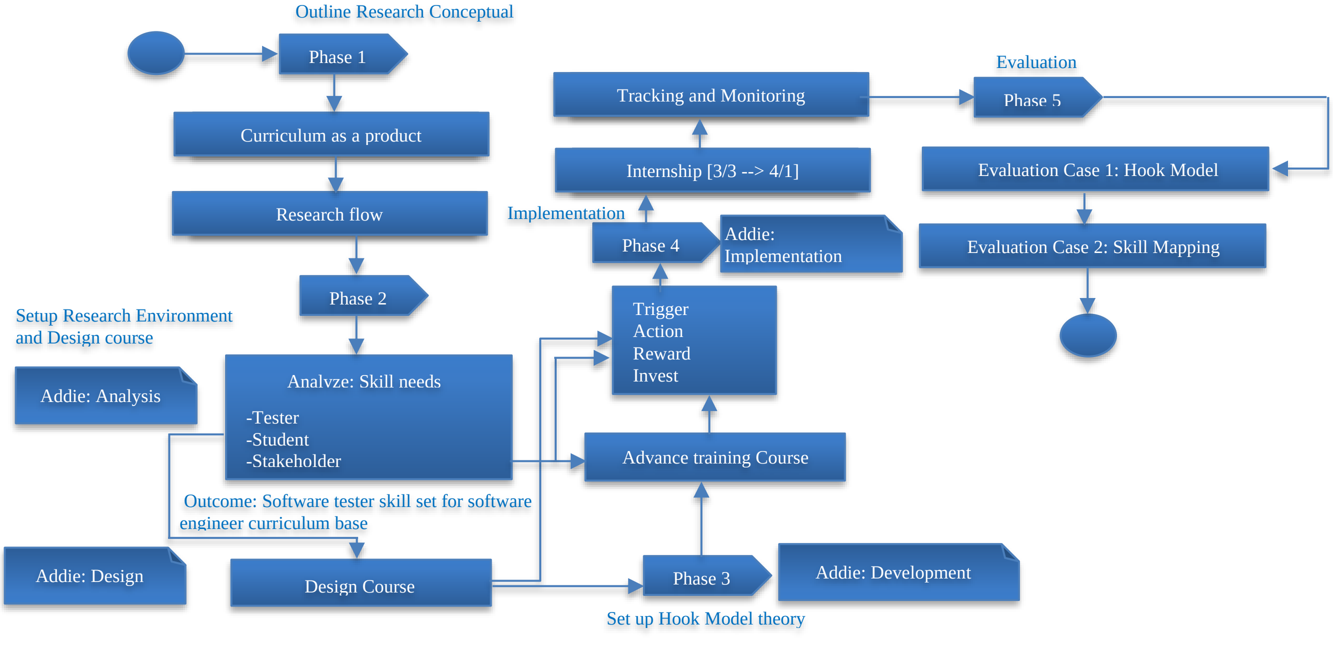

Fig. 6 Research framework process

Table 1. The explanation of 5 phases in the research framework

Phase 1	Explain the objectives of the innovative teaching model along with the process in the course "Software Testing for Sophomores", consisting of 41 students divided into 2 groups using the foundation of the ADDIE Model.
Phase 2	Prepare readiness factors related to the integration of teaching innovation.
Phase 3	Apply the Hook Model theory in the development state. The ADDIE model in the teaching innovation state.
Phase 4	Students apply acquired knowledge to real-world practical field training state.
Phase 5	Evaluate and measure learning outcomes and the effectiveness of the research framework.

Standard software testing methods were assigned to the control group. These methods include the verification and validation model (v-v model), manual testing, basic automated testing, and software testing documentation. The experimental group used the same standard software testing method and included the new tracking and motivating learning model. The Hook and ADDIE model is located at the main objective of this research process.

The research process will integrate the instructional design process using the Hook model in the Develop and Implement phases of ADDIE to develop a new format of instructional innovation, as shown in Figure 6. Throughout the refinement of this instructional format, there will be continuous monitoring and evaluation at each stage until the completion of the process. Additionally, students will be invited to participate in assessing their satisfaction with this learning format. This will extend to reviewing the research objectives to ascertain if they have been achieved according to the research hypothesis by following information of 5 phases in Table 1.

4. Result and Discussion

This section will describe and show the result of updating the software tester skill set that is required from real-world stakeholders who are partners of the University of Phayao, Thailand. There are 2 sample groups, which are the control and experimental groups of 2nd year students in software testing classes with the same teaching standard method of software testing. The difference will be included in the Interaction Learning Methodology with the experimental group. Furthermore, both groups will attend the same phase at phase 4, mentioned in Figure 6 in the SE internship (6 months). This is a significant phase in tracking and monitoring students aligned to software testing skill sets. Lastly, the voice of stakeholders is another important factor to combine with the research result. Moreover, the result of 5 phases will be discussed more below. Phase 1 is the overview of software testing's main concept and explains the research framework to the student. The students will be able to make their own decision to be a volunteer as an experimental group within this research objective or follow a standard software testing teaching methodology as a control group. According to phases 2 and 3 workshops for the experimental group, the software testing class provided two options to work with a mini project. The first option contains three levels to reach, the second option is a little bit more challenging to reach a level that includes a full hard skill set of software tester. The result in Table 2 below shows that the fourth level in option 2 has been the most chosen by students. These are beneficial aspects to note that the research started with a high positive expectation from students which is related to the interaction model supported.

On a top view of these phases managed by the ADDIE model to analyze student and stakeholder needs. The result of the analysis stated in the ADDIE Model shows that black-white box testing and performance testing are the most important for software tester hard skills for undergraduate students. Moreover, the Design and Development have been tracked and monitored every week until the software testing class is complete. The hooked model started to run during phase 3 until the end of phase 5.

The advanced training in performance testing provides an opportunity for outstanding students to be the first queue internship interview from the speaker's company in Figure 7. The hooked model in the state of action and reward for students supports these activities. The researcher has received funding from external sources. This research project is another opportunity for the student to apply their software testing skill set in a real-world project. The last reward in the hooked model is the student volunteer in a real software testing project. After the students join these activities above, they will have a chance to receive an extra point in the software testing class. In phase 4, the data will be collected during students attend six-month internships in the specific field of junior software tester, represented in Figure 8.

Fig. 7 Advance performance testing training camp

Table 2. The ADDIE and Hook model tracking sheet

			Expected Level	Implementation			Evaluation [Skill Mapping Matrix]					
				Investment [Personal link]			Black Box	White Box	Unit test	Automated Test	Performance Test	
	Student ID	Option	Score [17-20%]	Wk1	Wk2	Wk3						
TEAM A	64021799	Option 2	Level 4	back-end	front-end							Level 1
	64023858		Level 4									
	64021698		Level 4		back-end	front-end						Level 2
	64021744		Level 4									
	64021878		Level 4	back-end	front-end							Level 3
	64021711		Level 4	back-end	front-end							
	64021788		Level 4									Level 4
	64021968		Level 4									
TEAM B	64021924	Option 2	Level 3		back-end	front-end						Level 1
	64024398		Level 3									
	64021980		Level 4		back-end	front-end						Level 2
	64022004		Level 3		back-end	front-end						
	64021889		Level 4		back-end	front-end						Level 3
	64023847		Level 3									
	64021777		Level 4									Level 4
TEAM C	64021700	Option 2	Level 4		back-end	front-end						Level 1
	64023229		Level 4									
	64021722		Level 4									Level 2
	64023230		Level 4									
	64021766		Level 4		front-end	font-end						Level 3
	64021654		Level 4		front-end	Backend						
	64021755		Level 4									Level 4
	64021801		Level 4									
TEAM D	64024365	Option 2	Level 4	back-end	front-end							Level 1
	64023252		Level 4	back-end	front-end							
	64021890		Level 4	back-end	front-end							Level 2
	64021845		Level 4									
	64021610		Level 4		back-end	front-end						Level 3
	64021856		Level 4		back-end	front-end						
	64023241		Level 4									Level 4
Option 1				[Reach Level 1,2,3] BlackBox, Automated Test, Performance Test								
Option 2				[Reach Level 1,2,3,4] WhiteBox, BlackBox, UnitTest, Automated Test, Performance Test								

Pranswork Company BigData Company

Fig. 8 Phase 4 junior software tester internship

Table 3. Software tester skill set from stakeholders

Tester Skill Set	**Internship company evaluation**			
	Control Group		**Experimental Group**	
	Dudee indeed	**Prans work**	**Codium**	**20Scoops**
SRS Analyze	3	4	4	4
Black box testing	3	3	5	5
White box testing	3	3	4	5
Test Plan	4	4	4	4
Test Scenarios	4	3	4	4
Test case	4	4	5	4
Document	4	3	4	5

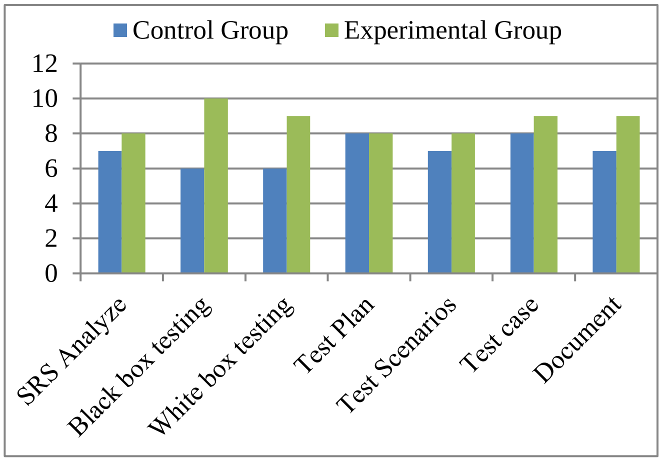

Fig. 9 The summarization chart for the control and experimental group

There are four partner companies divided into two groups. The control group was interning at Dudee Indeed and Pranswork companies. The experimental group was interning at Codium and 20Scoops companies. There are 8 software tester skill sets used to measure such as software requirement specification analysis (SRS Analyse), Black box testing, White box testing, test planning, test scenario, test case design, and report document. Each mentor from four companies has been marked score ranking from low to high level (1 to 5), as shown in Table 3. The result from Table 3 is a remarkable point to update in the skill mapping of software testers. Level 5 has been located in the experimental group in both companies. The high level mentioned in the field of specific software tester skills in black box, white box, and test case design. From an overall perspective, the highest score was noted at Black box testing (10 points).

White box testing, test case design, document report, and SRS analysis are the second majority scores (9 points). The result from Figure 9 can be seen that the experimental group reached a greater score than the control group. The subsequent outcome at step 5, which is the discussion of the hooked model, will use Fogg Behaviour B=MAT [9] to determine whether students are satisfied or not, as shown in the criteria chart in Figure 10.

The Fogg Behavior Model states that for a behavior to occur, it needs to be triggered, have sufficient capacity, and be motivated, as seen in Table 4. If any of these elements are missing, the likelihood of the behavior occurring is decreased [7][9].

Table 4. B=MAT data collected from the experimental group

Parameters	D1: Motivation	D2: Ability	D3: Trigger
Tester Salary	5	3	Motivation
Tester Career Path	5	4	Motivation
Team Project	4	3	Motivation
Internships	5	4	Motivation
Extra point	5	4	Motivation
The staff for the side project	4	4	Motivation Ability
Time management	4	4	Motivation Ability
Total	32	26	

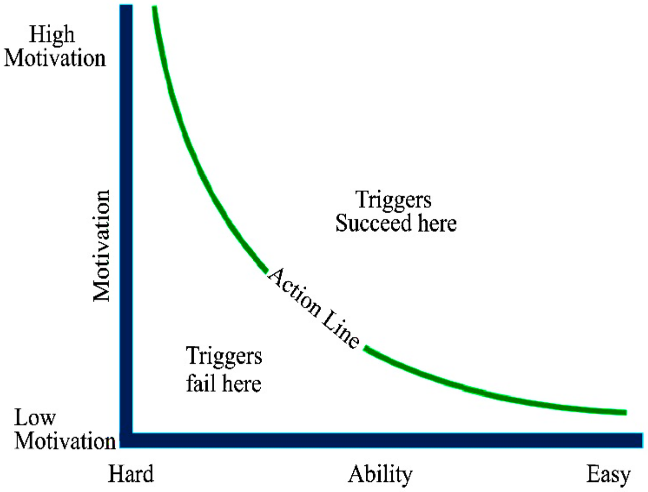

Fig. 10 Ideal char Fogg Behaviour B=MAT

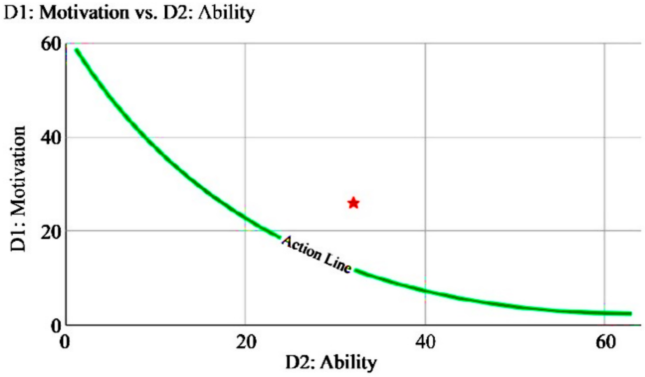

Fig. 11 The B=MAT result chart from the experimental group student

Moreover, the model suggests that behaviors could be impacted by changing these three variables. The possibility that the intended action will be carried out can be increased by raising motivation, making the task easier to perform in order to improve ability, or developing helpful triggers.The result from Table 4 can be summarised in the chart below in Figure 11. The remarkable point is located on the trigger success side with D1 motivation 32 and D2 ability 26. The total result can be proved that the experimental group are satisfied with this Hook model process from the beginning until the end of a class.

According to the result information above, some factors can be able to achieve better results. Firstly, clearly explain about learning process to students. Secondly, create necessary activities based on previous data of black box testing, white box testing and test case design. Lastly, motivation tracking is another important phase. It can be done by providing rewards such as scores and other student benefits opportunities.

5. Conclusion and Practical Implications

Combining the hook model and the ADDIE model of teaching is a method that can significantly enhance student engagement showed a greater number in Black box testing from the experimental group (6 to 10). It can be clearly seen that specific hard skills for software testers are very important to understand clearly and continuous improvement. Another important in this research area is to explain the research flow and the main objective to students and stakeholders mentioned in Phase 1. The hook model is another powerful teaching method. The reward state in the hook model is supposed to be related to other input and out processes to receive a piece of accuracy information. Currently, tech education needs to listen to the voices of customers. The feedback from stakeholders will be necessary data to improve teaching methodologies. On the other hand, the curriculum needs to provide an update on expected learning outcomes or program learning outcomes to its stakeholders, such as students and tech companies.

Furthermore, the real practical is very significant for IT students at this moment. Teaching undergraduate students in the field of technology based on competency base. Finally, this research study is a valuable resource for future investigations to identify additional aspects influencing teaching abilities as business and technology evolve. Additionally, gathering both good and negative feedback from the previous iteration of the software testing teaching framework will help to improve the present edition.

Funding Statement

This research was supported by the School of Information and Communication Technology, University of Phayao, Thailand.
(Grant No RMICT67005)

Acknowledgement

This research study was evaluated by software engineering students at the School of Information and Communication Technology, University of Phayao.

References

[1] Abdulrahman Alarifi et al., “SECDEP: Software Engineering Curricula Development and Evaluation Process Using SWEBOK,” *Information and Software Technology*, vol. 74, pp. 114-126, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Wan Nor Ashiqin Wan Ali, and Wan Ahmad Jaafar Wan Yahaya, “Waterfall-Addie Model: An Integration of Software Development Model and Instructional Systems Design in Developing a Digital Video Learning Application,” *Asean Journal of Teaching and Learning in Higher Education*, vol. 15, no. 1, pp. 1-28, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] D. Carrington, “Teaching Software Design and Testing,” *FIE '98. 28th Annual Frontiers in Education Conference. Moving from 'Teacher-Centered' to 'Learner-Centered' Education, Conference Proceedings (Cat. No.98CH36214)*, Tempe, AZ, USA, vol. 2, pp. 547-550, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[4] Honghong Chen, Xu Wang, and Liangguang Pan, “Research on Teaching Methods and Tools of Software Testing,” *2020 15th International Conference on Computer Science & Education* Delft, Netherlands, pp. 760-763, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Raluca Florea, and Viktoria Stray, “A Qualitative Study of the Background, Skill Acquisition, and Learning Preferences of Software Testers,” *Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering*, New York, United States, pp. 299-305, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Bushra Hamid, and Naveed Ikram, “Industry Perceptions of the Competencies Needed by Novice Software Tester,” *Education and Information Technologies*, vol. 29, pp. 6107-6138, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Elena Lukyanchikova et al., “A Case Study on Applications of the Hook Model in Software Products,” *Software*, vol. 2, no. 2, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Kun Ma et al., “Project-Driven Learning-by-Doing Method for Teaching Software Engineering Using Virtualization Technology,” *International Journal of Emerging Technologies in Learning*, vol. 9, no. 9, pp. 26-31, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[9] Veronica McCauley, Kevin Davison, and Corinna Byrne, “Collaborative Lesson Hook Design in Science Teacher Education: Advancing Professional Practice,” *Irish Educational Studies*, vol. 34, no. 4, pp. 307-323, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[10] Liranti Rahmelina et al., “The Effectiveness of the Flipped Classroom Model Using E-learning Media in Introduction to Information Technology Course,” *International Journal of Emerging Technologies in Learning*, vol. 14, no. 21, pp. 148-162, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Erik van Veenendaal, *Next-Generation Software Testers: Broaden or Specialize!*, The Future of Software Quality Assurance, pp. 229-243, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Qing Hong et al., “Occupational Ability Oriented Graduate Education in Software Engineering,” *International Journal of Emerging Technologies in Learning*, vol. 10, no. 8, pp. 25-29, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] Juan C. Yelmo, and Juan Fernandez-Corugedo, “An Experience of Educational Innovation for the Collaborative Learning in Software Engineering,” *International Journal of Emerging Technologies in Learning*, vol. 6, no. 2, pp. 26-32, 2011. [CrossRef] [Google Scholar] [Publisher Link]