Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Incorporating teamwork in waterfall model-based project

Noorihan Abdul Rahman [1, *], Nik Nahdiya Nik Kamaruzaman [1], Wan Salfarina Wan Husain [1], Siti Hasrinafasya Che Hassan [1], Rozianiwati Yusof [2]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Kelantan, Bukit Ilmu, Machang, Kelantan, Malaysia
[2]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Negeri Sembilan Kampus Seremban 3, 70300 Seremban 3, Negeri Sembilan, Malaysia

ARTICLE INFO

Article history:
Received 10 February 2018
Received in revised form
20 October 2018
Accepted 22 October 2018

Keywords:
Assessment
Development
Information system
Teamwork skill
Waterfall model

ABSTRACT

Developing quality information system is always an issue in the IT industry since quality information system is also affected by the values contributed by each team members. In educating university students to develop satisfied information system, the educators need to educate them in building teamwork throughout their development project. This study depicted how university students have used Waterfall model for developing quality information system. The students are required to accomplish their development project according to the required syllabus. The project is considered as their assessment and marks are given to them accordingly. Besides obtaining their assessment result, this study is also concern in looking at their teamwork level. This paper also reports on how the students anticipate within their own team in accomplishing their software project by using Waterfall Model. The study on students' teamwork has been conducted to analyze teamwork criteria namely shared leadership, team orientation, redundancy, learning and autonomy. Data gathered for this study is collected among students who are doing Computer Science-related project by using online survey. The result of this study has found out that students build up their leadership and improve their teamwork skill and hence gain team members' trust after accomplishing software project.

1. Introduction

In a system development, defining software project team is a critical task for ensuring effective project flow (Richardson et al., 2012). With the knowledge of selected software methodology, a software project team gives their effort to accomplish software projects which satisfy users' requirements with minimum software error. Software methodologies comprise of traditional method, and agile method (Leau et al., 2012). In this paper, traditional software methodology which is widely known as System Development Life Cycle (SDLC) or Waterfall model will be used as the software methodology by the final year students from Diploma in Computer Science in order to develop high quality information system. Waterfall model is developed by Royce 1970 and it is called as 'Waterfall' since the pattern looks like cascade of waterfalls (Kumar and Bhatia, 2014). This model is taught to final year students of Diploma in Computer Science as the foundation of developing an information system in sequence process starting from system planning, system analysis, system design, system implementation and system security and support.

Assessing students' teamwork in CS project is not a simple task (Fidalgo-Blanco et al., 2015) since the students need to be measured with a set of criteria to analyze their teamwork level. This paper is particularly focuses on teamwork assessment based on shared leadership, team orientation, redundancy, learning and autonomy. Specified literature survey has been conducted in order to thoroughly investigate mentioned criteria. The method of assessing students' teamwork is based on online survey whereby students answer a list of questionnaires which are related to those criteria. Teamwork assessment is conducted by CS lecturers to measure the teamwork quality and also to monitor group cohesion among team members. By looking at the assessment result, the lecturer is able to identify the teamwork skill among CS students for diploma level. This paper embraces four sections to elaborate more on the topic. Section 1 gives the introduction of the paper. Section 2 elucidates

* Corresponding Author.
Email Address: noorihan@kelantan.uitm.edu.my (N. A. Rahman)
https://doi.org/10.21833/ijaas.2018.12.015
2313-626X/© 2018 The Authors. Published by IASE.

teamwork in software development project. Section 3 reviews about student teamwork. Section 4 reports on research design of this study. Section 5 elaborates results on and discussion whereas the last section concludes the paper.

2. Teamwork in software project using waterfall model

Software development is one of the most difficult and time consuming task (Ghuman, 2013). A computer is useless without the software, so software development activity require concerted effort and team work of a group of people (Ghuman, 2013). To guide the people in developing software, many models available, however, Waterfall model or known as linear sequential life cycle model is an essential since it is a foundation for software development purpose. Waterfall model is a classic traditional model since it helps developers to understand required step to go through for accomplishing a development. This software methodology is time consuming, nevertheless, it helps to identify an accurate set of requirements and the requirements will be bring into the development phase. Indirectly, this method guides the developers to initiate proper procedure before designing and implementing source codes. The stages of waterfall model consist of five stages. The stages are system planning, system analysis, system design, system implementation and system operation and support. Each phase must be completed before next one starts. At the end of each stage, the project is reviewed to ensure compliance with requirements (Stoica et al., 2013).

According to Fig. 1, it shows documentations needed in order to accomplish Computer Science-related by teamwork.

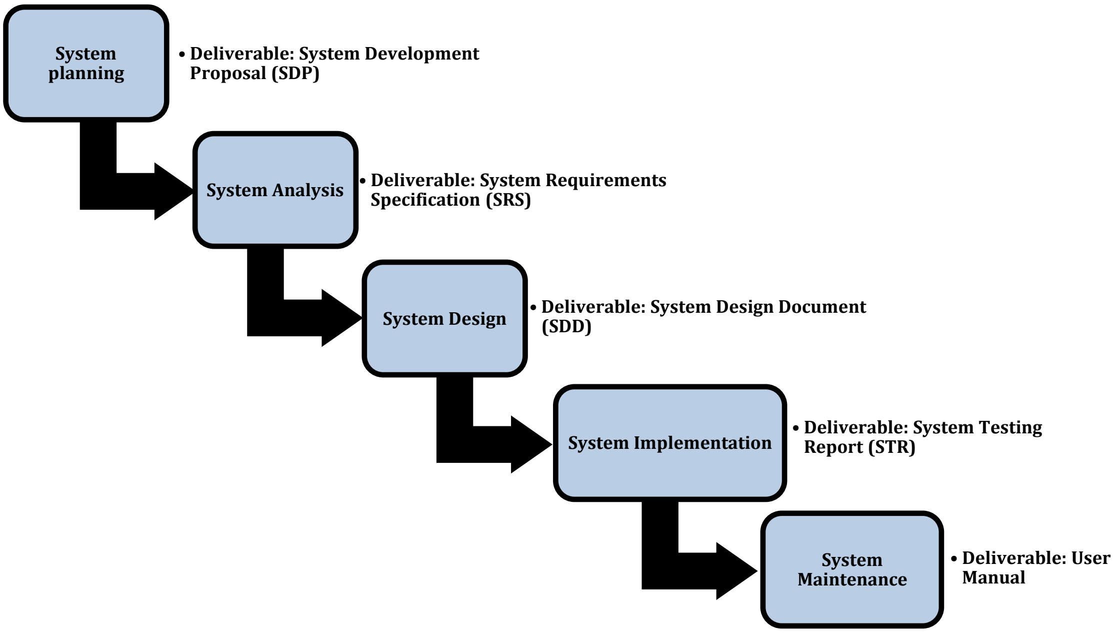
System planning
• Deliverable: System Development Proposal (SDP)
System Analysis
• Deliverable: System Requirements Specification (SRS)
System Design
• Deliverable: System Design Document (SDD)
System Implementation
• Deliverable: System Testing Report (STR)
System Maintenance
• Deliverable: User Manual

Fig. 1: The Requirements of documentation based on waterfall model (Kumar and Bhatia, 2014)

2.1. Development of system development proposal

Firstly, System Development Proposal (SDP) captures in a product requirements document containing all the requirements to a certain project so that, it allows people to understand what a project should do. In a teamwork project, there will be a project manager who plays a primary role in coordinating and controlling of all team members. To initiate the development of the project, a blueprint document which is known as Software Development Proposal (SDP) is presented. In this SDP, a project manager will discuss with his teamwork members in describing the software development activities such as determining the methodology used in the project as well as establishing each phase in software project development life cycle (SDLC) such as system planning, system analysis, system design, system implementation and system maintenance based on a project schedule. Besides, in this documentation each individual project team members will be assigned to the specific roles with the diverse range of skills that involve in one or more phases in the SDLC. Therefore, cooperation and teamwork among the project team members are crucially important in contributing to the overall project objectives.

Furthermore, before the system is developed and implemented further, it is responsibility of a system analyst to ensure that the user requirements and expectations of the system are captured and documented correctly in a SDP. To achieve this goal, it will involve analyzing and understanding the existing system and its current problems in order to ensure that the needs and requirements of user are represented correctly in the finalized project and also can understood by the users as well as the project team members.

2.2. Development of system requirements specification

To follow the waterfall model, students proceeds from one phase to the next in a purely sequential manner. For example, student first completes Software Requirements Specification (SRS), which describes the requirements of functionalities, performances, interfaces and etc. that compliance to customer needs.

Through the SRS stage, the students interact to each other's and freely exchange ideas about requirements functionalities such as requirements description, safety and security of the project and develop more common understanding of problem for their project. Moreover, team members then conduct the brainstorming session in highlighting the necessary of features for proposed system according to SDP that they conduct in previous stage. With the brainstorming from the team members, it will reduce uncertainty about the project such as software functionality and performance as well as the time and budget constraints of the project. This stage also will help the team members to do the project according to planning project milestones and deadlines.

2.3. Development of system design document

When the requirements are fully completed, student proceeds to design a blueprint which is known as System Design Document (SDD) for students in teamwork to follow as the requirements given. At this stage, student that given responsibility as project manager will instruct the team member that plays a role as user interface designer to discuss with other team members on how to design the storyboard of the proposed system. The storyboard portrays each interface of the system start from the main interface and from this storyboard; it will show the flow of the system.

Apart from that, project manager will ask the team member that plays a role as database designer to hold a discussion for designing the Data Flow Diagrams which comprise of Context Diagram and Diagram 0. These diagrams will show the views of each component and how the processes circulate within the system. The students are also required to draw Entity Relationship Diagram and from here the team members should properly identify the main entities, attributes and the relationship between those entities so that it reflects the overall system. In creating SDD, the responsibility falls under interface designer and database designer but they need to cooperate with other team members in getting the ideas and captures the related information with the design.

2.4. Development of system testing report

After the development phase completed, testing phase starts and in this phase, all the components that have been developed were tested to make sure the components are working as expected. In waterfall model, the testing process is often too little or too late in the life cycle. However, with the team's work project, the students able to define some error in effectively with the proper division task and role while they are doing testing on their software. The leader of the team will take full responsibilities to make sure that all team members have all the necessary resources to execute the testing activities. Then the report of testing activities will be produced. For this phase, all the team members should know the user's requirement for the software to fulfill the entire requirement

During system implementation phase, students need to plan possible values of inputs to exercise paths through the code and determine the appropriate outputs for the software. Each member in the team project takes part in checking the functionality of each module in the system. They will test every single line of code by using different scenarios to make sure the system can receive any possible input and produce correct output. If it cannot accept that output, each of them will report the errors happened. More collaboration and discussion happened after integration testing is initiated whereby each module in the information system is integrated with other modules. They must define whether all the modules are communicating with each other or not. In order to make the process successful, the have to repeat the process whereby each of them will use different test data to validate the test process on system. The testing report by each team member will be integrating and will be analyzed. They have to collaborate in finding the possible solution of the problems reported in report. With this collaboration, they can focus a small enough area of the code and can be more thorough to guarantee it works as expected. This method of test design can uncover many errors or problems; it might not detect unimplemented parts of the specification or missing requirements. It is important to test the functionality of software according to the requirement. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output either is the same as the expected value specified in the test case or otherwise. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. The students in a team having a brainstorm session to list all possible positive test scenario to check whether a system processes them correctly. After that, they list all possible negative test scenario to verify that system is able to detect the negative scenario. All the team members will determine expected outputs for all those input. With the collaboration between all the team members, it will ensure the system that has been developed will be function very well since the testing process is conducted in different perspectives.

When the design is fully completed, an implementation of that design is made by students as project team members. Towards the later stages of this implementation phase, disparate software components produced by different teams are integrated. After the implementation and integration phases are complete, the software product is tested and debugged; any faults introduced in earlier phases are removed here. The results of test performed are documented in System Testing Report (STR). Then, the software project is installed, and a user manual is documented to introduce the step by step of instructions on using the system. Thus, the waterfall model maintains that student should move to a phase only when the preceding phase is completed and perfected. Phases of development in the waterfall model are discrete, and there is no jumping back and forth or overlap between them.

In order to make sure the instructions are easy to read and are understandable by all users, students will have a brainstorm session in designing the user manual. They have to provide step-by-step sequences in the correct order and explain what a function is for in software. They have to test that manual whether it is easy to work through the task from start to finish, easy breaking out of task and getting back in, or it is easy to jump into the user manual half way through a task.

3. Student teamwork: Shared leadership, team orientation, redundancy, learning and autonomy

Teamwork skills are taught and refined through university courses and most Computer Science programs rely on outside sources of instruction, assuming that students learn these skills elsewhere. The teamwork assessment could give them opportunity to apply their teamwork skills to relevant problems in the field (Lingard, 2010). In conducting a Computer Science project, each group of students needs to identify their project members and also decide their project title. Each project which consists of four or five members per group need to accomplish five documentations that carry 20% and project report with demonstration that worth another 20%. The documentations highlighted in this project include System Development Proposal (SDP), System Requirement Specification (SRS), System Development Design (SDD), System Testing Report (STR) and User Manual.

In order to see the impact of team cooperation, each students need to evaluate themselves based on shared leadership, team orientation, redundancy, learning and autonomy (Poženel, 2013). Fig. 2 visualizes how the criteria of students' teamwork assessment are measured on project-based and documentation of software project.

3.1. Shared leadership

Shared leadership is a characteristic of teams that forms when leadership behavior is accomplish by multiple team members (Day et al., 2006). It occurs when leadership roles and responsibilities are shared among team members rather than taken by a single designated leader (Liu et al., 2014). This type of leadership entails a dynamic, interactive influence process among individuals in groups which the purpose is to lead one another to the achievement of group (Nicolaides et al., 2014). Furthermore, Liu et al. (2014) emphasized that shared leadership could influences the overall team's behavioral outcomes as well as individual members' perceptions, interactions and learning within team and some of the recent studies clarify that shared leadership was more influential than traditional leadership for team performance. By having shared leadership among team members of software development project, the team members could assist others when needed and should take the other team member's contributions into consideration (Lindsjørn et al., 2016). Hence, each of the team members will have a clear vision of project's assessment and this could lead to effectiveness of the team (Ensley et al., 2006).

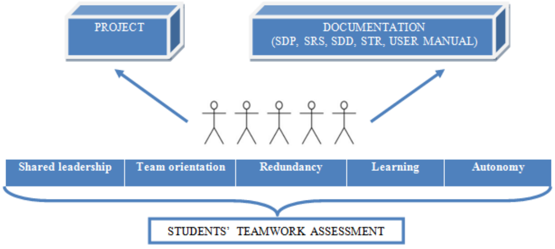

Fig. 2: The Inter-relation between project, documentation and student teamwork assessment measurement

3.2. Team orientation

Salas et al. (2005) define a team as consisting of two or more individuals who have specific roles, perform interdependent tasks, are adaptable, and share a common goal. Nowadays, teamwork orientation is considered as a standard practice in project-based learning to manage various requirements and tasks, in fact, almost every stage in project development is too complicated to be handled by just one person. Therefore, it seems that teamwork orientation in software development project could measure how well students can perform after they have an opportunity to learn from others in the context of collaboration. Through collaborative interaction, team members can freely exchange ideas and interpretations and develop a more common understanding of problem (Hogl and Parboteeah, 2007). Moreover, team members with an orientation toward teamwork must possess specific knowledge, skills, and attitudes (KSAs), such as the ability to exchange information, which enable individual team members to coordinate, support one's teammates, objectives, and mission (Salas et al., 2005). According to (Kay et al., 2006) team orientation propensities to take others' behavior into account during group interaction and belief in importance of team goal over individual members' goals. The goal setting works to reduce uncertainty about the qualitative properties of the project such as software functionality, robustness, and performance as well as the time and budget constraints of the project (Hogl and Parboteeah, 2003) as well as aids them in planning project milestones and deadlines.

3.3. Redundancy

Redundancy is considered as one of the criteria to measure students' teamwork assessment for CS's student project. Redundancy means the member is able to help other members with alternative pathways such as possess a wide array of abilities, skills and knowledge (Truong et al., 2015) whenever any problem encounters in accomplishing project task. In this case, the students are able to help each other whenever they have problems during the system development process (Chen et al., 2013). This shifting of work responsibilities to under -utilized team members so that the completion of the whole task or parts of tasks by other team members (Baker et al., 2005) can be done. This criterion also leads to cooperative environment among the members as it will increase teamwork skill for the project (Fuggetta, 2000).

3.4. Learning

Teamwork is used as a means for learning at all levels in educational systems, from early education level to higher education level. In higher education level, it is considered as important elements that needed in software development project. The purpose of teamwork in educational practice is to serve as an incentive for teach (Chiriac, 2014). For example, it is believed that during the teamwork activity in Computer Science Project using SDLC methodology, the students should "learn something". The ability to learn for each people are different depends on their limit. Many of our learning processes occur randomly throughout in our life, from new experiences, searching for information and also from studies. Researchers concluded that group work contributes to encouraging interest in the topic, facilitating resourcefulness and interdependence in learning, and developing skills such as problem identification and analysis, the exploration of solutions, teamwork, interpersonal communication and project managemen (Payne et al., 2006). American psychologist and educational reformer John Dewey was an early proponent of project collaboration and group work, believing that it was the learning process that was the key element in education rather than just the resulting knowledge set (Labeouf et al., 2016). From graduate students of view, they mentioned that collaboration, brain storming, sharing the wealth, and learning everyone's knowledge has been a great experience for teamwork (Labeouf et al., 2016).

3.5. Autonomy

Autonomy is a dimension that describes the extent to which the team is autonomous in its work and decision-making (Truong et al., 2015). For autonomy, the assessment result involve information in determining whether the team member can coordinate their time and manage their time properly in doing the task (Highsmith, 2013). The project leader establishes the tasks the team must perform, but in each iteration the team is free to decide how to work, with the goal of increasing team productivity (Stoica et al., 2013). These criteria also evaluated influence on team decisions in the project (Weimann et al., 2013). In other words, it can be defined as the ability to avoid or avert the influence of outsiders of other teams. However, it does not mean that the team will not take in feedbacks and advices from the outsiders to avoid dilemma like group-thinking (Truong et al., 2014). Autonomy is also useful to see students' opinions on whether they are respected by people and groups outside the team.

4. Research design

This study has carried out a quantitative method for obtaining assessment result on teamwork skills among CS students. This paper depicts the research design based on phases, activities, techniques and deliverables. Fig. 3 visualizes the research design of this study for accomplishing teamwork skill assessment.

4.1. Phases

Initially, an empirical research design starts with project planning and continued with data collection and analysis. In project planning, the authors identified classes which involved with Computer Science-related project. Projects that require teamwork would be an essential element during planning for assessing their progress during later stage in this study. Data collection and analysis is important for obtaining number of classes and groups of students involved in the Computer Science-related project.

4.2. Activities

A feasibility study has been carried out as one of the activities in this study. This study is conducted to identify the advantages and disadvantages if the Waterfall model is imposed as a software methodology. During the feasibility study, the authors will identify and list the syllabus content which implied in this study in order to initiate system development task in Waterfall model. The feasibility study is also conducted for justifying reasons for using Waterfall model. Waterfall model is very important for software development since the students will be able to understand a sequence of processes in order to accomplish quality information system. Next, the data collection and analysis phase emphasize on requirement elicitation and analysis part. Requirements elicitation or known as requirements gathering is essential in this phase since a set of elements for teamwork assessment needs to be investigated for revealing students' assessment elements. Requirements elicitation process in this study is to ensure that users are satisfied with the product (Rahman and Sahibuddin, 2017).

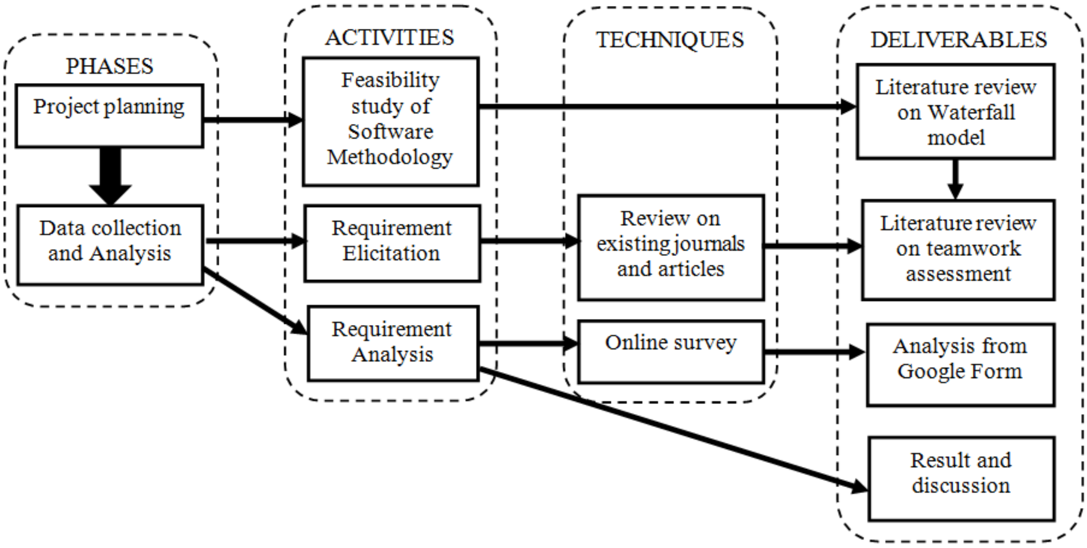
PHASES
Project planning
Data collection and Analysis
ACTIVITIES
Feasibility study of Software Methodology
Requirement Elicitation
Requirement Analysis
TECHNIQUES
Review on existing journals and articles
Online survey
DELIVERABLES
Literature review on Waterfall model
Literature review on teamwork assessment
Analysis from Google Form
Result and discussion

Fig. 3: Research design

4.3. Technique

The technique used for gathering information is reviewing on existing journals and articles regarding teamwork assessment. There are several researches that have been studied related to teamwork criteria which focus on shared leadership, team orientation, redundancy, learning and autonomy. The purpose of collecting these reviews is to measure the impact of student's teamwork assessment.

4.4. Deliverables

Subsequently, online survey has been conducted through Google Form and the analysis could be obtained instantly. The quantitative result is produced from the 53 respondents of Computer Science students to assist in monitoring their project's assessment. The respondents answered questions related with teamwork criteria through online survey. Based on the survey conducted, the results and discussion can be referred in Section 5.

5. Results and discussion

An online survey is conducted to 100 students and 53 students have submitted their responses through the survey questions from (Poženel, 2013). Based on Fig. 4, there is 69.8% of female students and 30.2% of male students who are involved in this survey to measure students' teamwork assessment among CS-related projects. The results report their 53 responses based on shared leadership, team orientation, redundancy, learning, and autonomy.

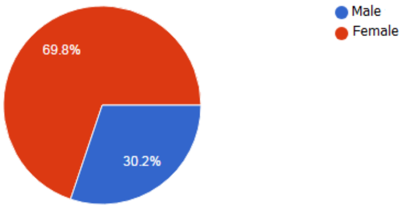

Fig. 4: Demographic Information based on gender

The first criteria are shared leadership which shown in Fig. 5 that highlights the percentage of students' teamwork based on the shared leadership criteria. From the observation, there are 58% of respondents who are agreed and followed by 40% of them who are strongly agreed that everyone in team is involved in the decision-making process. Moreover, 60% of respondents agreed that team members do not make important decisions without consulting other team members. This is supported by 25% of respondents who are strongly agreed with this situation. This shows a clear evidence of shared leadership notion that were embedded within the teamwork since they took other team members' ideas into consideration. Besides, most of 53% of respondents agreed that their team vision is well-defined and presented. Another 43% of respondents strongly agree with this statement. In addition, 66% of respondents agreed that the team is designed (and redesigned) according to its purpose. It is followed by 32% of them strongly agreed with this opinion. Thus, they could achieve a better performance of project assessment as they comprehend the purpose of the project.

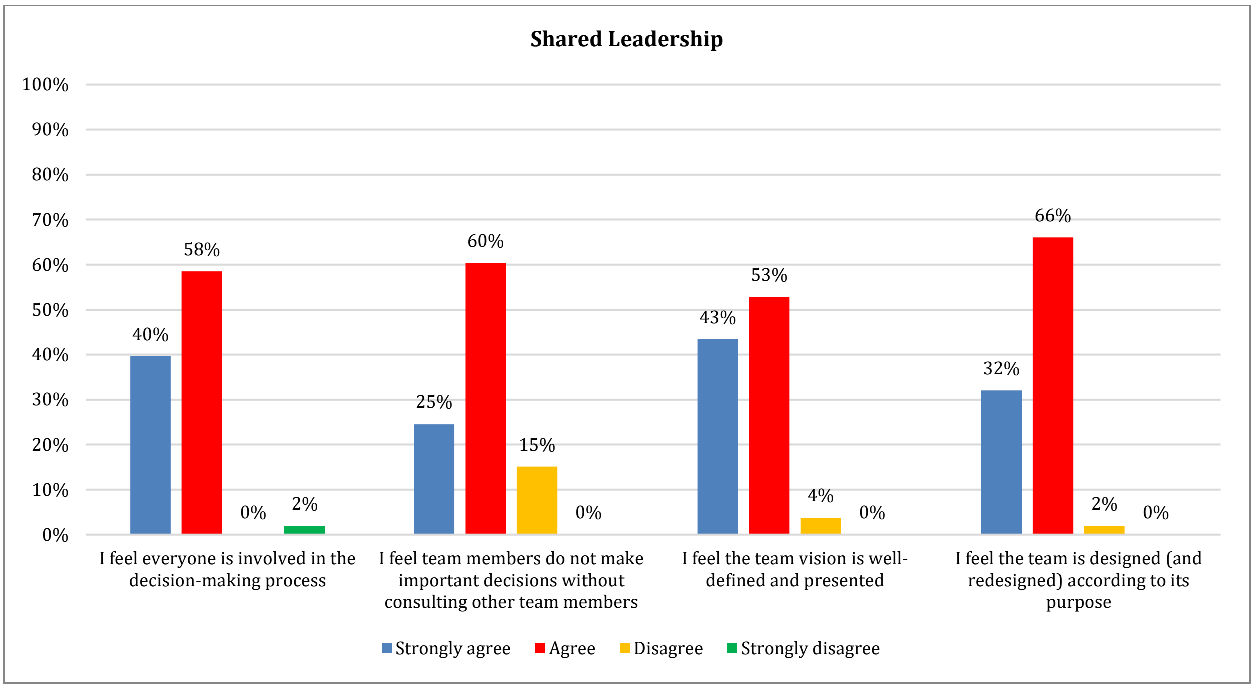

Fig. 5: Percentage of students' teamwork based on shared leadership criteria

The second assessment in teamwork performance is based on the team orientation, which is shown in the below Fig. 6. According to the survey, 68% of respondents agreed that the team takes into account the alternative suggestions in team discussions. It is followed by 28% of them who are strongly agreed with this situation. It verifies the team orientation existence whereby team members are being supportive towards another. Therefore, it seems that teamwork orientation is mostly adapted in project-based development so that the team members can freely exchange the ideas and common understanding of problem. In another perspective, 60% of respondents agreed that they feel the team values alternative suggestions; it is supported by 36% of respondents who are strongly agreed. In this case, the team values describe the team goal setting according to the objectives and mission of the project development. In addition, 72% and 21% of respondents agreed and strongly agreed respectively that the team members relate to the tasks of individual through the regular meetings and discussions. Besides that, 79% of respondents agreed that the team members regularly comment on other member's work for the better improvement. This is supported by another 9% of them who strongly agreed with this practice. From this figure, the team members demonstrate a high degree of cooperation between them. Hence, it can be concluded that a teamwork orientation gives the team members a chance to collaborate and supporting each other's in terms of knowledge, skill and attitude.

The next assessment of students' teamwork is redundancy as depicted in Fig. 7. The both percentages of respondents who are agreed and disagreed that it is easy to complete someone else's task are same which is 43%. Nevertheless, since there are 11% of respondents who are strongly agreed compared to only 2% of respondents strongly disagreed, we can conclude that most of students feel that it is easy to complete someone else's task, with team members' help and guidance. This situation can be truly considered since 66% of respondents agreed that he or she gets help from their team members if a problem occurs. Besides, 30% of respondents also strongly agreed with this statement. In addition, whenever other members have a problem, 79% of respondents agreed that he or she helps the others. This is same as 21% of respondents who are strongly agreed. In other perspective, 43% of respondents disagreed that it is easy to replace a person if he or she leaves the team. Meanwhile, 38% of respondents agreed with this perception. However, since the percentage of strongly agreed respondents are greater than strongly disagreed which is 13% and 6% respectively, we can conclude that more respondents feel it is easy to replace a person's task if he or she leaves the team. This is because each member is able to cooperate with each other by sharing their own abilities, skills and knowledge whenever a problem occurs.

Learning in teamwork is evaluated according to Fig. 8. Based on the survey, 77% of respondents agreed that the team keeps what works well in the development process. This is followed by 19% of respondents who also strongly agreed with this statement. Besides, 68% of respondents agreed that the team improves the development method when software development problems are identified. It is similar to 26% of respondents who are strongly agreed.

From here, it shows that the team members could identify problems occur during the process and discover the solutions together. In addition, the percentage of respondents who are agreed and strongly agreed that the team gives feedback on all aspects of each other's work is 75% and 23%, respectively. Therefore, it is believed that during the teamwork activities, it encourages the team members to interdependently learn something from the collaboration, sharing the information, brainstorming of problem analysis, exploration its solutions, interpersonal communication as well as coordinating of project management.

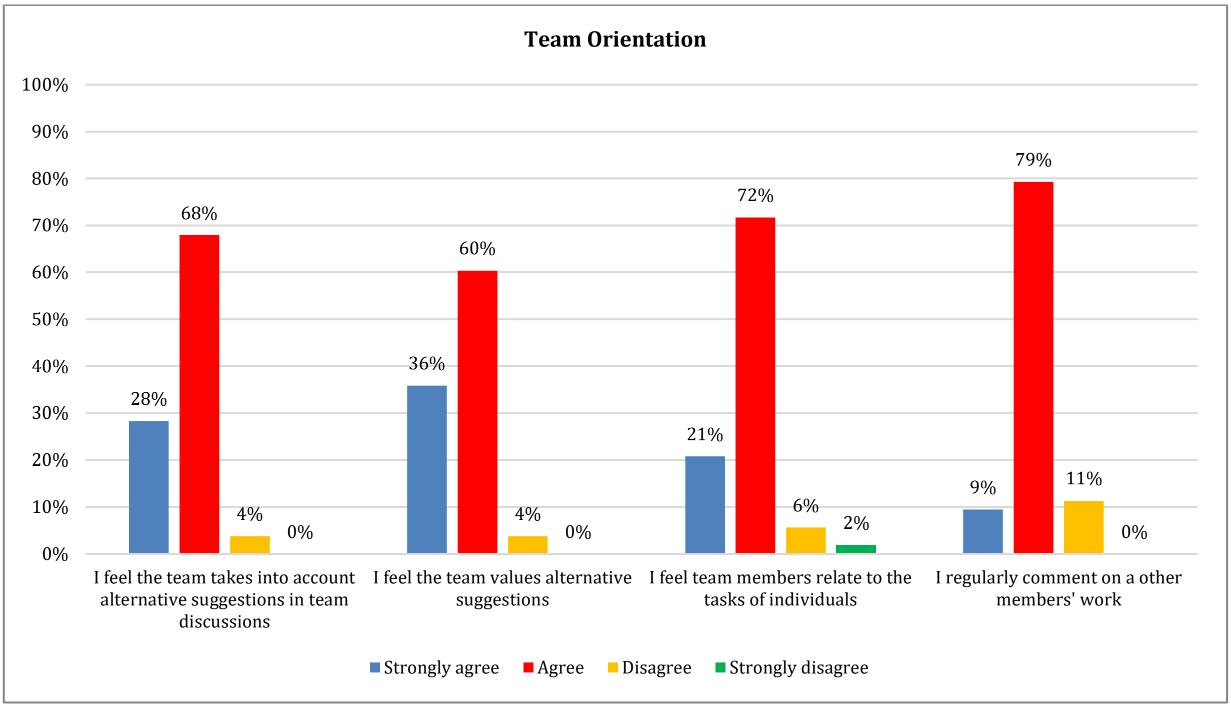

Fig. 6: Percentage of students' teamwork based on team orientation criteria

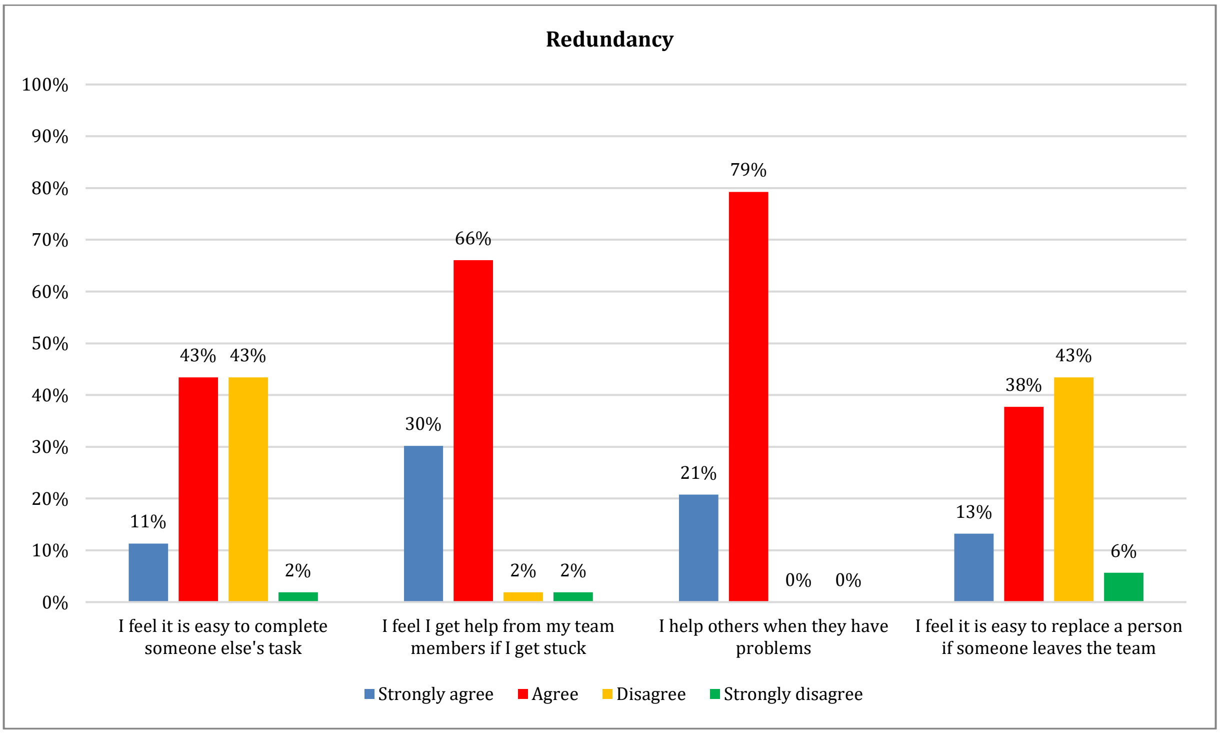

Fig. 7: Percentage of students' teamwork based on redundancy criteria

The last criteria of teamwork assessment are autonomy, which is shown as in Fig. 9. According to the survey analysis, 64% and 11% of respondents agreed and strongly agreed, respectively that the team does not lose too much time and resources to other projects. Meanwhile, another 25% of them disagreed with this statement. Besides, 64% of respondents agreed people and groups outside the team do not have influence over their own team's decisions in the project. It is followed by 21% of respondents who are strongly agreed, and another 15% of them who are disagreed. Lastly, 72% of respondents agreed that the decisions made by the team are respected by people and groups outside the team. This is supported by another 26% of respondents who are also strongly agreed with this perception. Hence, it clearly indicates that most of the teams are autonomous in decision making and distributing the task activities properly, without the interruption from the outsiders.

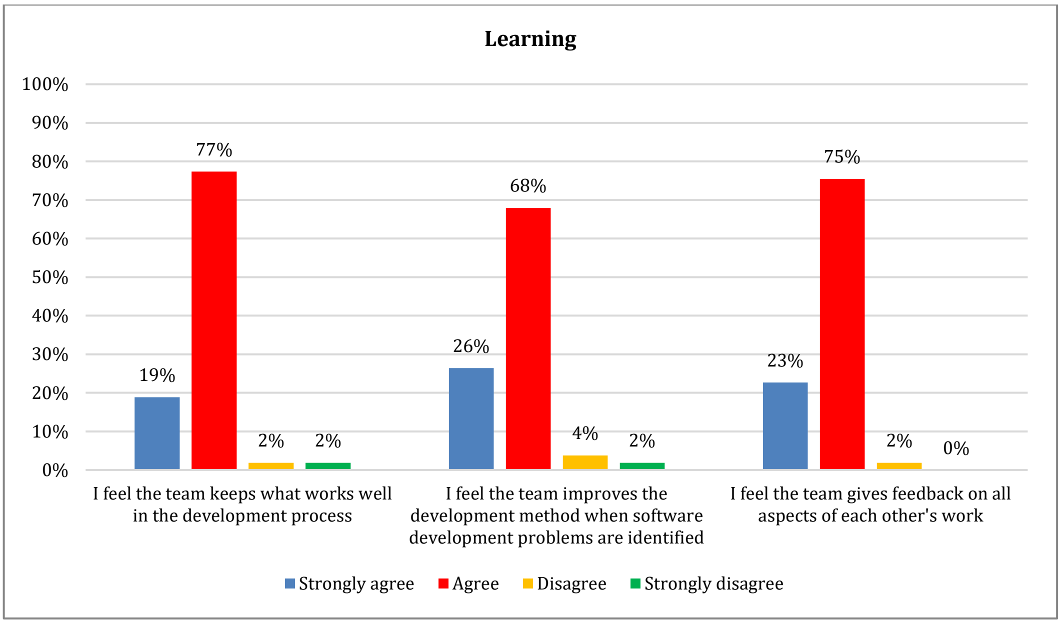

Fig. 8: Percentage of students' teamwork based on learning criteria

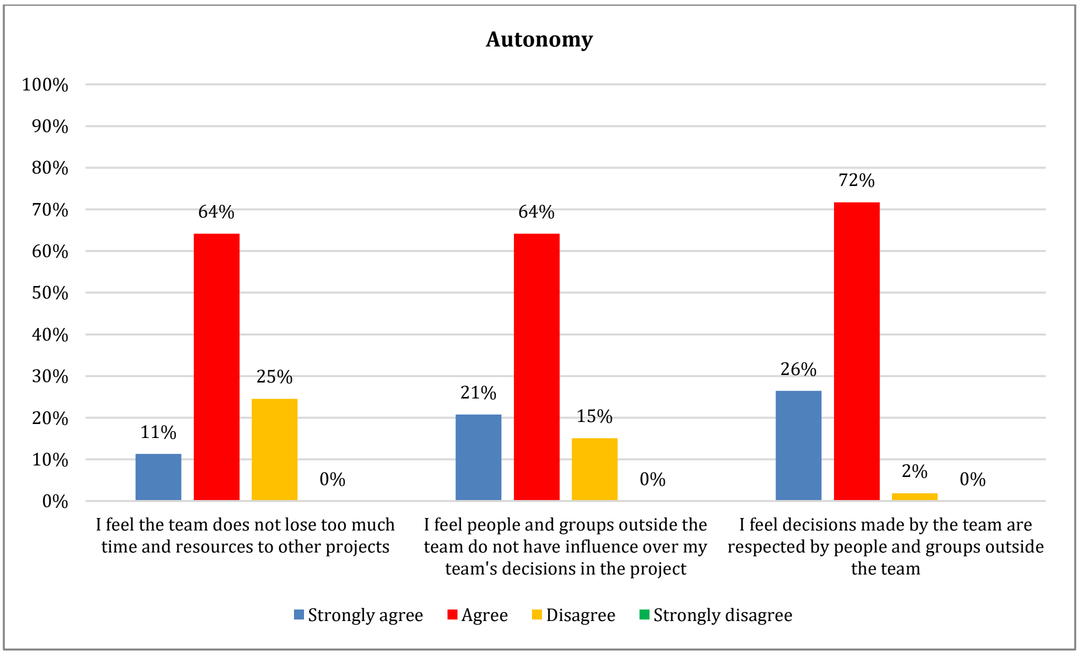

Fig. 9: Percentage of students' teamwork based on autonomy criteria

Based on the evaluation of these five criteria, most of the respondents acknowledge that teamwork skills have significant elements in accomplishing their project development. This study reveals a result in improvement of decision making among students while they were conducting their project. Teamwork skill has helped them to coordinate their task and distribute the task activities properly. The survey also concludes that teamwork skill can promote cooperation and sharing abilities with required knowledge related to software development project.

6. Conclusion

Software project team in education helps the university to shape CS students to understand and play their role as project team members. Besides obtaining result for the subject, the students also polish their skill in teamwork whereby they will learn on helping each other in achieving software goals. Teamwork skill assessment is measure to help lecturers to improve their teaching and learning skill and hence give better impact on software development environment towards the students.

A Waterfall software methodology has been implied in software development class in order to guide student in completing their project. However, the limitation of this study is time consuming issue whereby each phase needs to be taken care of completely in order to proceed to the next phase. In this case, during the testing phase which after development is accomplished, it is time consuming for the students to refine and modify their product development by using Waterfall model. The authors reckon that this study is possible to be expanded by using Agile Software Development Methodology since this methodology promotes agility and flexibility to the requirements changes.

Acknowledgement

The work reported herein was supported by the Universiti Teknologi MARA (UiTM) Kelantan, Malaysia and also the Faculty of Computer and Mathematical Sciences, Malaysia.

References

Baker DP, Salas E, King H, Battles J, and Barach P (2005). The role of teamwork in the professional education of physicians: Current status and assessment recommendations. Joint Commission Journal on Quality and Patient Safety, 31(4): 185-202.

Chen X, Li X, Clark JG, and Dietrich GB (2013). Knowledge sharing in open source software project teams: A transactive memory system perspective. International Journal of Information Management, 33(3): 553-563.

Chiriac EH (2014). Group work as an incentive for learning–students' experiences of group work. Frontiers in Psychology, 5: 558-568.

Day DV, Gronn P, and Salas E (2006). Leadership in team-based organizations: On the threshold of a new era. The Leadership Quarterly, 17(3): 211-216.

Ensley MD, Hmieleski KM, and Pearce CL (2006). The importance of vertical and shared leadership within new venture top management teams: Implications for the performance of startups. The Leadership Quarterly, 17(3): 217-231.

Fidalgo-Blanco Á, Sein-Echaluce ML, García-Peñalvo FJ and Conde MÁ (2015). Using learning analytics to improve teamwork assessment. Computers in Human Behavior, 47: 149-156.

Fuggetta A (2000). Software process: A roadmap. In the Conference on the Future of Software Engineering, ACM, Limerick, Ireland: 25–34.

Ghuman SS (2013). SDLC Models- A Survey. International Journal of Computer Science and Mobile Computing, 2(1): 33–38.

Highsmith J (2013). Adaptive software development: A collaborative approach to managing complex systems. Addison-Wesley, Boston, Massachusetts, USA.

Hogl M and Parboteeah KP (2003). Goal setting and team performance in innovative projects: On the moderating role of teamwork quality. Small Group Research, 34(1): 3-19.

Hogl M and Parboteeah KP (2007). Creativity in innovative projects: How teamwork matters. Journal of Engineering and Technology Management, 24(1-2): 148-166.

Kay J, Maisonneuve N, Yacef K, and Reimann P (2006). The big five and visualisations of team work activity. In the International Conference on Intelligent Tutoring Systems, Springer, Berlin, Heidelberg, Germany: 197-206.

Kumar G and Bhatia PK (2014). Comparative analysis of software engineering models from traditional to modern methodologies. In the 4th International Conference on Advanced Computing and Communication Technologies (ACCT), IEEE, Rohtak, India: 189-196.

LaBeouf JP, Griffith JC, and Roberts DL (2016). Faculty and student issues with group work: What is problematic with college group assignments and why?. Journal of Education and Human Development, 5(1): 13-23.

Leau YB, Loo WK, Tham WY, and Tan SF (2012). Software development life cycle AGILE vs traditional approaches. In the International Conference on Information and Network Technology, IACSIT Press, Singapore: 162-167.

Lindsjørn Y, Sjøberg DI, Dingsøyr T, Bergersen GR, and Dybå T (2016). Teamwork quality and project success in software development: A survey of agile development teams. Journal of Systems and Software, 122: 274-286.

Lingard RW (2010). Teaching and assessing teamwork skills in engineering and computer science. Journal of Systemics, Cybernetics and Informatics, 18(1): 34-37.

Liu S, Hu J, Li Y, Wang Z, and Lin X (2014). Examining the cross-level relationship between shared leadership and learning in teams: Evidence from China. The Leadership Quarterly, 25(2): 282-295.

Nicolaides VC, LaPort KA, Chen TR, Tomassetti AJ, Weis EJ, Zaccaro SJ, and Cortina JM (2014). The shared leadership of teams: A meta-analysis of proximal, distal, and moderating relationships. The Leadership Quarterly, 25(5): 923-942.

Payne BK, Monk-Turner E, Smith D, and Sumter M (2006). Improving group work: Voices of students. Education, 126(3): 441–449.

Poženel M (2013). Assessing teamwork in a software engineering capstone course. World Transactions on Engineering and Technology Education, 11(1): 6–12.

Rahman NA and Sahibuddin S (2017). Improving collaborative activities in e-learning using social presence requirement elicitation process. Pertanika Journal of Social Science and Humanities, 25: 201-210.

Richardson I, Casey V, Mccaffery F, Burton J, and Beecham S (2012). A process framework for global software engineering teams. Information and Software Technology, 54(11): 1175-1191.

Salas E, Sims DE, and Burke CS (2005). Is there a "big five" in teamwork?. Small Group Research, 36(5): 555-599.

Stoica M, Mircea M, and Ghilic-Micu B (2013). Software development: Agile vs. traditional. Informatica Economica, 17(4): 64-76.

Truong VT, Le BN, and Nguyen MD (2015). Assessing the maturity of teamwork capabilities through capstone project. In the 2015 International Conference on Learning and Teaching in Computing and Engineering, IEEE Computer Society, Washington, D.C., USA: 76-78.

Truong VT, Le BN, Nguyen MD, Nguyen TM (2014). Assessing the maturity of teamwork capabilities through cdio projects. In the 10th Annual International CDIO Conference, Universitat Politècnica de Catalunya, Barcelona, Spain.

Weimann P, Pollock M, Scott E, and Brown I (2013). Enhancing team performance through tool use: How critical technology-related issues influence the performance of virtual project teams. IEEE Transactions on Professional Communication, 56(4): 332-353.