Designing and Assessing Authentic Software Development Projects in Undergraduate Computing Education

Christopher Hundhausen
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA,
chris.hundhausen@oregonstate.edu

Phillip Conrad
Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA, USA,
phtcon@ucsb.edu

Olusola Adesope
Educational Psychology Program, College of Education, Washington State University, Pullman, WA, USA,
olusola.adesope@wsu.edu

ABSTRACT

Team software development projects are a crucial component of undergraduate computing education. Studies of software developers in industry underscore the need for undergraduates to acquire authentic software development experiences in which they contribute to legacy code bases within collaborative software development contexts. Despite their clear educational value, team software development projects are notoriously challenging to implement and assess. Drawing on experience and relevant research results, this Research in Practice Project Activity (RIPPA) brings together computing educators and researchers to discuss, research, and innovate approaches to designing and assessing authentic software development projects. Participants are invited to run empirical studies in their own courses in which they (a) collect subjective and objective measures of team and individual progress in software projects, and (b) explore alternative pedagogical and assessment approaches, including those supported by the use of custom learning management and data analytics tools. By the end of the RIPPA, we aim to publish one or more research papers that advance the state-of-the-art in team software project pedagogy, supporting technology, and assessment based on the results of multi-institutional research studies.

CCS CONCEPTS

• **Software and its engineering** → Software creation and management; Collaboration in software development; • **Applied computing–Education–Collaborative Learning–Interactive learning environments**;

KEYWORDS

collaborative and team software development projects, assessment metrics, software engineering education, learning management and analytics environments

ACM Reference Format:
Christopher Hundhausen, Phillip Conrad, and Olusola Adesope. 2022. Designing and Assessing Authentic Software Development Projects in Undergraduate Computing Education. In *The United Kingdom and Ireland Computing Education Research (UKICER) Conference (UKICER2022), September 01, 02, 2022, Dublin, Ireland.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3555009.3555037

1 INTRODUCTION

Team software development projects are a crucial component of undergraduate computing education. Studies of software developers in industry underscore the need for undergraduates to acquire authentic software development experiences in which they develop solutions to realistic problems within collaborative software development contexts [1, 3, 6].

Despite their educational value, team software development projects are notoriously challenging to implement and assess [4, 5]. With respect to implementation, a key challenge is to find appropriate legacy software projects that provide students with authentic development experiences in which they are required to understand, modify, and debug code written by others. With respect to assessment, a key challenge lies in assessing individual contributions to team projects [2, 4, 7]. While some instructors may choose to give all team members the same grade, a better approach may be to assign individual grades based on some combination of subjective (e.g., instructor and peer assessments) and objective (e.g., data on individuals' code contributions and communication) measures.

Drawing on the literature and our ongoing research into authentic team software development projects, this Research in Practice Project Activity (RIPPA) brings together computing educators to identify barriers and opportunities; explore tradeoffs between objective and subjective approaches to assessing individual and team processes and products; and develop learning management and data analytics tools to support implementation and assessment. To support this work, participants are invited to conduct studies in their project courses. The results will be integrated into multi-institutional research papers that advance the current state of the art with respect to designing, implementing, and assessing undergraduate team software development projects.

2 GOALS AND OUTCOMES

This RIPPA has six key goals:

1. Discuss alternative pedagogical strategies for running team software development projects.
2. Explore the design of learning technologies to support implementation and assessment of team software development projects.
3. Identify strategies for formative and summative assessment of individual and team performance in team software development projects.

UKICER2022, September 01, 02, 2022, Dublin, Ireland

ACM ISBN 978-1-4503-9742-1/22/09.
https://doi.org/10.1145/3555009.3555037

4. Empirically explore relationships among different assessment measures
5. Empirically evaluate alternative pedagogical approaches, assessment approaches, and learning technologies for team software development projects.

In line with these goals, participants in this RIPPA should achieve the following outcomes:

- An ability to identify the barriers and opportunities associated with team software development projects.
- An appreciation of ways software tools can support the implementation and assessment of team software development projects.
- An ability to empirically evaluate and compare alternative strategies for designing, and assessing team software development projects.
- An ability to implement and empirically evaluate collaborative software development projects in their own courses and at their own institutions.
- An expanded network of educators and researchers with whom to collaborate on software engineering education projects.

3 RIPPA SCHEDULE

This RIPPA launches with a start-up workshop held during a session of UKICER 2022. That workshop consists of a series of presentations, small breakout group activities, and large group discussions. The workshop agenda is as follows:

1. Introductions
2. Team software development projects
   a. Barriers and opportunities
   b. Pedagogy design
   c. Breakout group activity
   d. Large group discussion
3. Using technology to support assessment and monitoring of team software development projects
   a. Barriers and opportunities
   b. Tech demo of a prototype learning management and assessment tool for team software development projects
   c. Breakout group activity
   d. Large group discussion
4. Wrap-Up: Looking to the future
   a. Discussion of future research steps
   b. Discussion of opportunities for future research collaboration

Following the workshop, participants are invited to contribute to the RIPPA by running empirical studies to collect research data in their software project courses. Those interested in doing so will attend a follow-up Zoom meeting in which we provide detailed instructions and guidelines for collecting research data in fall 2022 and spring 2023 courses with software projects. We anticipate that this meeting will be held in early October, 2022.

In June of 2023, we will hold an interim Zoom workshop in which participants may present and discuss results from studies run during the 2022-23 academic year. Based on these results, we will decide as a group how to proceed with writing up the results as one or more multi-institutional studies, and where to submit the paper(s).

In August of 2023, we will hold a capstone Zoom workshop in which we discuss the submission of a final version of one or more research papers. At this point, we will make any final plans necessary to bring the paper(s) to completion, with a goal of submitting the paper(s)s by the end of October, 2023.

4 TAKEAWAY KNOWLEDGE

For participants, the key takeaways of this RIPPA are anticipated to be as follows:

- Team software development projects prove challenging to implement and assess, especially when they involve legacy code bases.
- A variety of quantitative and qualitative metrics should be considered to assess both individual and team performance; these metrics should be triangulated to improve the reliability and validity of the assessments.
- Online learning environments can be designed to ease implementation logistics and to support continuous monitoring and assessment of team processes and progress.
- Empirical research in software project courses can provide significant insights into (a) how to effectively design and assess authentic software projects, (b) correlations between assessment metrics, and (c) how to leverage learning management and assessment tools to make team software projects easier to implement, monitor, and assess.

ACKNOWLEDGMENTS

This research is supported by the National Science Foundation under grant no. DUE-1915196.

REFERENCES

[1] Begel, A. and Simon, B. 2008. Novice software developers, all over again. *Proceeding of the Fourth International Workshop on Computing Education Research.* ACM. 3–14.

[2] Clear, T. 2009. Thinking Issues: the three p's of capstone project performance. *ACM SIGCSE Bulletin.* 41, 2 (Jun. 2009), 69–70. DOI:https://doi.org/10.1145/1595453.1595468.

[3] Craig, M., Conrad, P., Lynch, D., Lee, N. and Anthony, L. 2018. Listening to early career software developers. *Journal of Computing Sciences in Colleges.* 33, 4 (Apr. 2018), 138–149.

[4] Farrell, V., Ravalli, G., Farrell, G., Kindler, P. and Hall, D. 2012. Capstone project: fair, just and accountable assessment. *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* (New York, NY, USA, Jul. 2012), 168–173.

[5] Herbert, N. 2018. Reflections on 17 Years of ICT Capstone Project Coordination: Effective Strategies for Managing Clients, Teams and Assessment. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2018), 215–220.

[6] Sherriff, M. and Heckman, S. 2018. Capstones and Large Projects in Computing Education. *ACM Trans. Comput. Educ.* 18, 2 (Jul. 2018). DOI:https://doi.org/10.1145/3229882.

[7] Tafliovich, A., Petersen, A. and Campbell, J. 2015. On the Evaluation of Student Team Software Development Projects. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2015), 494–499.