INFORMATION SYSTEM DEVELOPMENT SUBJECT: EDUCATION VERSUS INDUSTRY

Nor Zalina Ismail*, Mohd Rizal Razak, Muhammad Qamarul ‘Ashraf Abd Hilmi, Mohamad Izzudin Amran, Ahmad Fareez Mohamad Rusdi, Muhammad Syazwan Ismail

Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA UiTM Pahang, 27600 Raub

**Corresponding author: *nza1601@uitm.edu.my*

Abstract

Our education curriculum should be tailored to the industry practice. Industry always utilize the open source code to develop their software and many of the project in the industry involve updating or maintaining the existing system. Industry also applies the teamwork in their software development project. Nowadays, they also implement collaboration team which is the team member working virtually in different location. This is contrast with the curriculum for Information System Development(ITS332) that require students to develop new real-world computer software. Sport Venue Booking System is the information system project develop by Semester 4 Diploma in Computer Sciences students for ITS 332 subject. The objective of this research is to compare the software development in industry practice versus academic practice by using Sport Venue Booking System as a case study. This research will also compare the practice in SVBS with other higher academic institution. The methodology used is through empirical study in industry practice and other higher academic institution versus the real experience of the supervisor that supervise the project. This research is important to make sure students can be competent in real working environment. Based on the result, there is some lack of the program comprehension skill and collaboration skill necessary for the students to survive in IT world.

Keywords: Education, Industry, Information System Development

Introduction

Diploma in Computer Sciences (CS) students should have a knowledge and skill on software development project as a preparation for working environment after graduate or to further their study. As to fulfil this requirement CS student must pass the Information System Development (ITS 332) subject. Students that take this subject must undergo the software development group project in order to obtain 40% of the assessment mark. The project title is decided by the students through the discussion and consultation session with the supervisor appointed by the Faculty of Computer and Mathematical Sciences, UiTM Raub. They will develop the system from scratch. Sport Venue Booking System (SVBS) is one of the projects developed by the students that take the ITS332 subject. This system will become the case study for this paper.
Program comprehension skill is an important skill for the graduate students need by the industry. Program comprehension mean the skill to understand the existing source code. The reason why students need this skill because industry always take an advantage of open source code to develop their information system. The second reason; program comprehension skill is very demanding skill to fix or to upgrade the existing software used by the company. ITS332 subject does not implement any system using source code or enhance any existing system.

First objective of this paper is to compare and analyse the implementation of Sport Venue Booking System (SVBS) project with the recent industry approach. Then, second objective of this paper is to know and analyse the implementation of SVBS with other higher institution practice. This is important to make sure the approach used for ITS332 group project is relevant to become the mechanism for students gain the knowledge and experience in software development project that mimic the industry approach and become the fundamentals for the graduate students to jump into real working environment. In other words, this research is to confirm either the implementation of the group project in ITS332 is aligned with the industry need.

Sport Venue Booking System (SVBS)

This system is develop for Facilities Department, Universiti Teknologi MARA Cawangan Pahang Kampus Raub. The objective of this system is to convert the manual booking system into computerised booking system. This system was develop to reduce or minimise the problem occur when using the manual booking system (manual form using paper based and planner) such as difficult to keep track the booking record and the time and day available for booking. UiTM Raub has 4 sport venues; a football field, a tennis court, a volleyball court and a basketball court. This limited sport facilities has been hit to used by UiTM Raub and more than 2500 UiTM Raub students. So, it is important to have an efficient booking system. **Figure 1** shows the main menu for SVBS.

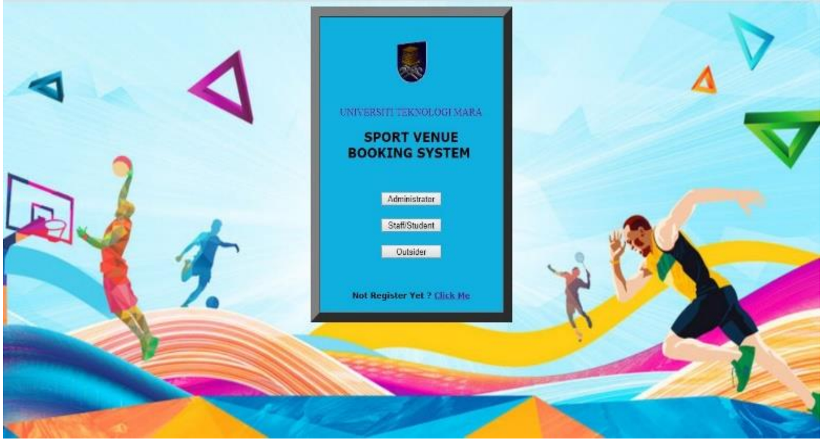
UNIVERSITI TEKNOLOGI MARA
SPORT VENUE BOOKING SYSTEM
Administrator
Staff/Student
Outsider
Not Register Yet ? Click Me

Figure 1 Sport Venue Booking System Main Menu

This online venue booking system is meant for students, staff,system administrator and also outsider that written using a programming language named hypertext markup language (HTML) and hypertext preprocessor (PHP). As an addition, Xammp is used as a database.

Research Methodology

This research is done by compare and analyse the Sport Venue Booking System supervisor's experience with the literature review. The literature review is divided into two main categories which is to find the literature on industry approach for software development process. Second category is to find the literature on other higher institution practice.

Findings and Result

The findings and result segment is divided into two part that based on the research objective define in introduction segment.

Sport Venue Booking System (SVBS) versus Industry Approach

i) Teamwork

Table 1 SVBS Versus Industry(number of team members)

SVBS	**Industry**
Small number of team members	Large number of team members

It is good that SVBS develop by four person in a group but, industry do a software engineering task in large number of team members (Dings & Dyba, 2008). As stated by Heckman et al., (2018), working in team project is close to industry practice. Industry working in large number of team members by considering the complexity of the project. In addition, if the large number of team members involve in SVBS, it may lead to the higher number of sleeping partners in the group and will decline the learning objectives. Based on the research done by Dunsmore et al., (1989), the number of 4-5 person in group is ideal for one semester group project and can avoid the "parasite" in the group.

ii) Physical location of team members

Table 2 SVBS Versus Industry (physical location)

SVBS	**Industry**
Situated in same location/area	Situated in distributed physical (different location and different geographical area) (Wagner & Ruhe, 2018)
Involve many face to face meeting	Involve many virtual meeting (tele conferencing) (Wagner & Ruhe, 2018)
Does not have established telecommunication facilities	Have established telecommunication facilities (Wagner & Ruhe, 2018)

Based on **Table 2**, it shows that recently, industry implementing the software engineering task by involving the virtual team that work in different location and area. But, they have a good telecommunication facilities to make sure the communication between team members is well establish. More over, they have good software tools that make sure the collaboration or integration of distributed task can work well in order to make sure the project is success.

In opposition with the SVBS, they are working in same area and same location. But, for the purpose of the mimic on the industrial experience, the lecturers can introduce some effective communication mechanism and software development tools that can gather and integrate the task that distributed to the team members. This can be used by utilize the cloud computing resources. As stated by Bouyer & Arasteh (2014), the applications and files stored in the cloud can be access by the members through internet. In order to integrate or collaborate the task, each of the team members should understand the task or source code done by others. This situation can increase the effectivenes of learning process and reduce the sleeping partner percentage.

One more important thing that industrial approach is differ with the SVBS is they using the open source code for their software development project. They customize the existing open source to tailor their needs. Certain higher education institution such as University of Carolina used the open source for their student's project. By using the open source code students not only can learn to do a coding but also can learn to comprehend the source code done by others. It is better if the ITS 332 subject require the students to have an open source code in their software development process. This can increase their programming skill and also the confident level in future working environment.

Last point, industry need the person that can work with the existing code written by other programmer. In ITS332, all the project is come out from the new software development project. Both software engineering and computer sciences students is only emphasize on developing new software (Cornelius et al., 1989). But, in industry it is different because the software written in the industry must be maintain, upgrade or enhance according to the new requirement (Cornelius et al., 1989). So, it is recommended for ITS332 project to promote program comprehension skill by using some part of open source code in their project or allow the student to maintain or upgrade an existing system.

Other Higher Institution Practice

Table 3 shows the comparison between SVBS and other university (North Carolina University, Durham University). It shows the comparison of team members, duration of the project and type of project given to the students.

Table 3 Comparison between SVBS and Other University Approach

Name of university	Number of team members	Duration	Type of Project
SVBS (UiTM)	4 person per group	About 5 to 6 weeks	New project, start from scratch Student and supervisor define the new project
North Carolina University	Four to 5 students	6 weeks	Improving legacy system named iTrust (electronic health records) The instructor/lecturer give the list of enhancement to be done.
University of Durham	Small number of team members (does not mention the exact number)	One term (but divide the task into 2 phase- maintenance workshop and maintenance task)	Given the existing system to be maintain or enhance The instructor give the list of maintenance or enhancement to be done
University of Adelaide	Small number of team members (does not mention the exact number)	4 weeks	Given the existing system to be maintain or enhance The instructor give the list of maintenance or enhancement to be done

University of North Carolina	3 to 4	Second half of semester	Use the open source code named Expertiza that available from GitHub The ideas of the project is "come from reported bugs, feedback of users, rejected projects from previous semesters, and new features requested by instructors using Expertiza"

Table 3 shows North Carolina University used the legacy system that develop by students team over 25 years (Heckman et al., 2018). Feedback from the graduate students, they appreciate the process of working with the legacy system because company confident to hire the students from this university (Heckman et al., 2018). Not many new programmer can do an enhancement of the existing program, it is more difficult than working with the new system from the scratch.

University of Durham also apply the same concept with North Carolina University. They believe in preparing the students for working environment, they must have the capability to do an amendment to the existing code. Cornelius et al. (1989) stated that an educator only focus on software development stage in software development life cycle but does not emphasize on the software maintenance task. But, in the industry, software maintenance is the most expensive phase. In addition, University of Adelaide emphasize the program comprehension task in order to do software maintenance task (Szabo, 2015). They believe the software maintenance task that involve program comprehension can help increase the students programming skill.

Another way the Computer Sciences or Software Engineering educator improvise the teaching and learning process in order to align with the industry needs and approach is the adaptation of open source software (OSS) projects. University of North Carolina used the OSS project in their teaching and learning process that will give an advantage to the students "to learn good coding practices from real world projects" (Hu & Gehringer, 2018).

Conclusion and Recommendations

Based on the literature review stated in findings and result section, shows the criteria that aligned with industry practice and the criteria can be improve in order to prepare the students for working environment.

Criteria of SVBS that aligned with industry practice:

a) SVBS project is done in team that aligned with the industry needs and other university practice. This can give an advantage to the students to experience real world environment project nature and also enhance the learning process.

b) SVBS project let the students experience the real-world software development project. It is good experience because students can practice what they learn in class and can understand the nature of the programmer job.

Criteria of SVBS can be improve in order to align with industry practice:

a) Introduce effective communication facilities to collabarate the task through the internet such as wiki(Kinsey, J., & Carrozzino, A. L. ,2011) and sofware tools that can collaborate the coding done or the changes done by the team members such as Travis CI2 and Code Climate3(Hu & Gehringer, 2018). This can mimic the distributed team members approach that become a practice of the industry nowadays.

b) Introduce sofware maintenance task or open source software in the project because it is the industry approach. Many universities create a project using software maintenance task(Heckman et al., 2018; Cornelius et al., 1989; Szabo, 2015) or open source software task(Hu & Gehringer, 2018).

Acknowledgement

This research was supported by UiTM Pahang Branch for publication. We thank all the ITS 332 lecturers who provided insight and expertise that greatly assisted the research. We would also like to show our gratitude to all the reviewers for their comments that greatly improved the paper.

Conflict of interests

Author declares no conflict of interest.

References

Bouyer, & Arasteh. (2014). The Necessity of Using Cloud Computing in Educational System. *Procedia - Social and Behavioral Sciences*, *143*, 581–585. https://doi.org/10.1016/j.sbspro.2014.07.440

Cornelius, B. J., M, M., Robson, D. J., & Munro, M. (1989). an Approach To Software Maintenance Education. *Software Engineering Journal*, *4*(4), 233–236. https://doi.org/10.1049/sej.1989.0030

Dings, T. & Dyba, T. (2008). Empirical studies of agile software development : A systematic review. *Information and Software Technology*, *50*, 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Dunsmore, H. E., Moffet, D. P., & Ward, S. T. (1989). Software engineering team project group member evaluations: Some empirical results. *ACM SIGCSE Bulletin*, *21*(2), 40–45. https://doi.org/10.1145/65738.65745

Heckman, S., Stolee, K. T., & Parnin, C. (2018). 10+ Years of Teaching Software Engineering With Itrust. In *Proceedings of the 40th International Conference on Software Engineering Software Engineering Education and Training - ICSE-SEET '18* (pp. 1–4). https://doi.org/10.1145/3183377.3183393

Hu, Z. & Gehringer, E. F. (2018). Open-Source Software in Class : Students ' Common Mistakes. In *ACM/IEEE 40th International Conference on Software Engineering:Software Engineering Education and Training* (pp. 40–48). https://doi.org/10.1145/3183377.3183394

Szabo, C. (2015). Novice Code Understanding Strategies during a Software Maintenance Assignment. *Proceedings - International Conference on Software Engineering*, *2*, 276–284. https://doi.org/10.1109/ICSE.2015.341

Wagner, S. & Ruhe, M. (2018). A Systematic Review of Productivity Factors in Software Development. *ArXiv Preprint ArXiv:1801.06475.* Retrieved from http://arxiv.org/abs/1801.06475