Developing process models for an interdisciplinary project-based class

Conference Paper · May 2019

6 authors, including:

Marisa E. Exter
Purdue University
44 PUBLICATIONS **129** CITATIONS
SEE PROFILE

Iryna Ashby
Purdue University
27 PUBLICATIONS **33** CITATIONS
SEE PROFILE

Mohan Yang
Purdue University
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Tadd Farmer
Purdue University
5 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Developing process models for an interdisciplinary project-based class

Marisa Exter Iryna Ashby Mohan Yang Tadd Farmer
mexter@purdue.edu iashby@purdu.edu yang1178@purdue.edu farmer32@purdue.edu
Brantly McCord Umair Sarwar
mccord4@purdue.edu usarwar@purdue.edu
Purdue University, 100 N. University, West Lafayette, IN

1. Introduction

Developing high-quality educational software is a complex task, and typically requires a team of professionals with diverse expertise, including software engineering (SE), software development (SD), instructional design (ID), user experience design (UX), computer graphics (CG), and other specialists, based on goals and complexity of the design [1-3]. While each discipline has its own preferred process model(s), in an interdisciplinary effort the processes followed by specialists must overlap, interact, or be incorporated into a larger project process. To prepare students to effectively and efficiently collaborate within interdisciplinary teams, it is key to acquaint them with diverse disciplinary process models [3-4] and be able to align their own team process with a model that is sensitive to the needs of a particular interdisciplinary team, clients, and users [5]. However, traditional academic programs tend to be siloed and rarely incorporate interaction with students or faculty from other programs, much less knowledge about processes, techniques, and language used in other disciplines.

The purpose of this paper is to explore interdisciplinary students' understanding of and interaction with process models within a graduate project-based educational software design course through the following research questions: (1) What process models do interdisciplinary students propose when given the opportunity to adapt existing models or create their own? (2) How and why do students apply models in their projects? (3) How is this impacted by the course design?

2. Background

2.1. About the Course

Educational Software Design is a studio-model (e.g., [6-8]), semester-long interdisciplinary graduate course centered around an authentic client-proposed team project [2]. The primary goal of the course is to prepare students to work in interdisciplinary teams by: (1) Developing an understanding of language, design processes, and techniques used across ID, SE, SD, UX, and CG; (2) Collaborating with peers as part of a diverse student design team; and (3) Working through a systematic design process to create a prototype software application for a real client and users. A flipped classroom approach (e.g., [9]) incorporates weekly readings and online discussion assignments, designed to fill knowledge gaps of students with different disciplinary backgrounds and levels of experience. Discussion prompts require each student to practice new skills and techniques, and to share and discuss their ideas, reflections, and potential application to their team project. Students are encouraged to include sketches and link to outside resources. During the 3-hour long weekly studio time, teams are directed to build off of the discussion activities to drive their project design and deliverables forward. Class activities, guest speakers, and regular critique sessions also occur during class time at key points during the semester.

2.2. Teaching Process Models

As the course designer/instructor, the first author struggled with competing objectives. The first was to ensure that each student had the opportunity to develop and apply skills across all aspects of designing educational software (e.g. learner analysis (ID), persona and scenario generation (UX), requirements writing (SE), and usability testing (UX)). This was achieved through a series of weekly discussion activities and production of documentation similar to that used in industry, with drafts due at scheduled deadlines, essentially imposing a traditional linear process. The second was to encourage ideation, iteration, and feedback from various stakeholders to encourage thinking "outside-the-box." This was addressed through 3-week cycles of research and ideation activities; collection of client-, instructor-, and peer-feedback; and design/prototype development and documentation. The third objective was to expose students to process models used across disciplines and encourage them to think critically about when and why to use these, and how they interact in the design of educational software. Readings introduced students to a variety of SE, ID, and UX models, from linear waterfall-style models to rapid prototyping and Agile (e.g., [10-15]). A guest speaker presented on how Agile is used in her educational software team. At mid-semester, students participated in an online discussion that asked them to: 1) discuss what model their team had used so far, what worked well, and what challenges they faced; 2) create a diagram of a process for their team, and explain how and why this model was selected, adapted, or created; 3) discuss what they had learned about process models based on their readings and their development of this model; and 4) respond to one-another's posts, discussing advantages and challenges of their peers' proposed models or similar models found in literature. Individual end-of-semester reflection papers included a discussion of what they had learned about the process of designing educational software.

2. Methods

In 2018, 10 graduate students from ID, CG, UX, and several engineering disciplines participated in the course. Four are co-authors on this paper, along with the instructor and TA. We used a sequential two-phased approach to data collection and analysis. Phase 1 included analysis of discussion forum posts, project documentation, and individual reflection papers of 10 class members. Phase 2 data included structured written reflections by each of four student co-authors, along with comments made on one-another's reflections. We have completed a thematic analysis using three coders [16].

3. Findings

Findings are presented by theme. At mid-semester, students were prompted to describe their team process thus far

(Theme 1) and propose a process (Theme 2) in an online forum. They discussed throughout the week and reflected further at the end of the semester on ideal process traits (Theme 3) and impact of class structure on the process followed (Theme 4).

Theme 1: Use of process models: The discussion analysis showed that the concept of a process model was not something that students considered early in the semester, with one commenting, “I don’t think we’ve followed much of a process so far.” Others insisted that their adherence to the process model was not intentional, with “seems” being a common word used to describe what they thought their process was. However, when asked to come up with ideas as to what such a process model might look like, students tended to try to find similarities between what they had done and existing models reviewed for class, thus engaging in a more reductionist approach, e.g., “...if I had to choose one, I’d say it would resemble an ADDIE [linear ID process model] or waterfall-type model that is characterized by a linear approach to design.” Some students felt that their team process model could be better described as a combination of two or more models often with added opportunities for iteration. Several suggested that this may be due to the nature of their particular project. For example, one project team had clients who did not provide exact specifications, but rather left the problem and scope open-ended, e.g., “[the] scope and nature of the product we have is the only reason I chose a modified waterfall over a regular waterfall as till now, we have always been going back and changing stuff after interviews etc.”

When reflecting on potential process models for their projects, there were several major considerations, including prior experience within their home disciplines, e.g., “It really derived from the prior experience and education I had in the field of instructional design”; the need to reflect the project needs and focus on elements of the project, rather than prescribed sequences; and ongoing negotiations with interdisciplinary team members, e.g.,

> Because my partner was from a total [sic] different field than me and we have different approaches in designing as well as project management, we did not set up a certain model to follow in the beginning. Instead, we kind of let it flow, on the basis that the project was in control.

Students reflected on the similarities across models, e.g., “I could see glimmers of familiar process models in [ADDIE] (same basic order of design > develop > iterate).” They also reflected on how technical differences, nature of designed products, regulations, and disciplinary culture impacted the ways models were used and taught. Several indicated this would impact their own teaching going forward.

Theme 2: Types of Process Models Suggested by Students: We saw three major patterns in diagrams students produced when asked to propose a model for their team: (1) Adaptation of ADDIE, a typically linear instructional design model (e.g. Fig 1a), (2) Adaptation of SAM2 (ID model with 4 phases with looping sub-processes), substituting steps derived from SE, UX, and other models for the sub-loops (e.g. Fig 1b), and (3) Augmentation of SE-type waterfall models with additional steps and/or small loops at particular point(s) (e.g. Fig 1c). None of the students created a fully novel model.

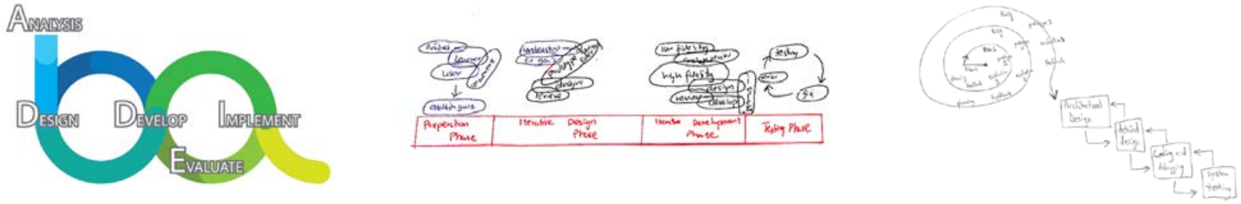

Figure 1a. ADDIE (linear ID model) through a “quasi-Agile” lens; b. SAM2 with “bubbles” adapted from other models; c. Waterfall-style SE model, with initial spiral to mitigate risks. [Descriptions based on student explanations]

Theme 3: Ideal Process Models: Reflections made later in the semester showed that the students believed a process model should be flexible (e.g., “There are too many moving pieces and parts that need to be addressed”); systematic and iterative (e.g., “I would definitely say it is a lot more systematic as the research you do ties in directly to the type of results of software”); rigorous and robust (e.g. “Software development must be rigorous and robust to routinely hit its marks”); and user-centered (e.g., “I learned how critical it is to design for the user. Everything you do is useless if it doesn’t cater to the end user you’re designing it for”). Students also reflected on how these activities impacted their understanding of design, and themselves as designers.

> The actual process was a reflection of how we, as designers, integrated our knowledge and prior experience into the design and development, and also a reflection of how we adapted to the unique challenges of our project, based on the guidance of the instructor, timeline and requirements of the course.

One student also noted that a process model is a way to get back on track when facing design challenges, yet it should be organic and not necessarily prescriptive to design processes.

Theme 4: Course Structure as a Process Model: In end-of-semester reflections, several students noted that their proposed process models were not necessarily authentic or fully representative of the process. While the instructors intended course deliverables to guide students to engage in all aspects of analysis, design, and usability testing, they also meant to encourage re-design and additional analysis through critique and negotiation of deadlines based on project needs. However, students felt that course activities and deadlines enforced a prescriptive, linear structure. Some students found it limiting, e.g.,

> The structure of the course (e.g., assignments, due dates, feedback) had a big impact on what we did and what we did not do. Sometimes the rigidity of the course prevented us from applying the iterative model of design we all seemed to appreciate.

Others reflected that the course structure helped them focus on important aspects of the process, for example completing a learner and needs analysis and integrating user and client feedback before proposing a final design solution.

> My instinct is to quickly sketch and scrap ideas until the dominant, reliable aspects rose to the top, but with the problem being so nebulous and large scale (ideally requiring some sense of curriculum, not just a witty fix-it solution), it was clear that rushing in wasn't going to get the job done, or worse, would disrupt the momentum I had in the analytical stage with my partner.

One mentioned "I think it was the negotiation among the team partners and the instructor leading to the actual design model."

4. Discussion and Implications

The goal of the course was to provide students from diverse disciplinary backgrounds with an authentic experience of educational software design. As prior research shows, the design of educational software is inherently interdisciplinary, requiring professionals to be able to understand and synthesize processes, approaches, and strategies to develop individual solutions [2]. The exploration of process models provides an opportunity to open the door to design processes and design thinking embedded across fields. Thus, understanding how students comprehend and negotiate process models may allow us to address and improve interdisciplinary learning experiences.

The analysis of the data showed that the selection, adaptation, or development of a predetermined process for software design did not come up organically, and was not part of proactive student team discussions. However, this was not necessarily due to limited student engagement. The reason for this in our case was two-fold: (1) often process models are predetermined by disciplinary conventions, resulting in little reflection on the variety of potential models prior to our discussion and (2) the actual course structure. During the first week of classes, the instructor covered main steps and deliverables expected from students. As a result, students may have expected to follow a process they perceived to be enforced within the course. However, the reflective processes both during (mid-semester discussion) and after the course (reflections) showed students' attempts to combine process models they had been exposed to. Students observed that process models bore similarities, even though their reflection appeared to be on a reductionist level of breaking the steps down, rather than seeing a process as a larger and more holistic system. These attempts at synthesis were limited as students tended to adjust existing models by adding steps or aspects of process models from other domains. Although they did not reach the level of reflection, critique, and creativity the instructor hoped to see, these discussions were helpful for overall student learning as they were able to make connections across disciplinary models, potentially impacting their own future design process. Two of our co-authors said that this will also impact their *teaching* of process back in their home departments.

Facing the challenging environment of an authentic ill-structured problem, students highlighted that design processes, and the process model itself, need to be sufficiently flexible to organically address the needs of the project, and more importantly of the users. This was not always possible when trying to adjust their processes to the deadlines for course deliverables, which led to many frustrations on the part of students, especially when they experienced difficulty balancing course requirements with their own sense of project needs and workflow.

Based on reflections on our own experience as both students and instructors, we suggest the following considerations - both for the next revision of this course and for others considering a similar approach:

- Interdisciplinary teams benefit from space to negotiate and establish their own process and timeline to align with their individual projects needs and limitations. This can be difficult to balance with the need to set deadlines for course deliverables. In our experience, even if the timeline was not *intended* to restrict students' pace, students *felt* it did and struggled with the perception that they were being held back from moving on to a solution, increasing frustration and decreasing motivation. At the same time, the instructor felt that without it students had a tendency to rush to a solution without dedicating the time needed for analysis, design ideation, and iteration.
- When faced with differing and even competing prior experience with process models, it is helpful to engage students in in-depth reflection to encourage them to understand the value and drawbacks of different types of models (e.g. waterfall vs iterative vs Agile), as well as the similarities and differences of models across disciplines. The mid-semester online discussion activity drove students to discuss process at a level never experienced, as most students indicated they were simply *told* to use specific models in their traditional, discipline-specific courses. However, student authors note that the process of writing this paper brought encouraged much deeper reflection and understanding than course activities, leading to our conclusion that discussions about process should be brought back multiple times throughout the semester.
- Allow students to negotiate course-imposed process model and deliverables to support student engagement and avoid stunting their progress. We are still considering how we might adjust our course design to accomplish this without reducing necessary scaffolding or place additional burden on students in an already complex, work-intensive course.
- Consider what you actually want to accomplish through having students engage with process models, and how this experience might impact students' own discovery process of larger learning outcomes in the class. In our discussions while writing this paper, we recognized that some of the most profound realizations came from student co-authors' reflecting on their own learning - including the realization that existing process models must be built from past experience of addressing project and client needs. They also found that this helped them develop a holistic view of their own project experience, and the points of interaction across disciplines represented within their teams. Such learning cannot be forced (or as one co-author expressed it, "hammered in"), but must be discovered through engaging with deep discussion in combination with the authentic project experience.

References

1. Exter, M. E. (2014). Comparing Educational Experiences and On-the-Job Needs of Educational Software Designers. *Proceedings of the 45rd ACM technical symposium on Computer Science Education (SIGCSE).* Atlanta, GA. DOI:10.1145/2538862.2538970
2. Exter, M., & Ashby, I. (2018). Preparing Today's Educational Software Developers. *Journal of Computing in Higher Education.* DOI: 10.1007/s12528-018-9198-9
3. Landes, D., Sedelmaier, Y, Pfeiffer, V., Mottok, J., & Hagel, G. (2012). Learning and teaching software process models. *Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON),* Marrakech (pp. 1-8)
4. Robillard, P., Kruchten, P., & d'Astous, P. (2001). Yoopeedoo (UPEDU): a process for teaching software process. *Proceedings 14th Conference on Software Engineering Education and Training. 'In search of a software engineering profession' (Cat. No.PR01059),* Charlotte, NC, USA, (pp. 18-26)
5. Lucena, P., Braz, A., Chicoria, A., & Tizzei, L. (2016). IBM design thinking software development framework. *Proceedings of the 7th Brazilian Workshop on Agile Methods (WBMA'2016), Curitiba.*
6. Cennamo, K., Douglas, S., Vernon, M., Brandt, C., Scott, B., Reimer, Y., & McGrath, M. (2011). Promoting creativity in the computer science design studio. *SIGCSE '11* (pp. 649–654). Dallas, Texas: ACM.
7. Lee, J., Kotonya, G., Whittle, J., & Bull, C. (2015). Software design studio: A practical example. *ICSE '15 Proceedings of the 37th International Conference on Software Engineering, Vol. 2.,* Florence, Italy, May 16-24. (pp. 389-397)
8. Nurkkala, T., & Brandle, S. (2011). Software studio: Teaching professional software engineering. *SIGSCE '11,* Dallas, Texas: ACM. (pp. 153–158).
9. Erdogmus, H., & Peraire C., (2017). Flipping a graduate-level software engineering foundations course. *ICSE-SEET '17 Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track,* Buenos Aires, Argentina, May 20-28, (pp. 23-32)
10. Allen, W. (2006). Overview and evolution of the ADDIE training system. *Advances in Developing Human Resources, 8*(4), 430-441
11. Allen, N., & Sites, R. (2015). *Leaving ADDIE for SAM: An agile model for developing the best learning experiences.* Retrieved on January 31, 2019 from https://www.alleninteractions.com/sam-process
12. Instructional Design Central. (2019). *Instructional design models.* Retrieved on January 31, 2019 from https://www.instructionaldesigncentral.com/instructionaldesignmodels
13. IDOE Design Kit. (n.d.). *The field guide to human-centered design.* Retrieved on January 31, 2019 from http://www.designkit.org/resources/1
14. McConnel, S. (1996). *Rapid development. Taming wild software schedules.* Redmond, WA: Microsoft Press
15. Tripp, S., & Bichelmeyer, B. (1990). Rapid Prototyping: An alternative instructional design strategy. *Educational Technology, Research and Development, 38*(1), 31-44
16. Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77-101