A teaching case: self- and peer-assessment system for group-project

Qingxiong Ma
University of Central Missouri

ABSTRACT

This instructional case is designed for software engineering students to help them understand system design process. It can be used in courses such as systems analysis & design, system development, and database design. It provides a scenario for developing self- and peer-assessment systems for group-project. Since the case is derived from real life in academic environment in which students are familiar with the settings and roles each actor plays in the project, by incorporating this case in the class, students will have a better and more accurate understanding of the requirements. Leveraging group-project-based learning techniques, students will create and work with projects which challenge them to design, implement, and demonstrate a system solution for a business or organization. To develop students' team skills and make them work effectively in a team is also listed as one of student outcomes by ABET Computing Accreditation Commission. In addition to the detailed case description, three alternatives of case application are discussed too. The resultant system from the case may be used to facilitate the evaluation/assessment process in the student's group projects and enhance goal of teaching team skills and competences. It can also be used to help instructors in group-project grading. This case had been used by the author in software engineering and database classes.

Keywords: peer-evaluation, self-assessment, group project, instructional case, software engineering, systems analysis & design

INTRODUCTION

Leveraging group-project-based learning techniques, students will create and work with projects which challenge them to design, implement, and demonstrate a system solution for a business or organization. To develop students' teamwork skills and make them work effectively in a team is not only integrated as part of their educational development by many engineering and information technology/systems programs, it is also listed as one of student outcomes by the Accreditation Board for Engineering and Technology (ABET) [1]. Thus, it is easy to understand that the faculty in computer technology programs frequently expects students to do some team projects in their classes [7]. One of the effective approaches to enhancing such teaching goal is using guided self- and peer-assessment [5]. However, assessing or evaluating the performance of each student objectively, timely, and conveniently has never been an easy work in practice.

This instructional case is designed for software engineering students to help them understand system design process. It can be adopted in courses such as systems analysis & design, system development, and database design. It provides a scenario for designing an online system for developing self- and peer-assessment for team-project. The resultant system from the case may be used to facilitate the evaluation/assessment process in the student's team projects and enhance goal of teaching team skills and competences. It can also be used to help instructors in team-project grading. It will allow assessing the team member performance multiple times in a semester along the project progress. Thus, students can get feedback immediately from their peers and improve their performance. It reduces manual effort of the instructors and also reduces human errors by automating the process using the web-based system. Since this case is derived from real life in academic environment in which students are familiar with the settings and roles each actor plays in the project, by using this case in the class students will have a better and more accurate understanding of the requirements.

The software development life cycle is typically divided up into stages going from abstract descriptions of the problem to designs then to code and testing and finally to deployment. In the development process, an output artifact of one phase serves as input of next phase. However, this output artifact does not need to be completely developed. Analysis and design may occur in parallel, and in practice, the results of one activity can feed the other in a short feedback cycle through an iterative process. Therefore, the case presented in this paper can be used as an on-going project or a project at each phase in software life cycle. This case had been used by the author in software engineering class and database class.

CASE SYNOPSIS

To teach effective-team skills and competences, the CIS faculty at a local university always requires students to do some projects in classes. These projects are identified from local community and developed by students in an iterative and incremental process. It is believed that multiple assessments is preferable because student can learn from the feedback from the previous stage in the team project. At present, the instrument used for this assessment is paper-based (see attachment) and the rubrics was developed based on previous research and department faculty brain-storming. However, some challenges and difficulties are encountered when using this instrument in practice [4, 14]. First, it is the anonymous issue. Some students can peek or see the evaluation results of other members. Second, it is tedious and time-consuming for faculty to do analysis. Generally, there are 5-6 teams in each class and each team is composed of 3-4 voluntary students. The volume of evaluation data set is tremendous and the calculation for analysis is complicated. Third, with paper-based instrument and manual processing it is difficult for students to get an instant feedback to improve their performance. Dr. Larry Henson, a software engineering professor, decides to ask his students to design an online system to overcome these difficulties.

THE CASE

Students in the CIS department at a local university are required to do some team projects in their classes because their instructors believe the skills and competence to work as an effective team are important to assure a successful IT project. Group projects help IT students apply system design knowledge, solve real world business problems, and serve back to the community. They also enhance learning effective team skills and improve students' business communication skills.

Each group project will have specific and detailed requirements in different courses. For instance, students in database class are required to gather system requirements by interviewing business users, analyzing business documents, identifying potential entities including their attributes, identifier, the relationships among entities, and then creating the entity-relationship (E-R) Diagram. In addition, they should validate and revise the diagram based on design principles and guidelines such as normalization. In the end, they should implement the database system and write SQL statements to support each function in the system requirements.

Nevertheless, how to develop students' team skills and make them work together effectively so that they can have a successful project is always a challenge topic. In order to ensure an effective and functional team, each team has 3-4 members and a team leader is elected. The team makes a project plan based on the project activities. The team communication format is also determined. During the semester, students are asked to do multiple self- and peer-assessment using the instrument in Appendix I. The rubric form for the assessment (Appendix II) were developed based on previous studies [2, 3, 6, 8, 15] and department faculty brainstorming.

Due to the current paper-based instrument and manual process of administering the assessment, collecting the forms, data re-entry, calculation, and analysis, it is tedious and time-consuming for faculty members to implement the assessment, and it is difficulty for students to get an instant feedback from previous stage to improve their team performance. To address this issue, Dr. Larry Henson, a software engineering professor, decides to ask his students to design a web system that can collect student evaluation, do analysis, and generate reports.

To have a better understanding of the functions and requirements of this assessment system, the faculty in the department had a meeting. The summary of the meeting minutes is presented below.

1. The system should allow Faculty assign students to different project teams. The same student in different class may participate in different project teams.
2. The system should allow users to login with different roles such as faculty, students, and administrator.
3. The system should allow a Faculty manages project teams for each class the faculty is teaching.
4. The system may also be adopted by faculty in other departments as well as CIS faculty because some program core courses are offered by other departments. It is

good to specify a system administrator. The administrator can manage faculty, classes, students, and rubric information.

5. Students can do both self- and team-based peer assessments based on the specified rubrics.
6. Students are allowed to view the assessment done by the team members so that they get instant feedback to improve their performance.
7. The assessment should include both close-ended and open-ended questions. For instance, students are allowed to make comments upon assessing other team members
8. The system should automatically generate different reports such as the average at the level of individual, team, and class.
9. Students are also allowed to make comments upon assessments were done by other members.
10. All students' comments should be reviewed and approved before exposed to other members.

In addition to the meeting minutes, Dr. Larry encourages his students to interview other CIS faculty and students who had team-projects experience for more detailed expectation and requirements.

CASE APPLICATIONS AND LEARNING OBJECTIVES

This case is suitable for both undergraduate and graduate courses such as software engineering, systems analysis & design, system development, and database design. See below the three application alternatives and specific learning objectives for the case.

Application One: System Requirements in Software Engineering

Requirements engineering is the first step in the software design process. According to Rosenberg [11], three types of requirements are identified: functional requirements, domain modeling, and behavioral requirements. Functional requirements define what the system should be capable of doing; domain modeling makes sure understanding the problem scope in unambiguous terms; behavioral requirements define how the user and the system will interact (i.e., write the first-draft use cases). It is recommended that starting identify all the use cases with a GUI prototype when exploring the requirements. From this requirements analysis, use cases can be identified, a domain model is produced and some prototype GUIs are created.

Teaching objectives:

1. Learn the guidelines of writing functional requirements and apply these skills to a specific business case.
2. Learn how to describe the system usage via scenarios and identify use cases based on functional requirements, domain objects, and GUI prototype.
3. Learn UML modeling language conventions and create Use Case Diagram and Class Diagram.

Proposed assignments:

1. Ask students to define the scope of the project and identify functional requirements.
2. Require students to identify domain classes and create a domain model.

3. Develop preliminary screen shots or mock-up for each user interface
4. Ask students to define the way that the user and the system will interact by using use cases and GUI prototypes.

Application Two: Robustness analysis and diagram

Among many methods and techniques of software development, Agile and UML modeling have been heavily researched and documented in literature. Different from other endeavors, ICONIX Process claims to be a minimalist (core subset of UML), streamlined approach that focuses on that area that lies in between use cases and code. ICONIX process leverages the benefits of robustness analysis [11, 12]. In software design literature, robustness analsysis and diagram have been used in research papers with topics such as "agile project modeling", Model-View-Controller architecture, and reverse engineer of legacy systems. Examples of commercial UML modeling tools supporting robustness diagrams are Visual Paradigm for UML [16], Enterprise Architect [13] and MagicDraw UML [9].

The outcome of robustness analysis is a robustness diagram, which models the behavior in a use case using objects. Robustness diagrams are not a formal part of the UML, but similar to UML collaboration diagrams with far less constructs and syntax rules. A robustness diagram can be readily evolved into more detailed UML design artifacts such as collaboration, activity or sequence diagrams. Therefore, robustness analysis and diagrams can be considered as a valuable tool to bridge the gap between the analysis and design phases [11].

Robustness analysis diagrams are very helpful to organize objects and discover missing objects. Objects are represented with the three icon types. The first one represents an interface class - it interfaces with an actor. The second one represents an entity class. Entity classes keep track of information - so these are things like databases. The third icon represents control classes. Control classes manage processes; they usually perform actions and they don't usually keep track of information - they usually turn into methods or functions.

Teaching objectives:

1. Decompose narrative use case flow into smaller and manageable "steps".
2. Use objects from the domain model and link them together to simulate the steps.
3. Present information main flow in a diagram.
4. Discover new objects (boundary, control, entity) and add them in the domain model. Complete new information in use case description that might be missing.
5. Add alternative flows in use case descriptions and highlight them in the robustness diagram to distinct from the nominal flow.

Proposed assignments:

1. Create a domain model for each use case based on use cased description
2. Develop robustness diagram for each revised use case flow.
3. Add new objects and revise the use case description if necessary.
4. Revise and validate robustness diagram to assure it is complete, appropriate, and fault-free.

Application Three: System Data Modeling and Relational Database Design

It is impossible to over-estimate the importance of database and some consider data storage to be the heart of an information system [10]. To efficiently store, update and retrieve the data, the database of application systems has to be well designed. The first step, of course, is to create a logical data model of the business information. One of the tools can be used is Oracle Data Modeler. Specifically, an Entity-Relationship (ER) model is constructed using Oracle Modeler and presented in the form of what is called the Logical and the Physical models. The Oracle database schema is produced based on the ER model. The second step in building applications is to create basic queries to retrieve the data. Queries are used to answer business questions and serve as the foundation for application forms and reports.

If the case is used as a stand-alone term project in relational database system design for a typical undergraduate database course, top-down modeling approach should be adopted. This approach includes gathering information about business requirements and the internal environment, and proceeds to define processes, a logical model of the data, one or more relational models, and one or more physical models for each relational model.

If the case is used as an on-going project, the deliverables from the applications above can also be used to design the database at the backend. Specifically, the object model and class diagram created from earlier stage can be used as input for data modeling and database design.

Teaching objectives:

1. Create an Entity Relationship Diagram by identifying entities, attributes, relationships and constraints from a set of requirements
2. Normalize the Entity Relationship Diagram to third Normal form
3. Enhance the Entity Relationship Diagram to utilize several data modeling techniques
4. Engineer the Entity Relationship Model into a relational database design
5. Implement database design

Proposed assignments:

1. Ask students to identify potential business domain objects
2. Require students to identify attributes for each object and relationships between objects.
3. Develop the appropriate logical data model (Class Diagram or Entity-Relationship Diagram)
4. Using Oracle Data Modeler to create the logical mode and physical model.
5. Generate SQL script files to create tables and insert sample date.
6. Write SQL statements for each business functions in the instructional case.

REFERENCES

[1] ABET (2017) Criteria for Accrediting Engineering Programs, 2016 – 2017. Retrieved from http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2016-2017/

[2] Amato, Christie H., and Louis H. Amato. "Enhancing student team effectiveness: Application of Myers-Briggs personality assessment in business courses." Journal of Marketing Education 27, no. 1 (2005): 41-51.

[3] El-Attar, Mohamed, Mahmoud O. Elish, Sajjad Mahmood, and James Miller. "Is in-depth object-oriented knowledge necessary to develop quality robustness diagrams?" *Journal of Software* 7, no. 11 (2012): 2538-2552.

[4] Fellenz, Martin R. "Toward fairness in assessing student team work: A protocol for peer evaluation of individual contributions." *Journal of Management Education* 30, no. 4 (2006): 570-591.

[5] Haas, Amie L., Robert W. Haas, and Thomas R. Wotruba. "The use of self-ratings and peer ratings to evaluate performances of student group members." *Journal of Marketing Education* 20, no. 3 (1998): 200-209.

[6] Lingard, Robert W. "Improving the teaching of teamwork skills in engineering and computer science." Systems, Cybernetics and Informatics 8, no. 6 (2010): 20-23.

[7] Marques, Maira, and Sergio F. Ochoa. "Improving teamwork in students software projects." In *Software Engineering Education and Training (CSEE&T), 2014 IEEE 27th Conference on*, pp. 99-108. IEEE, 2014.

[8]Murray, D. MGS351 Course Website(n.d.); University of Buffalo, http://mgt2.buffalo.edu/departments/mss/djmurray/mgs351/PeerEval.doc accessed at Nov. 20, 2016.

[9] No Magic, Inc. . MagicDraw UML, Version 19 Available at https://docs.nomagic.com/display/MD190/Robustness+diagram [Online]. Last Accessed February, 2018.

[10] Post, Gerald V. Designing & Building Business Applications with Oracle. McGraw-Hill/Irwin, 2004. pp. 143.

[11] Rosenberg, Doug, and Matt Stephens. "Use case driven object modeling with UML." *APress, Berkeley, USA* (2007).

[12] Rosenberg, Doug, Matthew Stephens, and Mark Collins-Cope. "Agile development with ICONIX process." *Jim Sumser*(2005).

[13] Sparx Systems. Enterprise Architect, Version 12.1. Available at http://sparxsystems.com/enterprise_architect_user_guide/12.1/building_models/generated_robustness_diagram.html Accessed February, 2018.

[14] Strong, James T., and Rolph E. Anderson. "Free-riding in group projects: Control mechanisms and preliminary data." *Journal of marketing education* 12, no. 2 (1990): 61-67.

[15] Van den Berg, Ineke, Wilfried Admiraal, and Albert Pilot. "Design principles and outcomes of peer assessment in higher education." *Studies in Higher education* 31, no. 03 (2006): 341-356.

[16] Visual Paradigm, "Visual Paradugm". Available at https://www.visual-paradigm.com/support/documents/vpuserguide/1283/177/6563_applystereot.html Last Accessed February, 2018.

APPENDIX

Appendix I Student Team Project Self-and Peer-Assessment Instrument

Semester: ________________ Your name:_________________
Course: __________________ Group/Project name: _________
Meeting time: ____________ Team number:

Guidelines:

- ✓ Read carefully the definition for each criterion and the explanation for each category.
- ✓ Evaluate based on member's typical work and behavior. Do not be influenced by unusual cases.
- ✓ Determine the category that best describes the member's accomplishments in that criterion.
- ✓ Fill out the assessment form listed below (1 to 5 or NA) for all of your team members. Make sure to include yourself.
- ✓ Complete comments on back to support your evaluation.

Member	Quality (average 4/5)	Timeliness (average 4/5)	Collaboration (average 4/5)	Interaction (average 4/5)	Attendance (average 4/5)	Responsibility (average 4/5)	Involvement (average 4/5)	Contribution (Total = 100)
Mary								
John								
Smith								

Appendix II Sample Self-and Peer- Assessment Matrices for Student Team Project

Criteria	**Possible Scores**				
	1	2	3	4 (Average)	**5**
Quality of Work: Team member provided accurate and complete work	Provides unacceptable work, fails to meet minimum requirements.	Partially meets minimum team or project requirements.	Provides work that meets minimum team or project requirements.	Provides work that partially exceeds project or team requirements.	Provides work that exceeds project or team requirements.
Consistency: Team member consistently offered high quality contributions	Rarely provides satisfied work	Occasionally offers satisfied work	Sometimes offers high quality contributions	Consistently produces high quality contributions	Always offer high quality work exceeding team or project expectations
Openness: Team member was open to listening others' opinions, allowing his/her ideas to be criticized	Failed to listen others' opinion	Occasionally open to listen to others	Sometimes open to listen to others	Most of time listened to others	Very open and happy to listen to others
Timeliness: The team member's timeliness of work.	Failed to meet deadlines set by the team.	Occasionally misses deadlines.	Regularly meets deadlines.	Meets deadlines and Sometimes ahead of schedule.	Always ahead of schedule.
Collaboration: The amount of support to other team members.	Failed to support other members.	Sometimes support to other members.	Regularly support to other team members.	Consistently support to other team members.	Consistently gives support more than expected.
Interaction: Behavior of team member to other team members.	Behavior is detrimental to team.	Behavior is inconsistent and occasionally distracts team meetings.	Regularly demonstrates appropriate team behavior	Consistently demonstrates appropriate team behavior.	Always demonstrates exemplary team behavior.
Communication: Timely and effective communication with other team members	Failed to communicate timely and effectively with other members	Occasionally communicated with other members	Sometimes communicated successfully	Successfully communicated with other members	Demonstrates exemplary communication skills.

Meeting Attendance: The team member's attendance at the meetings inside/outside class	Failed to attend team meetings.	Attended less than 1/3 of the team meetings.	Attended less than half of the team meetings.	Attended almost all of the team meetings.	Attended all of the team meetings.
Responsibility: The degree to which the member can be relied upon to complete a task.	Unwilling to carry out assigned tasks.	Sometimes carries out assigned tasks but never volunteers to do a task.	Carries out assigned tasks but no volunteer work.	Consistently carries out assigned tasks and sometimes volunteers for extra tasks.	Consistently carries out assigned tasks and always volunteers for other tasks.
Involvement: The team member participates in the exchange of information (brings outside knowledge to team).	Fails to participate in team discussions and fails to share relevant information.	Sometimes participates in team discussions and rarely contributes relevant materials.	Takes part in team discussions and shares relevant information.	Demonstrates initiatives, regularly participates in team discussion and sometimes exceeds expectations.	Consistently demonstrates initiative, exceeds team expectations for participation and consistently contributes relevant information.
Contribution: Consider the share of the work each team member participated	On a zero to 100 scale, rate the member's overall contribution to the team's work, both inside and outside of class. The total contribution of the team is 100. DO NOT simply add the seven scores above.				