DESIGNING SPACES FOR ACTIVE LEARNING IN TEACHING SOFTWARE ENGINEERING COURSES

Raja-Jamilah Raja-Yusof[1], Aishah Binti Abu Bakar[2], Siti Salwah Salim[3]

Department of Software Engineering, Faculty of Computer Science & Information Technology[1, 3].
University of Malaya, 50603 Kuala Lumpur, MALAYSIA
rjry@um.edu.my[1], salwa@um.edu.my[3]

Department of Civil Engineering, Faculty of Engineering[2].
University of Malaya, 50603 Kuala Lumpur, MALAYSIA
aishah_ab@um.edu.my[2]

Abstract

In recruiting fresh graduates for employment, software companies often look for good technical skills and knowledge of software engineering. Many universities rely on industrial training programmes to expose the students to the real working environment before they complete the course. The students can have the similar experience by carrying out the jobs of software engineering in a simulated real working environment in the university. Active learning is a process in which students participate in the activities that encourage higher-order learning skills, including analysis, synthesis, and evaluation. This active learning strategy enables students to learn the following skills: solve real-life problems; identify suitable resources for problem solving; use effective self-directed and self-motivated learning skills; continuous monitoring and evaluation of the sufficiency of their own knowledge and problem-solving skills; teamwork, which helps students develop communication and leadership skills, social and ethical skills. The most prominent outward characteristics of active learning are student-centred tables and technology to facilitate the learning, such as displays, whiteboards, projectors and microphones. The focus of this paper is on the learning spaces - we share our experience in planning and equipping our learning spaces to teach software engineering. The first part of the paper introduces the different active learning strategies of software engineering. The second part, which is the key aim of this paper, analyses the key aspects & criteria required and different types of spaces available for an active learning lab. Finally, in the last section, we demonstrate our proposed learning spaces.

1.0 Introduction

Active learning is a process in which students participate in activities that encourage higher-order learning skills, including analysis, synthesis, and evaluation. According to Bonwell and Eison, this is the definition of strategies that stimulate active learning: “instructional activities involving students in doing things and thinking about what they are doing” (Bonwell and Eison, 1991). These strategies allow students to engage in the learning activities and at the same time construct their own knowledge and understanding – making meaning in what they do in the learning activities. Active learning approaches focus on developing students’ skills rather than transmitting information. The students are required do something such as reading, discussing, and writing; all these doings entail higher-order thinking. Active learning approaches also encourage the students to explore their own attitudes and values. This active learning strategy enables students to learn the following skills: solve real-life problems; identify suitable resources for problem solving; use effective self-directed and self-motivated learning skills; continuous monitoring and evaluation of the sufficiency of their own knowledge and problem-solving skills; teamwork, which helps students develop communication and leadership skills, social and ethical skills. There is a wide range of active learning strategies: it can be as simple as pausing during a lecture so that students have the opportunity to clarify and organise their

ideas; or it can be as complex as using real cases in the real-world environment. In active learning, learners are always engaged in activities that involve the use of their higher-order thinking in groups. Section 2.0 presents and discusses the common strategies of active learning used in the software engineering courses.

2.0 Active learning strategies in software engineering

The common strategies of active learning in the software engineering courses were found through a literature search. We searched through Web of Science (WoS) database and Google Scholar Search Engine. The keywords used were "Active Learning" and "Software Engineering". After the filtering process, we short-listed 49 papers presented in conferences or published in journals between the years 2005 - 2018. Table 1 shows the result of the search.

It was discovered that many strategies can be employed to conduct active learning in the software engineering subjects. The highly popular strategy is team and collaborative learning through projects or assignments; researchers who advocated or supported this strategy include Sibona, Pourreza, & Hill (2018), Tiwari, Saini, Singh, & Sureka (2018) and Marcos-Abed (2018). We noted that this approach was reported in more than 20 different articles. A moderately popular strategy is project which was reported by Díaz Redondo, Fernández Vilas, Pazos Arias, & Gil Solla, (2014); another moderately popular strategy is game-based learning which was reported by (Caulfield & Veal, 2011). These strategies were reported between 10 and 20 times in various articles. Here are some less popular active learning strategies: discussion (Tiwari, Saini, Singh, & Sureka, 2018), problem-based (Fonseca & Gomez, 2017), case-based (Hainey, Connolly, Stansfield, & Boyle, 2011), role play (Damian & Borici, 2012), peer to peer (Semushin, Tsyganova, Ugarov, & Afanasova, 2018) and others. These strategies were reported less than 10 times in various articles.

Based on the search results, some of the most popular subjects that apply active learning are Implementation / Development/ Programming (Fonseca & Gomez, 2017) and Software Engineering (Sedelmaier & Landes, 2015). Moderately popular subjects that use active learning are Project Management (Claypool & Claypool, 2005) and Requirement (Damian & Borici, 2012). The less popular subjects that utilise active learning are Design (Claypool & Claypool, 2005), Architecture and Modelling (Sedelmaier & Landes, 2015), Evaluation, Testing (Holmes, Allen, & Craig, 2018) and Configuration (Krusche, Reichart, Tolstoi, & Bruegge, 2016).

Table 1: Active Learning strategies in Software Engineering Subjects/ Topics

ACTIVE LEARNING **Total Search Document Used = 49** **Year of Paper Publication: 2005 - 2018**			
POPULARITY	**SE SUBJECT /TOPICS**	**Strategies**	**FREQUENCY**
High	**Implementation /programming/ development Software Engineering**	Team/ Collaborative	more than 20
Moderate	**Project Management Requirement**	Project-Based Game-Based	more and equals 10
Less	**Agile SE methodology Design** Architecture **/ Modelling Evaluation/Testing Configuration**	Discussion Problem-Based Learning Case-based learning Role play Peer-to-peer Pitching/ Presentation Learning Outside class Experiential Simulation Open learning (students sharing source code) Studio-based learning Laptop exercises Learning by doing Real Mentorship Design-based learning Brainstorming Review	less than 10

3. Research Approach

High-impact teaching and learning practices like the case-based, problem-based, project-based, integrated design and many other active learning require a new environment; the normal traditional classroom setting needs to be changed to a space more conducive to learning.

"Spaces are themselves agents for change. Changed spaces will change practice" (JISC, 2006)

Classrooms and libraries are learning spaces; common areas within a university campus have the potential to be part of the learning space too. There is a big body of knowledge about the effects of the classroom design and environment on student learning. Generally, the design of a good learning space has these implications: affects student motivation and promotes learning as an activity; supports collaborative learning, provides a mixture of personalised and inclusive environment; and is flexible to support reconfiguration in real-time and future-proof. The

demand for ROI (return on investment) of the capital expenditure for public buildings suggests that any investment in the learning space should support all the above-mentioned goals; and at the same time the buildings must be creatively designed to inspire both learners and teachers engaging in their respective tasks.

This section reviews the aspects which serve as a guideline in the design of active learning spaces.

3.1 Design for Active Learning Space - Basic Principles, Space types and Basic Criteria

In designing a learning space for active learning, the following basic principles need to be seriously considered (ADeC 1, 2016).

- Learning First - the space which enables learning activities to be carried out; as learners do, they learn. Connection to the information universe via the internet, though extremely important, needs to be balanced by its usefulness; that is turning the information into new knowledge. Hence, the space must be conducive to learning through conversation, discussion and brainstorming among students and lecturers.

- Flexible spaces to encourage innovation - spaces that can be reconfigured to accommodate various activities in various formats. Preferably, the furniture and equipment are of movable types.

- Support for collaborative work practices - Spaces that naturally bring students together to interact in a relaxed and non-threatening environment may speed up the collaborative work practices. Comfortable chairs, bean bags or small coffee tables may reduce learning tensions when students interact with one another in the learning activities.

- Facilitate capturing of information for archive and reflection - the immersive nature of active learning may override the reflection process, and thus reducing its impact. The space should have facilities to capture information for archive and reflection, for example, using smart board, and other digital devices or software that may help students to develop in-depth knowledge from their active learning experience.

There are several space types that can be chosen for formal and informal learning purposes. Examples include the following:-

a. Brainstorming Space – a collaborative space conducive to brain storming, often equipped with whiteboard and flipped charts to record ideas that will benefit everyone.
b. Learning commons - large spaces, centrally used for learning activities; for example, library books and computing and other facilities such as maker-space are widely accessible.
c. Research commons – a place to collaborate and connect with peers and the faculty staff when doing research. It incorporates presentation facilities so that research findings from the peers can be shared.

d. Connective Space - focusing on embedding technology in teaching and learning; hence, the space must be equipped with technology for connectivity via wired computer, wireless network and facilities for PDAs and mobile phone connection.

The criteria for new Learning Space development should be centred on maximising the potential of using these spaces to support collaborative/interactive learning and flipped classroom.

The following are the basic criteria for the new learning space (ADeC 2, 2016):

a. Stakeholder engagement
   Key stakeholder engagement during the planning of the project needs to be evidenced. These are the owners, users (both learners and educators), instructional designers, educational expert, design professionals, facility managers and other relevant parties.

b. Room requirements
   The room for learning space must be well lit (minimum 500 lux) and equipped with dimmer switches so that lights can be dimmed when the projection screen is being used. Room temperature must be controllable between 20° - 26° Celsius during the operational hours.

   Acoustics

   The room should incorporate sustainable energy usage and eventually have national or international sustainability certification.

c. Network performance
   The wired or wireless broadband connection must cover all users with a connection speed of at least 2Mbps/person with simultaneous usage. The network should be able to support the streaming of HD videos.

   There must be enough wireless access points so that all room occupants can access the internet simultaneously. The bidder should demonstrate upscaling capability for the network to support higher broadband speeds and telepresence in the future.

d. Furniture design
   Majority of tables and desks must be reconfigurable and easily moveable by a single person. Stackable furniture design is encouraged to facilitate storage and room reconfiguration. Integration of the furniture with power points location should be demonstrated.

e. Power requirements
   Power points must be provided to cater for at least 20% of the room occupants, apart from the power points for all the electrical equipment required during the formal learning sessions.

f. Monitors and projection devices

Connection to projection devices should be wireless-ready, accessible via the local wireless connection or Bluetooth.

g. User access
   Access to the learning spaces must be promoted and granted to all the authorised students of the faculty outside of formal learning sessions. Online booking for the rooms must be initiated and the use of the rooms logged and reported to the faculty management.

Taking into consideration all the above aspects, subject matter and context together, the proposed active learning spaces in this research are presented in Section 5.0.

5. The Proposed Design of an Active Learning Lab

Figure 1 shows the proposed design of an active learning space to be used for teaching software engineering courses for the Department of Software Engineering, Faculty of Computer Science and Information Technology, University Malaya. The learning space consists of the following:

- Colour use: Six different shades of Monochromatic Green and Blue (or any other colours) are used. These different colours symbolise six different groups and can be used as team names.
- Furniture: foldable and movable chairs, movable tables, and space under the table to put bags. Triangular tables are chosen as part of our designed furniture for space optimisation and closer interaction between group members.
- Power source: On all four sides of the walls for charging the battery of each piece of equipment.
- Walls: Writable, suitable area for pasting post-it notes as discussion points for brainstorming and other related activities
- Lines on the floor: As markers for positioning tables in 4 different types of arrangements
- Projector: Can project in 2 areas
- PC: a desktop can be placed on each table at one of the triangle's point, Movable PC position, multi-controlled.
- Maximum Capacity: 36 students

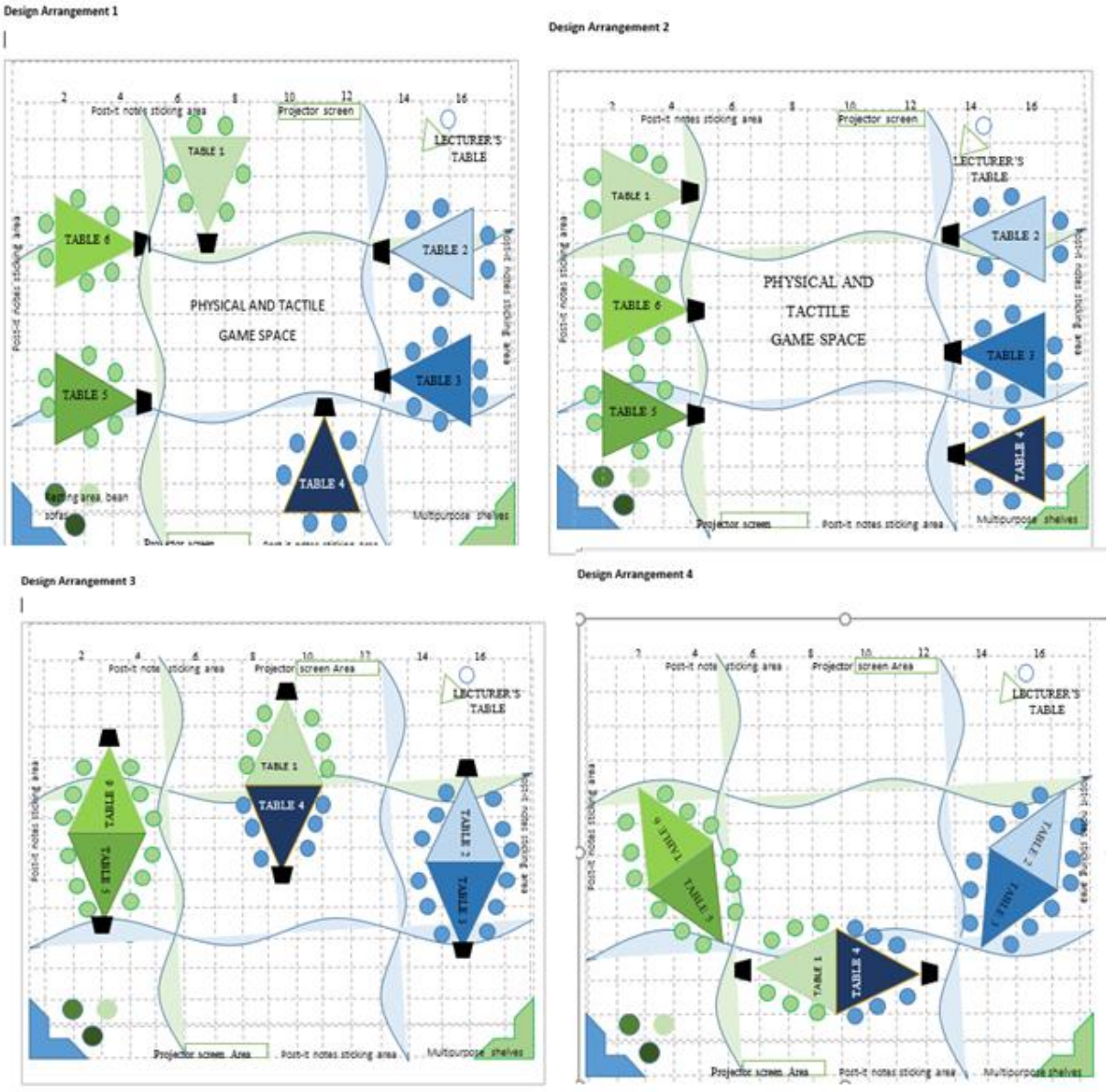

Figure 1: The proposed active learning space

There are four types of arrangements to support learning among students and lecturers; flexible and movable spacing can support a few types of collaborative work and facilitate capturing of information for archive and reflection. Our design includes brainstorming space, research commons and connecting space. Arrangements 1 and 2 are possible table arrangements for 6 people in a group. Arrangements 3 and 4 are possible table arrangements for 12 people in a group. The space in the middle of the room in Arrangements 2 and 4 can be used to conduct physical and competitive activities, which are a method adopted to teach software engineering subjects such as requirements, software architecture, human computer interaction, software testing, algorithm and programming. Other types of active learning such as project and game-based learning, problem-based learning, experiential learning are also possible with the table arrangement suggested. It is also possible to arrange the tables in a way other than the one suggested above. In principle, devices, power supply points and network requirements are needed as part of the room furniture.

6. Conclusion

The process of high-order thinking such as analysis, synthesis and evaluation can be instilled through active learning by engaging the learners with the lesson taught. Through our literature findings, the most popular active learning strategy for software engineering classes is team or collaborative activities, followed by project and game-based method. Other techniques include peer-to-peer, pitching, presentation, experiential learning, etc.

The principles used for our design of learning space is learning first, flexible spaces, supporting collaborative work and facilitating the capturing of information for archive and reflection. Devices, power and good network performance are essential parts of the requirement to construct the designed space.

Therefore, a collaborative designed space with 4 different arrangements was used as the basis of our design. This is because one of the most popular software engineering active learning strategies is indeed collaborative and project-based learning.

Our future study would include how the designed space is used in the real-time classroom. Data gathered would be used to analyse students' acceptance of the designed space and how efficient the different active learning strategies are conducted in the designed space.

Acknowledgement

We would like to thank Erasmus+ for giving us the grant with no: 586297-EPP-1-2017-EL-EPPKA2-CBHE-JP (2017 -3629 / 001 – 001).

References

Academic Development Centre (ADeC 1) (2016). UM Learning Space Grant; Project paper

Academic Development Centre (ADeC 2) (2016). Criteria for UM's Learning Space.

Bonwell, C. C., & Eison, J. A. (1991). *Association for the Study of Higher Education.; ERIC Clearinghouse on Higher Education*. Retrieved from https://files.eric.ed.gov/fulltext/ED336049.pdf

Caulfield, C., & Veal, D. (2011). Artigo-A systematic survey of games used for software engineering education. http://doi.org/10.5539/mas.v5n6p00

Claypool, K., & Claypool, M. (2005). Teaching software engineering through game design. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education - ITiCSE '05*, 123. http://doi.org/10.1145/1067445.1067482

Damian, D., & Borici, A. (2012). Teamwork, Coordination and Customer Relationship Management Skills: As Important as Technical Skills in Preparing Our SE Graduates. In *Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences (EduRex '12)*. Retrieved from http://ezproxy.um.edu.my:2123/10.1145/2670000/2663688/p37-

damian.pdf?ip=103.18.0.17&id=2663688&acc=ACTIVE SERVICE&key=69AF3716A20387ED.E7759EC8BE158239.4D4702B0C3E38B35.4D4702B0C3E38B35&__acm__=1530551549_b0a44ef3e371ec1dcc3b437a1303b29d#URLTOKEN#

Díaz Redondo, R. P., Fernández Vilas, A., Pazos Arias, J. J., & Gil Solla, A. (2014). Collaborative and role-play strategies in software engineering learning with web 2.0 tools. *Computer Applications in Engineering Education*, *22*(4), 658–668. http://doi.org/10.1002/cae.21557

Fonseca, V. M. F., & Gomez, J. (2017). Applying Active Methodologies for Teaching Software Engineering in Computer Engineering. *IEEE Revista Iberoamericana de Tecnologias Del Aprendizaje*, *12*(3), 147–155. http://doi.org/10.1109/RITA.2017.2738178

Holmes, R., Allen, M., & Craig, M. (2018). Dimensions of Experientialism for Software Engineering Education. *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '18)*, *9*. http://doi.org/10.1145/3183377.3183380

Joint Information Systems Committee (JISC) (2006), Designing Space for Effective Learning:A Guide to 21st Century Learning Space Design, p. 30, <http://www.jisc.ac.uk/uploaded_documents/JISClearningspaces.pdf>.

Krusche, S., Reichart, B., Tolstoi, P., & Bruegge, B. (2016). Experiences from an Experiential Learning Course on Games Development. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. http://doi.org/10.1145/2839509.2844599

Les Watson, pro vice-chancellor, Glasgow Caledonian University, quoted in Joint

Information Systems Committee (JISC), Designing Space for Effective Learning: A Guide to 21st Century Learning Space Design (Bristol, U.K.: Higher Education Funding Council for England, 2006), p. 24, <http://www.jisc.ac.uk/uploaded_documents/JISClearningspaces.pdf>.

Marcos-Abed, J. (2018). Poster: A Matrix for Analyzing Projects in Software Engineering Courses. In *Proceedings of the 40th International Conference on Software Engineering (ICSE '18)*. Retrieved from http://ezproxy.um.edu.my:2123/10.1145/3200000/3195053/p139-marcos-abed.pdf?ip=103.18.0.17&id=3195053&acc=OPEN&key=69AF3716A20387ED.E7759EC8BE158239.4D4702B0C3E38B35.6D218144511F3437&__acm__=1530550984_fab139ed7cf6b3e31bd5972fe02741c5#URLTOKEN#

Sedelmaier, Y., & Landes, D. (2015). Active and Inductive Learning in Software Engineering Education. In *2015 IEEE/ACM 37th IEEE International Conference on*

Software Engineering (Vol. 5, pp. 418–427). IEEE. http://doi.org/10.1109/ICSE.2015.174

Semushin, I. V., Tsyganova, J. V., Ugarov, V. V., & Afanasova, A. I. (2018). The WHATs and HOWs of maturing computational and software engineering skills in Russian higher education institutions. *European Journal of Engineering Education*, *43*(3), 446–472. http://doi.org/10.1080/03043797.2017.1385594

Sibona, C., Pourreza, S., & Hill, S. (2018). Origami: An Active Learning Exercise for Scrum Project Management.: EBSCOhost. *Journal of Information Systems Education*, *29*(2), 105–116. Retrieved from http://ezproxy.um.edu.my:2664/ehost/pdfviewer/pdfviewer?vid=9&sid=834670c5-08d0-4828-8e32-455b085f54ff%40sessionmgr4006

Tiwari, S., Saini, V., Singh, P., & Sureka, A. (2018). A Case Study on the Application of Case-Based Learning in Sooware Testing. *Proceedings of the 11th Innovations in Software Engineering Conference (ISEC '18)*. http://doi.org/10.1145/3172871.3172881