Programming Project in an Undergraduate Software Engineering in the New Normal: Challenges and Proposed Solutions

Rex Bringula [a], Sheila Geronimo[b] and Ariel Aviles[c]

[a]University of the East, 2219 CM Recto, Manila, Philippines
[b]University of the East, 2219 CM Recto, Manila, Philippines
[c]University of the East, 2219 CM Recto, Manila, Philippines

Abstract

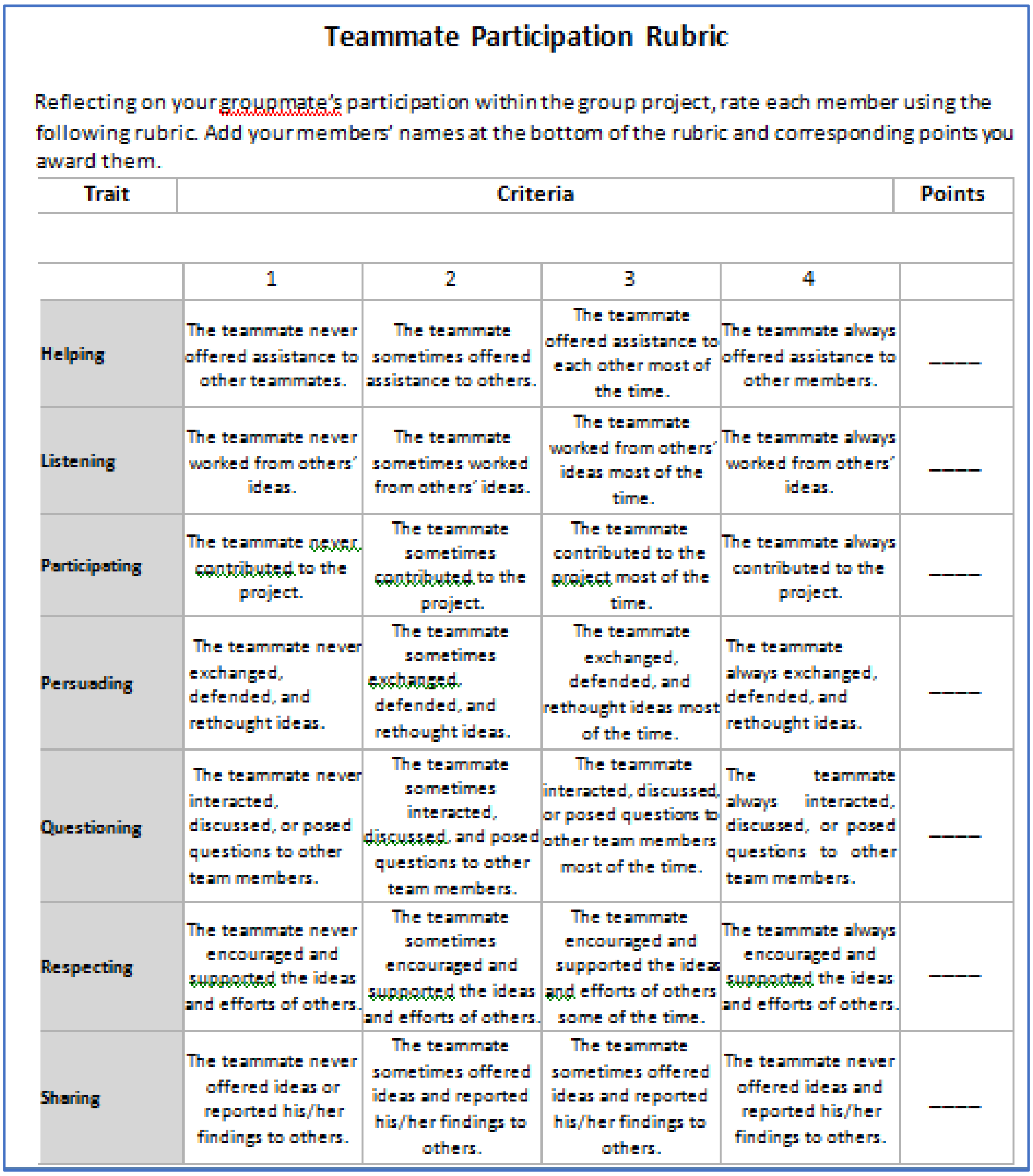
This paper reports the practices of two faculty members of a software engineering course in one university in Manila in terms of programming project requirements during face-to-face sessions (i.e., before the COVID-19 pandemic) and the pandemic. It presents the challenges of implementing this course requirement in the time of the pandemic as well as the proposed solutions to address these issues. It concludes that the activities relating to the software project can still be pursued with modifications and with the teaching styles of the teachers. Future research directions are also offered.

1. The Undergraduate Software Engineering Course before the Pandemic

The undergraduate Software Engineering course (subsequently referred to as SE) is a 3-unit course offered in one university in Manila. This is one of the major courses in Computer Science, Information Technology, and Information Systems degree programs. The course requires 54 contact hours that are split into a 36-hour lecture and 18-hour laboratory sessions. In the lecture sessions, teachers use different teaching-learning strategies (e.g., discussions, group activity, role-playing, etc.) and assessment tasks (e.g., seat works, quizzes, exams, case analysis, group discussions, etc.) [1]. During the laboratory sessions, students develop the software they choose. Both sessions are conducted in a face-to-face setting which requires the presence of the teacher. The course syllabus entails discussions of introduction to SE, process models, Agile software process development, project scheduling, requirements engineering, requirements modeling, design concepts, quality concepts, configuration management, and risk management. The course requires students to analyze, design, develop, and test system models. Students are also expected to develop team collaboration and management skills as well as communication skills. To achieve these goals, the students are required to present their developed systems.

A team of four to five members develops software based on the client's needs (Table 1). The team is composed of a team leader, programmer, user-interface designer, systems tester, and document analyst. The team leader oversees the overall progress of the project and the performance of the team. The programmer does the actual coding while the user-interface designer ensures the ease of using the interfaces of the system and the usability of the system as a whole. The primary function of the systems tester is to examine all possible loopholes of the software and to recommend improvements of the software to the programmer. Finally, the document analyst generates and compiles all records, and develops a systems manual. Throughout these activities, they are required to submit a progress report to the team leader who, in turn, submits it to the teacher.

Joint Proceedings of SEED & NLPaSE co-located with APSEC 2020, 01-Dec, 2020, Singapore
EMAIL: rexbringula@gmail.com (A. 1); ariel.aviles@ue.edu.ph (A. 2); sheila.geronimo@ue.edu.ph (A. 3)



Teammate Participation Rubric

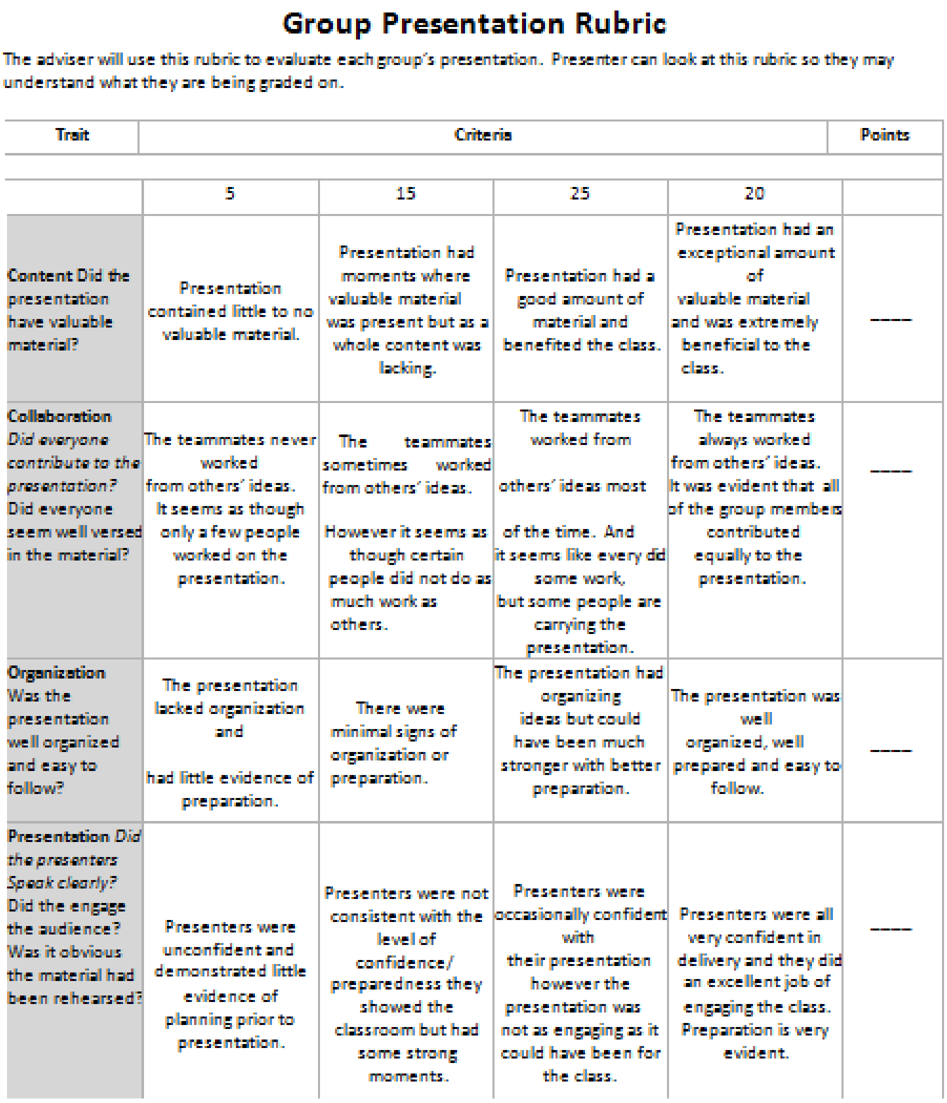
Reflecting on your groupmate's participation within the group project, rate each member using the following rubric. Add your members' names at the bottom of the rubric and corresponding points you award them.

Trait	Criteria				Points
	1	2	3	4	
Helping	The teammate never offered assistance to other teammates.	The teammate sometimes offered assistance to others.	The teammate offered assistance to each other most of the time.	The teammate always offered assistance to other members.	____
Listening	The teammate never worked from others' ideas.	The teammate sometimes worked from others' ideas.	The teammate worked from others' ideas most of the time.	The teammate always worked from others' ideas.	____
Participating	The teammate never contributed to the project.	The teammate sometimes contributed to the project.	The teammate contributed to the project most of the time.	The teammate always contributed to the project.	____
Persuading	The teammate never exchanged, defended, and rethought ideas.	The teammate sometimes exchanged, defended, and rethought ideas.	The teammate exchanged, defended, and rethought ideas most of the time.	The teammate always exchanged, defended, and rethought ideas.	____
Questioning	The teammate never interacted, discussed, or posed questions to other team members.	The teammate sometimes interacted, discussed, and posed questions to other team members.	The teammate interacted, discussed, or posed questions to other team members most of the time.	The teammate always interacted, discussed, or posed questions to other team members.	____
Respecting	The teammate never encouraged and supported the ideas and efforts of others.	The teammate sometimes encouraged and supported the ideas and efforts of others.	The teammate encouraged and supported the ideas and efforts of others some of the time.	The teammate always encouraged and supported the ideas and efforts of others.	____
Sharing	The teammate never offered ideas or reported his/her findings to others.	The teammate sometimes offered ideas and reported his/her findings to others.	The teammate sometimes offered ideas and reported his/her findings to others.	The teammate never offered ideas and reported his/her findings to others.	____

Figure 1: Individual Participation Rubric (Adapted from [5])

The team members are graded individually (Figure 1) as well as a group (Figure 2). For the first rubric (Figure 1), a student may acquire a minimum of 7 points and a maximum score of 28 points. The lowest score that a team can receive is 20 points and the highest possible score is 80 points (Figure 2). The rubric in Figure 3 is used for individual assignments or activities (e.g., investigation of related systems or software).

The current COVID-19 pandemic challenged the current practice of development and presentation of the application of a real-world system (e.g., [15]). Before the pandemic, students were required to develop and implement an application system for a client. This practice is no longer feasible because of the current lockdown. Furthermore, the course will be implemented in a fully online environment. Therefore, the delivery of the content will also be challenging since this is the first time that the course will be implemented online. This paper reports the practices of two male faculty members handling Software Engineering course in one university in Manila before the COVID-19 pandemic. It also presents the challenges in the SE project brought by the pandemic and the proposed solutions in addressing these challenges.

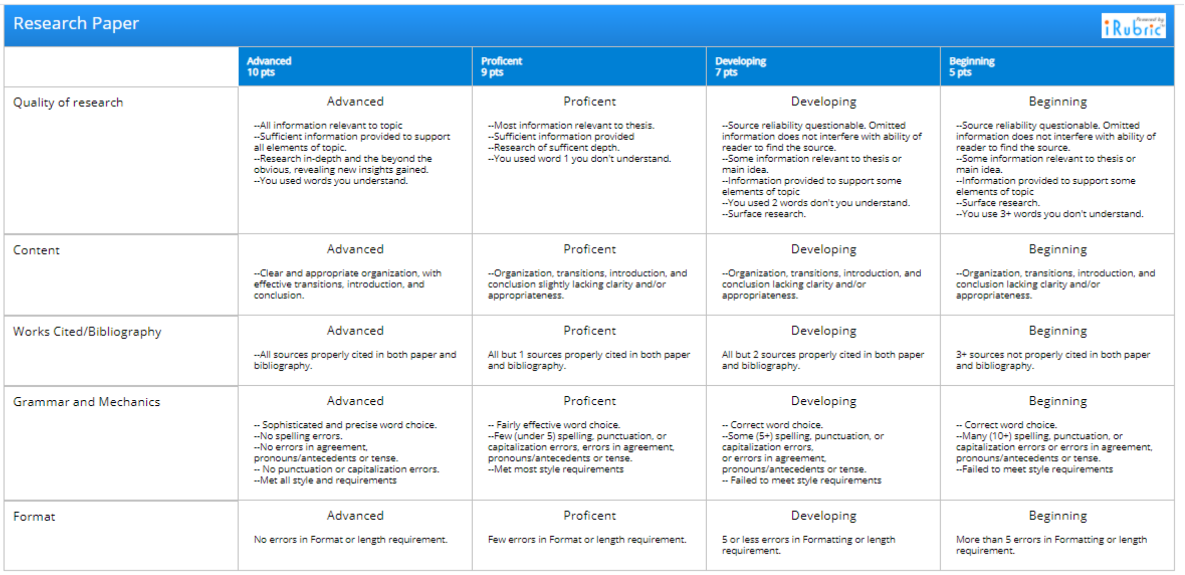


Group Presentation Rubric

The adviser will use this rubric to evaluate each group's presentation. Presenter can look at this rubric so they may understand what they are being graded on.

Trait	Criteria				Points
	5	15	25	20	
Content **Did the presentation have valuable material?**	Presentation contained little to no valuable material.	Presentation had moments where valuable material was present but as a whole content was lacking.	Presentation had a good amount of material and benefited the class.	Presentation had an exceptional amount of valuable material and was extremely beneficial to the class.	____
Collaboration *Did everyone contribute to the presentation?* Did everyone seem well versed in the material?	The teammates never worked from others' ideas. It seems as though only a few people worked on the presentation.	The teammates sometimes worked from others' ideas. However it seems as though certain people did not do as much work as others.	The teammates worked from others' ideas most of the time. And it seems like every did some work, but some people are carrying the presentation.	The teammates always worked from others' ideas. It was evident that all of the group members contributed equally to the presentation.	____
Organization Was the presentation well organized and easy to follow?	The presentation lacked organization and had little evidence of preparation.	There were minimal signs of organization or preparation.	The presentation had organizing ideas but could have been much stronger with better preparation.	The presentation was well organized, well prepared and easy to follow.	____
Presentation *Did the presenters Speak clearly?* Did the engage the audience? Was it obvious the material had been rehearsed?	Presenters were unconfident and demonstrated little evidence of planning prior to presentation.	Presenters were not consistent with the level of confidence/ preparedness they showed the classroom but had some strong moments.	Presenters were occasionally confident with their presentation however the presentation was not as engaging as it could have been for the class.	Presenters were all very confident in delivery and they did an excellent job of engaging the class. Preparation is very evident.	____

Figure 2: Group Presentation Rubric (Adapted from [17])



Research Paper

	Advanced 10 pts	Proficent 9 pts	Developing 7 pts	Beginning 5 pts
Quality of research	Advanced --All information relevant to topic --Sufficient information provided to support all elements of topic. --Research in-depth and the beyond the obvious, revealing new insights gained. --You used words you understand.	Proficent --Most information relevant to thesis. --Sufficient information provided --Research of sufficent depth. --You used word 1 you don't understand.	Developing --Source reliability questionable. Omitted information does not interfere with ability of reader to find the source. --Some information relevant to thesis or main idea. --Information provided to support some elements of topic --You used 2 words don't you understand. --Surface research.	Beginning --Source reliability questionable. Omitted information does not interfere with ability of reader to find the source. --Some information relevant to thesis or main idea. --Information provided to support some elements of topic --Surface research. --You use 3+ words you don't understand.
Content	Advanced --Clear and appropriate organization, with effective transitions, introduction, and conclusion.	Proficent --Organization, transitions, introduction, and conclusion slightly lacking clarity and/or appropriateness.	Developing --Organization, transitions, introduction, and conclusion lacking clarity and/or appropriateness.	Beginning --Organization, transitions, introduction, and conclusion lacking clarity and/or appropriateness.
Works Cited/Bibliography	Advanced --All sources properly cited in both paper and bibliography.	Proficent All but 1 sources properly cited in both paper and bibliography.	Developing All but 2 sources properly cited in both paper and bibliography.	Beginning 3+ sources not properly cited in both paper and bibliography.
Grammar and Mechanics	Advanced -- Sophisticated and precise word choice. --No spelling errors. --No errors in agreement, pronouns/antecedents or tense. -- No punctuation or capitalization errors. --Met all style and requirements	Proficent -- Fairly effective word choice. --Few (under 5) spelling, punctuation, or capitalization errors, errors in agreement, pronouns/antecedents or tense. --Met most style requirements	Developing -- Correct word choice. --Some (5+) spelling, punctuation, or capitalization errors, or errors in agreement, pronouns/antecedents or tense. -- Failed to meet style requirements	Beginning -- Correct word choice. --Many (10+) spelling, punctuation, or capitalization errors or errors in agreement, pronouns/antecedents or tense. --Failed to meet style requirements
Format	Advanced No errors in Format or length requirement.	Proficent Few errors in Format or length requirement.	Developing 5 or less errors in Formatting or length requirement.	Beginning More than 5 errors in Formatting or length requirement.

Figure 3: Rubric for Research Activity (Adapted from [18])

2. Software Engineering Programming Project in an Online Learning Setup

According to McCombs and Vakili [14], e-learning follows psychological principles. One of these principles is setting standards and assessments. This principle aims to assess the learners in their learning progress. Another principle states that learners may experience constraints on learning. Thus, the curriculum must be customized based to allow learners to see the progress they are making in the course. In other words, the curriculum must be flexible and dynamic which is based on the needs and/or constraint considerations.

Edwards [7] shared his teaching strategies in an online graduate Software Engineering program at Virginia Tech. The learning activities include reading assignments, class discussions, homework assignments, group project assignments, and final examination. The first group project assignment entails small group projects where a team of students with 3-4 members collaborates to develop a realistic application. The instructor provides synchronous (e.g., chat sessions) and asynchronous (e.g., email and discussion board) support Students may meet face-to-face or online in performing the assigned tasks. The second group project involved testing the developed system of the other groups. The preliminary evaluation of this course delivery technique showed that the students enjoyed the course and understood the course materials.

The study of Jacobsen et al. [9] reported their experience with two distance graduate courses in software engineering. They utilized WebCT, electronic mail, a list server, and NetMeeting in the implementation of the course. They recommended that there should be mechanisms and support systems that will allow iterative development and assessment of distance learning. They also reported that time lags, unexpected barriers, and longer course development periods in teaching the course online. Some features of WebCT are not used because they are not applicable in graduate distance learning. It was shown that distance learning requires well-structured and well-indexed web materials and resources to be established. Recently, mobile applications are gaining popularity as a resource for software engineering materials [13, 20].

Edwards and Sridhar [6] investigated the effectiveness of global virtual teams in software engineering projects. Specifically, the study determined the relationship between the independent variables (e.g., ease of use, the structure of project tasks, the effect of the time difference, trust between teams, academic orientation, cultural orientation, and size of the team) and the dependent variables (e.g., learning effectiveness, quality of projects, team project experience, and software engineering process). The participants of the study are master's degree students in India and Canada with 3-6 members in a team. It was revealed that ease of use of technology, trust between the teams, and well-defined tasks structure were related to learning effectiveness, project experience, and software engineering process. The study concluded that ease of use of technology, trust between the team members, and well-defined task structure were the factors that have a positive relationship with the efficiency, effectiveness, and satisfaction level of global virtual teams.

Leung [11] compared the effectiveness of e-learning and traditional courses delivery of a graduate course in software engineering. The e-learning course is supported by online teaching materials, assessment tools, online tutorials via audio conferencing, and bulletin board discussions. The class formed a team consisting of 3-4 members for a group project. The project involved a survey of a particular testing method or a detailed proposal to implement a certain testing method. The traditional model involved 13 weeks of face-to-face lectures and tutorials with the same course requirements. It was found that the e-learning mode is as effective as the traditional mode of teaching.

Bouillon et al. [2] that communication, coordination, and collaboration were the challenges in implementing software engineering projects. A software engineering project is a major requirement of the course. Students should develop a project of realistic scope using realistic software development tools. To achieve this goal, the researchers used software named ECLIPSE which allowed distance learning students to collaborate in a team to produce a large software system. The study found that the system facilitated communication among the team members. However, it failed to define clearly the tasks and responsibilities of group members.

Demir [4] investigated the difficulties faced by 78 software practitioners in managing a software project. The author found that the top 10 challenges software engineers face are in the areas of scope management, requirements management, project planning/estimation, communication, staff/hiring, project monitoring/control, risk control, technical complexity, stakeholder involvement, and leadership. The author also found that the project is more difficult to manage when the team is relatively large (with 11-100 members).

3. Methodology

This qualitative study interviewed two faculty members in one university in Manila. Both faculty members have at least 20 years of teaching experience and have international certification in Scrum Methodology. Telephone interview sessions with the two were conducted to gather data. They were asked about the nature of the SE programming project, their teaching practices before the pandemic, the challenges brought by the pandemic in terms of executing the software project development, and the proposed solutions to these challenges.

Responses (subsequently referred to as text) were analyzed through the grounded theory approach to determine the emerging themes. The themes were categorized based on the activities in the development of the project (Table 1). The activities are based on the practice of the informants. The approach includes open, axial, and selective coding, and constant comparative methods [1]. Using a word processor, the first researcher coded the texts independently. The codes were guided by the questions "*What keyword/s, phrase, or sentence that could capture the idea of the informants' response with the question?*" and "*How can these codes be valuable in explaining their response*?". Open codes were identified through the keywords, phrases, or sentences that could represent the responses of the informants were identified in the text.

The open codes were verified through consultation with the second researcher. The first researcher presented the codes to the second researcher. The purpose of the presentation is to determine whether the second researcher would agree. In case of disagreements, the researchers deliberated until a consensus was achieved (i.e., at least 2 researchers have the same views on the issue being resolve). These steps were repeated until all open codes were analyzed. The texts are then combined to determine if the texts are relevant to the themes.

To maintain the accuracy of this report, these writers sent the paper to the informants (i.e., respondents of the interviews) and sought their comments and suggestions. The paper was revised until the authors and the informants reached a consensus.

4. Programming Project during the COVID-19 Pandemic, Key Challenges, and Proposed Solutions

Table 1 shows the summary of activities for the programming project requirements before the pandemic and the key challenges during the pandemic. The grounded theory approached identified 13 key challenges.

(Key Challenge 1) *Attainment of Course Outcome,* (2) *Determination of Scope of Work,* (5) *Digital Collection of Software Projects,* (6) *Topics Duplication,* (7) *Beneficiaries of the System,* (8) *Project Scope Definition, and* (12) *Systems Implementation*

All activities leading to the completion of the project resembled a virtual team since all members who were distributed geographically, collaborated online [12,16]. The primary challenge brought by the pandemic was to relax the course requirements without sacrificing the need to meet the course outcomes. The previous practice of selecting a topic and defining the scope of the project involved a frequent visit to the library and a series of personal interviews with the stakeholders, clients, or customers. These prior activities are no longer feasible because of the lockdown. The key issue involving topic selection and scope definition is that the library's digital collection of previous projects is not available online. Students' prior works are submitted as hard copies and are not digitized. Therefore, there is a need to limit the scope of the programming project, at least for SE, since students are not able to physically visit the library and a company, to interview key personnel, and to gather insights from a company that could serve as their client. Even if the lockdown is lifted, the threat of being infected by the virus is still high because the number of infections is still rising.

Table 1. Summary of Activities for Programming Project before COVID-19 Pandemic, Key Challenges, and Proposed Solution

Activities	Before the Pandemic	Key Challenges	Proposed Solution	Basis
Description of the project requirement	• Students are required to submit a fully functional system	1. Attainment of the course outcome	• Submission of a prototype system	• McCombs and Vakili [14]
Determination of Team size and composition	• 4 to 5 students per group	2. Determination of the scope of work of each team member	• Retention of the team size and composition • Re-definition of tasks	• Edwards [7], Edwards and Sridhar [6]; Bouillon et al. [2]
Member inclusion	• After a few face-to-face meetings, students select their team members during class sessions.	3. Students, not knowing who is officially enrolled in the course 4. Difficulty in determining if the team reached the maximum number of members	• Distribution of the class list thru the Learning Management System (LMS) and selection of members by the students • Use of the LMS Grouping function	• Jacobsen et al. [9]
Topic selection	• Students may visit the library for hard copies of existing systems • Stakeholders/clients/customers can be consulted	5. Non-digitized copies of software projects 6. Duplication of topics **7.** The difficulty of finding possible stakeholders/clients/customers (i.e., beneficiaries of the system)	• Proposal of a "generic" prototype system instead of a fully functional software	• Bouillon et al. [2] • Mahatanankoon, et al. [13], Vollmer [20]
Scope definition	• Students may visit the library for hard copies of existing systems • Stakeholders/clients/customers can be consulted	8. Not well-defined project scope	• Use of the library digital collection and Google Scholar for review of related systems	• Jacob et al. [2000], Demir [4]
Actual coding	• Independent and group work	9. Systems integration since they are working remotely/ independently	• Proposal of a "generic" prototype system will be proposed instead of a fully functional software	• Bouillon et al. [2], Demir [4]
Monitoring	• Progress is reported during class sessions (i.e., group reporting)	10. The difficulty of taking turns through real-time meeting applications 11. Intermittent Internet connection	• Progress report during synchronous or asynchronous sessions	• Jacob et al. [2000],
Implementation	• Students implement the software to the company	12. Implementation not feasible because of lockdown	• Proposals of web-based or mobile-based systems. • Alternative redefinition of the course requirement.	• Jacob et al. [2000]
Presentation	• A working system together with its documentation is presented during class sessions	13. Intermittent power supply to some areas Intermittent Internet connection	• Presentation and documentation during synchronous sessions • Submission ahead of the presentation • Provision of ample time for the presentation • Pre-recording of the presentation	• Jacob et al. [2000]

(3) *Member Inclusion and* (4) *Team Size*

Another activity is the determination of member inclusion. Because students work as a group, the inclusion of a member with a tendency not to perform well has an impact on the overall project and the team [19]. In a classroom setup, a teacher may require a series of group work and individual activities so that students can easily identify who among their classmates they are comfortable to work with. This group dynamics may not be observed during online sessions. Furthermore, it is easier to determine whether a team reached the required number of members by simply asking the team one by one. This has become a challenge in an online setup since students do not know who is officially enrolled in the class and how to initiate forming a team. The proposed solution for the issue is to send the official class list to the students through the Learning Management System. Afterward, teachers may use the Group Discussion function of the LMS. This LMS functionality allows teachers to set a virtual space for the teams. The virtual space is partitioned into several groups and each group has a specified number of members (e.g., 4-5 members per group). Students of the same team will occupy one virtual space. Students will not be able to occupy multiple virtual spaces. Team composition is retained since the team size in this study is consistent with prior studies [2, 6, 7].

(9) *Systems Integration,* (10) *Monitoring Teams' Progress*

While a particular student may be designated as a programmer, all members of the team may play this role. Students collaborate in developing the software; sometimes, they schedule sleepover sessions to finish the project. In other words, before the pandemic, the software is built in a cooperative environment. In this period of a public health crisis, students are compelled to write the program all by themselves because assistance and feedback from their teammates may not be easily available. The team may work independently on different modules and later integrate them to build the software. However, this poses a challenge to the team since each of them may have different programming level skills, styles [3], and personalities [10].

The progress of a project is tracked through the Gantt chart and it is presented during classroom sessions in a periodic manner. Each member of the team presents their accomplishments on their tasks. They are rated accordingly using the rubrics shown in Figures 1 to 3. Progress report activity and the use of the same rubrics are practiced in an online class since this is an integral of the syllabus. The key challenge in this activity is the intermittent Internet connection which, in turn, leads to the difficulty of taking turns in the presentation. To address this, a flexible form of presentation can be done either through synchronous or asynchronous sessions. During the asynchronous sessions, students may record their presentations and upload them to the LMS. This practice is similar to that of Edwards [7].

It was found that both informants agreed that the monitoring of progress is favorable to the teachers and students. Unlike in a classroom setup which is very rigid in terms of class attendance, asynchronous learning sessions allow students to work without pressure. Two-thirds (or 36 hours) of the total class hours allotted to asynchronous sessions are sufficient to develop the prototype system. From the teachers' perspective, these asynchronous sessions can be used to check the progress of the students and check other course requirements. The interview also revealed that the implementation of the completed software may not be viable during this pandemic. One of the proposed solutions is to shift all proposals to web- or mobile-based systems. However, this may not be feasible since students may not have taken yet the skills in the development of these kinds of systems. An alternative solution is to re-define the course requirement, i.e., a prototype of these systems may be presented instead of a fully functional system.

(11) *Intermittent Internet Connection* and (13) *Intermittent Power Supply*

Finally, the primary concern of teachers is the intermittent Internet connection during project presentation. This finding is consistent with the findings of Fabito and his colleagues [8]. The informants agreed that this can be resolved through early submissions of the documentation and the pre-recording of the presentation. Teachers may read the documentation and list all the questions ahead of the presentation. A 15-minute video presentation will be presented through the LMS conference system or any online meeting application. Teachers may then ask questions after the presentation. One of the informants said that some students also experienced intermittent power supply. Therefore, teachers must account for these situations in the timetable of the project presentation.

The two informants further pointed out that in an asynchronous setup, students are forced to practice self-regulated learning strategies. Here, students heavily rely on the lectures, videos, and presentations of the faculty. Students are reinforced to plan for their learning assignments, monitor their performance, and evaluate the outcomes. The asynchronous learning sessions also pushed them to create their learning environment in a convenient time and space. Overall, the study found 13 key challenges in teaching SE in this time of pandemic (Table 1).

The course requirement is changed in terms of the nature of the system. This is consistent with McCombs and Vakili [14] considering the constraints in the implementation of the prior requirement. Nonetheless, even though the nature of the project is changed, students are still required to develop a generic prototype of the system and undergo the process of software development. Changing the nature of the system did not entirely change the process of software development. To retain the learning outcomes, all projects must be a prototype of web-based systems (e.g., e-commerce websites, online billing systems, hotel reservation systems, etc.) or mobile applications. As mentioned in the literature review section, there is a growing body of research on mobile and web-based application systems. Thus, students will find it easier to analyze similar systems and formulate their software projects. With these types of systems, students can still simulate the analysis, design, development, and testing of system models. Team collaboration, management skills, and communication skills can still be achieved since the current LMS, instant messaging tools, and online meeting tools (e.g., Zoom, Google Meet, etc.) can facilitate collaboration, communication, and coordination. The requirement of the course in terms of presentation is achievable through the proposed solutions. With the use of online meeting applications, students can still present their projects and can be graded accordingly using the rubrics. Individual performance tasks are monitored and assessed since the work definition of each student is clearly defined. Therefore, the proposed solutions can still achieve the desired learning outcomes of the course. The above recommendations are in agreement with the studies of Bouillon et al. [2], Edwards [7], Edwards, and Sridhar [6], Jacobsen et al. [9], and Leung [11].

14.Conclusion and Future Research

This workshop paper reports the practices of two SE faculty members in one university in Manila in terms of managing the final requirement of the course (i.e., software project). While it is obvious that the COVID-19 pandemic disrupted the classroom practices of software programming, it found that the teachers think of ways to address the key issues and willingly share their proposed solutions to address the identified 13 key challenges in teaching a software project. Of these challenges, the faculty members could not directly address the issue of Internet connection and power supply. It is worth noting that all activities and the rubrics are retained because of the creativity and flexibility of the faculty. Thus, it can be concluded that the software engineering programming project can be pursued during the pandemic.

At the end of the semester, the two informants plan to evaluate the effectiveness of the proposed solutions in terms of appreciation of course content. Feedback may be elicited from the students to identify improvements in the proposed solutions.

15.Acknowledgement

This paper is partly funded by the University of the East. The authors are indebted to the informants of the study and Dr. Mila Arias.

16.References

[1] R. Bringula, R Elon, L. Melosantos, J. R. Tarrosa, Teaching Agile Methodology through Role-Playing: What to Expect and What to Watch Out, Proceedings of the 2019 3rd International Conference on Education and Multimedia Technology, 2019, pp. 355-359.

[2] P. Bouillon, J. Krinke, S. Lukosch, Software engineering projects in distant teaching, in: Proceedings of the 18th Conference on Software Engineering Education & Training (CSEET'05), IEEE, 2005, pp. 147-154.

[3] F. Q. Da Silva, C. Costa, A. C. C. Franca, R. Prikladinicki, Challenges and solutions in distributed software development project management: A systematic literature review, 5th IEEE International Conference on Global Software Engineering, 2010, pp. 87-96.

[4] K. A. Demir, A Survey on Challenges of Software Project Management. Software engineering research and practice, 2009:579-585.

[5] District 186 Springfield Public Schools, Teammate participation rubric, available at https://www.sps186.org/downloads/basic/285671/Teammate?fbclid=IwAR3QAhWc_7Vt-i0C25Ty2i8fEirj_YrEjIL71VMvpfKuAWHzbReLxAPWBfg

[6] H. K. Edwards, V. Sridhar, Analysis of the effectiveness of global virtual teams in software engineering projects, in: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, IEEE, 2003, pp. 9-17.

[7] S. Edwards, Can quality graduate software engineering courses really be delivered asynchronously on-line?, in: Proceedings of the 22nd international conference on Software engineering, ACM Press, New York, NY, 2000, pp. 676-679.

[8] B. S. Fabito, A. O. Trillanes, J. R. Sarmiento, Barriers and challenges of computing students in an online learning environment: Insights from one private university in the Philippines, International Journal of Computing Sciences Research, 5(2021): 451-458. https://dx.doi.org/10.25147/ijcsr.2017.001.1.51

[9] M. Jacobsen, R. Kremer, M. L. Shaw, Experiments with distance learning in software engineering graduate courses. ACM SIGCSE Bulletin, 32.2(2000): 56-59.

[10] J. S. Karn, S. Syed-Abdullah, A. J. Cowling, M. Holcombe, A study into the effects of personality type and methodology on cohesion in software engineering teams, Behaviour & Information Technology 26(2007): 99-111.

[11] H. K. Leung, Evaluating the effectiveness of e-learning, Computer Science Education, 13.2(2003): 123-136.

[12] S. C. Lilian, Virtual teams: Opportunities and challenges for e-leaders, Procedia-Social and Behavioral Sciences 110 (2014):1251-1261.

[13] P. Mahatanankoon, H. J. Wen, B. Lim, Consumer-based m-commerce: exploring consumer perception of mobile applications. Computer standards & interfaces, 27(2005): 347-357.

[14] B. L. McCombs, D. Vakili, A learner-centered framework for e-learning. Teachers college record, 107(2005): 1582-1600.

[15] G. C. Ng, A Study of an Agile methodology with scrum approach to the Filipino company-sponsored I.T. capstone program. International Journal of Computing Sciences Research, 2.2(2018): 68-88. doi:10.25147/ijcsr.2017.001.1.25

[16] G. Paré, L. Dubé, L. Virtual teams: An exploratory study of key challenges and strategies, in: ICIS 1999 Proceedings, 2019, p. 50. Available at http://aisel.aisnet.org/icis1999/50

[17] PBL 101, Workshop Resources, available at https://pbl101.weebly.com/uploads/3/1/3/1/31318861/group_presentation_rubric.pdf

[18]RCampus, iRubric: ENG/LA 7 Research Activity rubric, available at https://www.rcampus.com/rubricshowc.cfm?code=RX69A79&sp=yes&

[19] R. N. C. Recario, M. B. B. de Robles, K. E. P. Bautista, J. P. Pabico, Social Loafing Among Members of Undergraduate Software Engineering Groups: Persistence of Perception Seven Years After, arXiv preprint arXiv:1509.05338, 2015.

[20] T. Vollmer, There’s an App for That. Libraries and mobile technology: An introduction to public policy considerations. USA: ALA, Office for Information Technology Policy, 2010.