

Connecting Entrepreneurial Mindset to Software Development

Prof. Ben Tribelhorn, University of Portland

Ben Tribelhorn teaches Computer Science at the University of Portland. His research includes machine learning for chaos in Lorenz systems, dynamic obstacle avoidance algorithms for unmanned aerial vehicles, improving software engineering pedagogy, and ethical concerns in artificial intelligence.

Dr. Heather Dillon, University of Washington Tacoma

Dr. Heather Dillon is Professor and Chair of Mechanical Engineering at the University of Washington Tacoma. Her research team is working on energy efficiency, renewable energy, fundamental heat transfer, and engineering education. Before joining academia, Heather Dillon worked for the Pacific Northwest National Laboratory (PNNL) as a senior research engineer.

Dr. Andrew M. Nuxoll, University of Portland

Andrew began his career as a software engineer. Lately (since 2007) he has been teaching computer science at the University of Portland. He is an active researcher in artificial general intelligence and computer science pedagogy. He also loves playing bridge and being outdoors.

Dr. Nicole C. Ralston, University of Portland

Dr. Nicole Ralston is an Assistant Professor and co-Director of the Multnomah County Partnership for Education Research (MCPER) in the School of Education at the University of Portland in Portland, Oregon. She received her Ph.D. in Educational Psychology with an emphasis in Measurement, Statistics, and Research Design from the University of Washington. An elementary school teacher at heart, she now teaches educational research and STEM methods to undergraduate and graduate students. Her research focus involves bringing active learning strategies to STEM, best practices of research-practice partnerships, and applied research in partnership.

Connecting Entrepreneurial Mindset to Software Development

Abstract

The purpose of this research was to develop classroom project modules that supported students in developing an entrepreneurial mindset in the context of software engineering. The modules connect the software development life-cycle from beginning to end including user focused requirements elicitation and evaluating quality attributes. The modules were implemented in a junior level software engineering course in 2019. A student survey was developed and measured student perceptions of learning objectives that tie directly into ABET accreditation outcomes. Students reported they found the activities most helpful for designing, building, and testing real world systems.

Qualitatively, we found that the student work completed in these modules to be higher quality than similar work submitted in prior years. Exam scores were improved when measuring students ability to create use cases, especially clarity and completeness. Student performance was greatly improved when writing use cases, especially clarity and completeness which was reflected in improved projects. Quantitatively, the same mindset objectives were assessed in other course modules as part a larger curriculum wide effort in Engineering. The numerical results indicate that the modules in this course outperformed other modules in the curriculum for most of the mindset objectives. Ultimately, the results indicate these types of modules may play an important role in entrepreneurial mindset development for computer science students.

Introduction

This paper describes a set of modules designed to develop an entrepreneurial mindset (EM) in computer science students. An entrepreneurial mindset is defined by the Kern Entrepreneurial Engineering Network (KEEN) as supporting students in developing advanced skills and mindsets to equip students to create personal, economic, and societal value [1]. Traditional computer science education often focuses on technical and collaboration skills. The mindset is a critical skill to develop in a software engineering course as many students can rapidly create novel applications as demonstrated by the rise and success of mobile apps. This class was redesigned to build a set of skills and mindsets to focus the students on creating value.

The modules described in this paper, and contextualized in purple in Figure 1, have been placed in a junior level Software Engineering course that is required for all majors and commonly taken by minors. The prerequisite courses are Data Structures and an Object Oriented Design course that includes a large semester long project. This course widely covers the software design life-cycle (SDLC) including requirements, project management, design methods, integration, quality assurance, testing, maintenance, and tools. It is also a key course used for ABET assessment. Our approach to software engineering is to teach the agile process using Scrum. Agile is a very general set of guidelines encouraging iterative software development, and Scrum is a variant of

agile that aims to be lightweight and is therefore appropriate in the context of a course. Specifically Scrum specifies that in each iteration of work, a sprint, should include planning, daily stand-ups, reviews, and retrospectives. It also specifies how to keep the requirements backlog organized along with roles to manage the process.

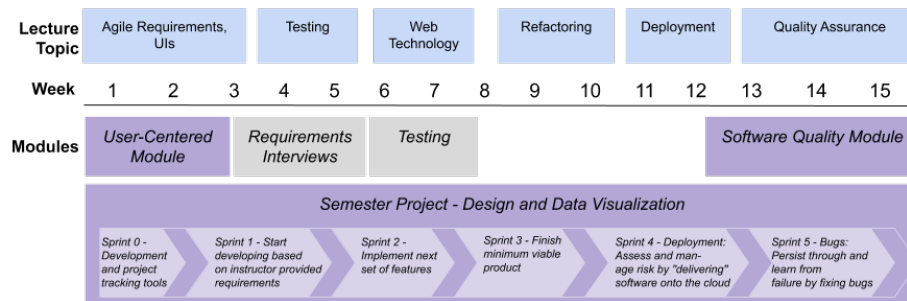


Figure 1: Summary of the EM modules (purple) discussed in this paper, in the context of the course timeline.

The modules developed had several learning objectives focused on the entrepreneurial mindset:

- Students will identify and describe links between course knowledge and real world systems. [connections]
- Students will improve their ability to self-reflect and evaluate preconceived ideas, thoughts, and accepted solutions to recognize opportunities. [curiosity]
- Students will increase their ability to understand the ramifications of design decisions. [connections]
- Students will integrate engineering solutions by creating use cases [connections]
- Students will appreciate the value of quality in software development. [creating value]
- Students will design, build, and test real world engineering systems. [creating value]
- Students will increase their ability to identify and evaluate sources of information. [connections]

The modules also contribute to a set of technical course objectives:

- Students will understand that to be a professional software developer, one must always be learning new technologies, quickly.
- Students will learn real world development technologies.
- Students will experience first-hand a test-driven, medium-sized software development project.
- Students will learn and apply the software development life-cycle.
- Students will reflect on professionalism in software development.
- Students will increase their ability to work in teams and communicate technical information.

The research goal of the project was to determine if a set of structured modules in a computer science course could help students enhance EM skills and mindsets. Many of the learning

objectives are important for computer science students, and tie directly to accreditation outcomes. The software engineering course is used to measure performance indicators for program outcomes #2 and #5. Although not used directly for ABET assessment, this course also strongly contributes to outcomes #1 and #3 while contributing to a smaller extent to #4.

Computer Science ABET Student Outcomes:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Communicate effectively in a variety of professional contexts.
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

This course modification is also part of a larger effort at the University of Portland to embed the entrepreneurial mindset across the curriculum [2, 3, 4, 5, 6]. Over the course of two years more than 30 different EM modules were tested in computer science, electrical engineering, mechanical engineering, and civil engineering classrooms. Many of the EM modules across the university used the same set of learning objectives for module design, allowing a larger sample for comparison with this class project.

Background

The entrepreneurial mindset (EM) has been studied by prior research teams. Some faculty have worked to build EM into each part of a curriculum, program [7], or university [8, 9]. Other groups have focused on understanding how students learn the entrepreneurial mindset and meta-cognition [10, 11]. Antonaci et al. [12] used advanced computer science tools to develop games that support the development of the entrepreneurial mindset in students.

Embedding a module focused on entrepreneurial mindset within an existing course has been explored by others using different tools [13, 11]. A summary of prior work on entrepreneurial mindset in the computer science classroom is shown in Table 1.

The modules described in this paper extend the prior work in several ways. The modules have been scaffolded in a single course to provide nuanced skills for the students. The methods include overlapping learning objectives between multiple modules for a rich student exploration.

Methods

Design

Our Software Engineering course is scheduled as a semester long (15 week) course as shown in Figure 1. There is a single semester long project that the students complete in groups of four to

Table 1: Summary of literature on entrepreneurial mindset modules in computer science classes.

Citation	Year	Course Type	Assessment	Methods and Topics
Bowers and Yerion [14]	2013	Intro to CS	Pre/Post Survey	Programming robots
Stanchev [15]	2015	Database Systems	Student Survey	Project-Based Learning
Estell et al. [16]	2016	Intro to Engineering	Student Survey	Software design
Tabrizi [17]	2017	Digital Systems	Student Survey	Jigsaw
Salas [18]	2017	Software Entrepreneurship	Student reflections	Software design
Tabrizi [19]	2017	Computer Architecture	Student Survey	Customer Model
Santiago and Guo [20]	2018	Digital Communication Systems	Student Survey	EM report
Tribelhorn and Nuxoll [21]	2021	Software Engineering	Instructor Review	Jigsaw
Present Work	2021	Software Engineering	Student Survey	Problem-based learning, Jigsaw

five students using a variant of the Agile development process. The large group size mirrors the larger teams found in industry and encourages development of communication skills. The course project has included various primary programming languages including Java, Python, and JavaScript. Past course projects include creating a web browser and creating web-based applications. In most offerings the assigned work has been setup with increasing requirements provided by the instructor in addition to students self-directing the final product based on a written requirements specification and successive sprint planning. These modules reinforce the learning outcomes discussed previously.

The first module, “Design and Data Visualization” is a framework for the course project and an application of Agile development into problem-based learning. It spans the entire semester and is used to inspire the students to focus on non-technical components of a software development project such as value creation. The other modules highlight the value creation aspects of software engineering. This entrepreneurial mindset motivates the students to think of the course project as more than an assignment but as a creative endeavor. The creation of these activities serves as a continuous reminder of the value that they will create in their own careers.

The times are estimated for a course size of about 20-25 students. Additional information on materials and handouts can be accessed by request to the corresponding author.

Module - Design and Data Visualization (Semester Project)

In this semester long project students work in teams to develop a software product over the course of a semester using techniques from Agile and SCRUM. The teams implement a complete software product, as well as develop a viable business plan for the product. The project culminates with an end of semester team presentation of the design, product, and the management process implemented. Students share lessons learned including both successes and failures along the way.

Scenario: *You were hired by the Mayor as the new director of informatics for the City of ____.* The

mayor tells you that she wants to improve the city functions using the data that they collect. She is frustrated because she can't get the city council to agree to some of the changes she wants to make because they don't understand the data. You and your team are given much of the data collected by various city agencies. The Mayor suggests that you should make a web-based application so that she can easily show the city council visuals of the data that you will be processing. She tells you that you'll get a bonus for every efficiency or insight that you identify.

Sprint Information: Each sprint runs approximately two weeks and sprint planning is done during class on first day of the sprint (last day of prior sprint). Students take tasks (product features) from the backlog or requirements document and add to their current sprint board. These tend to be large and vague, so they have to work to define each task better and then assign a work amount and a person to complete the work. This is how the students apply systems thinking to complex problems and how they begin to assess and manage risk. Assessing each task, estimation is done: this is a group based *Think-Share Inquiry and Debate*: What does it take to build this feature? Which features have the most value? Students typically rate work as a number of hours to complete and can use their hands rock-paper-scissors style to reveal their best guess for each task before then discussing to agree on an amount.

At the end of each sprint each team presents a live software demo and reviews their testing code. They always must answer what value did you create? Following the demos, retrospectives are conducted within each team. Students work to improve the process by reflecting on what when well, what did not go well, and what to do differently next sprint.

Module - User-centric value creation

Students design use cases for a real world software application to make a project more profitable for a fictional employer. The introduction to uses cases lecture includes an informal activity where the students create use cases for a simple product, such as an ATM machine. This lecture period is used to focus on the less obvious details of working with use cases, such as non-functional requirements (NFRs), pre-conditions, alternate flows, and refactoring use cases. In later class periods the following scenario is presented.

Scenario: You work for a well known conglomerate. Your product manager comes in and tells you that your department is losing funding to self-driving car software because your product is not exciting upper management. You won't be getting any bonuses or raises this year. Your manager thinks that you should create some new and exciting features for your software to get upper management to allocate more to your product. In twos or threes, brainstorm as many Use Cases as you can to extend your software. Do not create any use cases for basic functionality.

10-15 min - The software here is the course project, but it could be an existing software product if students don't have an applicable project. Carefully separate pairs from their other team members to avoid early ideation convergence. A template for a Use Case is projected. The scaffolding here is to focus them on interesting functionality for version 2 of the software. There is a temptation to rehash basic functionality that they have already implemented.

20 min - Have pairs present to the other pair/triplet in their team and consider the following: The median software developer earns about \$100,000/yr, which is about \$50/hour. *How much does*

each use case cost to implement? How much value does the use case represent? (State your assumptions.) How to value differentiation or quality?

20-30 min - Next each full team presents to the class and the whole class discusses the value of each use case in terms of both the cost to develop and the “value” (differentiation, profits, etc.). Are any of the use cases worth developing? In this discussion, it is helpful to force each team to give actual numbers even if they have no idea how to estimate. The instructor can offer suggestions of approximate numbers in the cases where they are struggling to come up with reasonable estimates. This helps the students to focus on important details that impact design including user numbers, subscription fees/ad revenue, hosting and delivery costs, etc.

Module - Software quality

This project examines software in terms of how well it is designed. Students work in expert groups to evaluate software quality attributes with respect to value creation using a modification of the powerful active learning Jigsaw technique [22]. This set of activities takes at least two class periods to complete.

5 min - This begins after a lecture and discussion of software security. Each student starts with ranking an instructor provided list of software programs based on their perception of its security. Any quality attribute or non-functional requirement (NFR) could be ranked; security is a good choice due to its importance and nebulous nature.

5 min - Combine students into pairs and have them re-rank the list. Pairing students in twos and threes within their project team works well.

10-15 min - Finally, the whole project team re-ranks the software list. The students write their ranked list on the board upon completion. Larger groups can really stretch this time with great discussions.

2-5 min - Instructor leads a brief discussion of how different groups perceived security, highlighting differences between groups. This leads to the introduction of valuation of a NFR.

20+ min - Students set their lists aside and each project group assigns each team member to a separate expert group. Each expert group is assigned to a different NFR such as: usability, reliability, availability, or performance. Expert teams must determine how to value that NFR. The output of each group are written criteria. This requires some guidance, after a few minutes hints about desiring criteria specified in a numeric form and weighted criteria can be provided. This can be finished as a homework assignment.

45 min - Students return to their project teams and try to value two specific software programs (assigned by instructor) from the list with respect to all of the expert group NFRs using the criteria from each expert group and generating at least some informal criteria for security. Can they put a monetary value on it?

Students struggle to develop numeric outputs for each of the topics. The authors find that using lots of scaffolding to slowly move them towards actual numbers is required. Optionally a handout with hints about user numbers, subscription fees, ad revenue, etc. could be provided. After about

20-30 min, the instructor will need to request most groups revise their numbers because they “assume” that each NFR is equally important. Finally, encourage the students to commit to a real number for both pieces of software. How much of a subscription fee is attributable to its software quality? How many users are attributable to quality?

15+ min - Each project group presents their estimated value for the two pieces of software.

10 min - Final discussion: How much of software value is from NFRs? How much value have they added in NFRs to their course project? An optional writing assignment can be added for them to reconsider their own course project. The authors have the students edit their README file with a ranked list of the project’s top NFRs achieved.

Assessment Methods

The overall results of this research involved two parts, an external observation and an optional student survey (IRB approved). The survey asked students to self-assess the specific learning outcomes on a five point Likert scale and write an open-ended response about each learning objective. The survey questions included the five learning objectives that mapped to the larger university project, allowing us to compare to other courses. The complete survey is included as an Appendix.

An example of the format for this survey question is shown in Figure 2. As these course projects are the primary output showcasing a range of student skills (design, reflection, teamwork, implementation, etc.), exam scores are not reflective of student work or knowledge gain. The authors present these results having seen a massive boost in the quality of student projects from implementing the modules described.

1. For the learning objective listed below, please rank the extent to which your capacity increased **during this class**. (circle one) Then, please elaborate with a specific example.

Question	Not at all	Very Little	To a Small Extent	To a Moderate Extent	To a Great Extent
To what extent has your ability to design, build, and test real world engineering systems increased during this class?	1	2	3	4	5
Describe a specific example of how your ability to design, build, and test real world engineering systems increased in this class.					

Figure 2: Example of student survey questions where students were asked to consider each learning objective on a Likert scale.

After the students completed the survey, the results were combined for two sections of the class, including two different instructors. The essay results were analyzed using a natural language processor in the R statistical program [23] and using standard qualitative methods. We used the standard statistical tests in R to perform a t-test between the survey data from this class and the larger collection of student responses across the university.

We used the text mining framework documented in more detail by Feinerer et al. [24]. We converted the text to lower case, removed special characters and numbers, and we removed the white-space. After the text was cleaned the software performed a text frequency analysis to determine the most common terms in the student comments. The results were displayed as a word cloud with the size of the word representing the frequency of occurrence in Figure 3.

Results

The results for the interventions were successful and suggest a few ways to improve the modules over time.

Module Results

Qualitatively, we found that the student work completed in these modules to be higher quality than similar work submitted in prior years. Exam scores were improved when measuring students ability to create use cases, especially clarity and completeness. This qualitative improvement was also noticed by the instructors of our senior capstone course. The module on quality attributes noticeably increased student commitment at the end of the project attributable to the perspective it provides the students. Projects from the course most recently have included web-apps for discovering and rating art installations in Hawaii, displaying comparative statistics about clean/dirty energy generation by city or region, visualizing various statistical information juxtaposing states, and an e-sports video game league client. The course long project module was particularly helpful in getting the students to generate the requirements and milestones for their projects with reduced guidance from the instructors. Finally, in terms of assessment, we believe that these modules greatly assisted in achieving the highest level of outcome in Bloom's taxonomy, "create," for the ABET outcome #2: "Design, implement, and evaluate a computing-based solution[...]" as the process of truly evaluating software outputs is the most difficult component implement in an educational setting.

Survey Results

The student survey was completed by 33 students over two sections, representing 75 percent of the possible respondents. Most of the respondents were computer science majors and four were double majors or CS minors. The summary of the results is shown in Table 2 and Figures 3-4.

The small samples size resulted in no statistically significant comparison based on a t-test. However the same mindset objectives were assessed in other course modules as part of the larger curriculum wide effort. These averages are also shown in Table 2, indicating that the modules in this course outperformed other modules in the curriculum for the mindset objectives except "Identify and evaluate sources of information."

The student essay portion of the survey was analyzed using a natural text processing algorithm as shown in Figure 3, and qualitatively as summarized below.

Learning Objective 1: Understand ramifications of design decisions

people can actually die as a result.”

Learning Objective 2: Identify and evaluate sources of information

- Students reported that the KEEN module helped them identify resources (47%; n = 15 students).

“I quickly learned how to identify information that would be useful.”
- Students described how the KEEN module helped them evaluate sources of information (38%; n = 12 students).

“A lot of the actual code learning in this class was based on researching for ourselves. I’ve learned to better recognize the more reliable sources.”

“Trying to solve problems led me to having a better idea of what sources are good are what are not.”
- Students described how the KEEN module helped them learn independently (41%; n = 13 students).

“Since a lot of the coding in the class was self-taught it forced me to go out and find the information myself which is very important to me and I found very valuable.”

Learning Objective 3: Ability to self-reflect on your own ideas

- Students reported that the KEEN module helped them self-reflect on their own ideas (63%; n = 20 students).

“I was able to see how crucial my input is.”

“I feel I had to go back and re-do things I did as a result of necessary changes, which made me reflect on the quality of what I did originally.”
- Students described how the KEEN module helped them develop and create new ideas (34%; n = 11 students).

“This course has helped me gain confidence in my abilities. As a result, it’s helped me feel more open to idea creation because these ideas seem more reachable now.”
- Students described how the KEEN module helped them learn the value in critiquing the ideas of their peers (56%; n = 18 students)

“Being judges and having peers rely on you is something I think is accurate to the real work team and having that experience does a great job in forcing you to reflect on yourself.”

Learning Objective 4: Design, build, and test real world engineering systems

- Students described how the KEEN module helped them understand the whole process of designing, building and testing engineering systems (50%; n = 12 students).

“I learned a lot of what it takes to engineer a whole system, different methods, teamwork, etc.”

”We built a website and I had absolutely no clue how to do that before. Now I can build one, test one, and improve one along with being able to work with a team.”
- Students described how the KEEN module helped them specifically understand the importance of testing engineering systems (33%; n = 8 students).

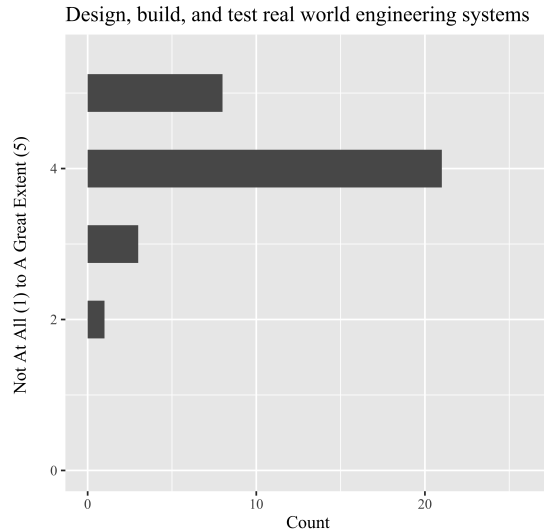


Figure 4: Student responses to the learning objective, “Design, build, and test real world engineering systems.”

“My increased knowledge in the testing filed makes me better understand the importance of reliability and solidly structured code.”

“Now I know how important testing is to the whole process.”

- Students reported that the KEEN module helped make a connection to real world engineering systems (29%; n = 7 students).

“I now know how to create a web application. Before this class most of the stuff I knew was not very important to real world stuff.”

Learning Objective 5: Identify links between course knowledge and real world systems

- Students described how the KEEN module helped make connections between the content and the real world (38%; n = 12).

“This course really incorporated real world situations and was one of the first classes to put me in a ‘work’ type setting.”

“I saw a lot of connection between the types of tasks we learned about in class and real-world testing.”

Discussion

A new project was developed for computer science students to help them develop specific skills to support an entrepreneurial mindset. Three new modules were incorporated in an existing course. The modules were well received by students and the projects developed demonstrated the students commitment to creating value. The qualitative improvement in the course projects completed by the students contrasted with prior years also supports the improvement in design thinking described in the results section. Based on these results, the authors recommend that other instructors incorporate the entrepreneurial mindset into their projects, possibly by adapting these modules to their existing curricular structure.

Students in a junior level computer science course tested this approach in the spring of 2019. Students reported that the module had been successful for most of the learning objectives, particularly for designing, building, and testing real world systems. This emphasis on real world systems encourages reflection on entrepreneurship and ultimately helps with a shift in mindset from a narrow coding focus to a broader, value-focused mentality. This is critical to student preparation given the ease of creating modern software as successful software is set apart when it provides excellent value to the end user. The authors believe that this work provides a novel focus to CS education research by focusing on mindset to achieve technical outcomes.

Future work will include adjusting small elements of the project to further improve student ability to learn independently and integrate information from multiple sources. The instructors plan to incorporate real-world clients into the projects, and continue to experiment with larger project group sizes including increasing to a class-wide project.

Acknowledgments

The funding for this work was provided by the Kern Entrepreneurial Engineering Network (KEEN) as part of a grant to the University of Portland. Special thanks to Jeff Welch and Rebecca Levison for data support.

References

- [1] KEEN, "KEEN - The Framework." [Online]. Available: <https://engineeringunleashed.com/mindset-matters/framework.aspx>
- [2] J. Farina, H. E. Dillon, R. D. Levison, and N. Ralston, "Increasing Student Curiosity with Cooling Systems," in *American Society for Engineering Education Annual Conference*, Montreal, Canada, 2020.
- [3] H. E. Dillon, J. M. Welch, N. Ralston, and R. D. Levison, "Creating an Engineering Action Plan for Ethics," in *American Society for Engineering Education Annual Conference*, Montreal, Canada, 2020.
- [4] R. Dvorak, H. E. Dillon, N. Ralston, and J. M. Welch, "Exploring Ethical Hacking from Multiple Viewpoints," in *Education Annual Conference*, Montreal, Canada, 2020.
- [5] C. J. Poor, H. E. Dillon, J. M. Welch, and N. C. Ralston, "Implementation of real-world class activities in an Introduction to Environmental Engineering Class," in *American Society for Engineering Education Annual Conference*, 2020.
- [6] K. Bieryla, N. A. Schulz, R. D. Levison, and H. E. Dillon, "Play-Doh and pendulums : making mass moment of inertia fun Play-Doh and pendulums : making mass moment of inertia fun," in *American Society for Engineering Education Annual Conference*, 2020.
- [7] J. Blake Hylton, D. Mikesell, J.-D. Yoder, and H. LeBlanc, "Working to Instill the Entrepreneurial Mindset Across the Curriculum," *Entrepreneurship Education and Pedagogy*, vol. 3, no. 1, pp. 86–106, jan 2020. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/2515127419870266>
- [8] G. Secundo, V. Ndou, and P. Del Vecchio, "Challenges for Instilling Entrepreneurial Mindset in Scientists and Engineers: What Works in European Universities?" *International Journal of Innovation and Technology Management*, vol. 13, no. 5, oct 2016.
- [9] D. Ridley, B. Davis, and I. Korovyakovskaya, "Entrepreneurial Mindset and the University Curriculum," *Journal of Higher Education Theory and Practice*, vol. 17, no. 2, apr 2017. [Online]. Available: <https://articlegateway.com/index.php/JHETP/article/view/1569>
- [10] J. M. Haynie, D. Shepherd, E. Mosakowski, and P. C. Earley, "A situated metacognitive model of the entrepreneurial mindset," *Journal of Business Venturing*, vol. 25, no. 2, pp. 217–229, 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.jbusvent.2008.10.001>
- [11] L. Bosman and S. Fernhaber, *Teaching the entrepreneurial mindset to engineers*. Springer International Publishing, aug 2017.

- [12] A. Antonaci, F. M. Dagnino, M. Ott, F. Bellotti, R. Berta, A. De Gloria, E. Lavagnino, M. Romero, M. Usart, and I. Mayer, “A gamified collaborative course in entrepreneurship: Focus on objectives and tools,” *Computers in Human Behavior*, vol. 51, pp. 1276–1283, oct 2015.
- [13] R. S. Harichandran, N. O. Erdil, M.-I. Carnasciali, J. Nocito-Gobel, and C. Li, “Developing an Entrepreneurial Mindset in Engineering Students Using Integrated E-Learning Modules.” *Advances in Engineering Education*, vol. 7, no. 1, 2018.
- [14] S. Bowers and K. Yerion, “Programming Personal Robots within an Introductory Computer Science Course for Engineering Majors,” *J. Comput. Sci. Coll.*, vol. 29, no. 1, pp. 133–139, oct 2013.
- [15] P. L. Stanchev, “Teaching the entrepreneurial Mindset in Database Class,” *Computer Science and Education in Computer Science*, vol. 11, no. 1, pp. 85–93, 2015.
- [16] J. K. Estell, D. Reeping, and H. Sapp, “Curiosity, connection, creating value: Improving service learning by applying the entrepreneurial mindset,” *ASEE Annual Conference and Exposition, Conference Proceedings*, vol. 2016-June, 2016.
- [17] N. Tabrizi, “Fostering an entrepreneurial mindset in ”Digital systems” class through a jigsaw-puzzle model,” in *Proceedings - Frontiers in Education Conference, FIE*, vol. 2017-October. Institute of Electrical and Electronics Engineers Inc., dec 2017, pp. 1–8.
- [18] R. P. Salas, “Teaching entrepreneurship in computer science: Lessons learned,” in *Proceedings - Frontiers in Education Conference, FIE*, vol. 2017-October. Institute of Electrical and Electronics Engineers Inc., dec 2017, pp. 1–7.
- [19] N. Tabrizi, “Fostering an entrepreneurial mindset in ’computer architecture and organization’ class through a producer-customer model,” in *IEEE Global Engineering Education Conference, EDUCON*. IEEE Computer Society, jun 2017, pp. 1843–1850.
- [20] J. M. Santiago and J. Guo, “Developing an entrepreneurial mindset using the KEEN framework for a digital communication system course,” *ASEE Annual Conference and Exposition, Conference Proceedings*, vol. 2018-June, 2018.
- [21] B. Tribelhorn and A. Nuxoll, “Using agile and active learning in software development curriculum,” in *2021 ASEE Annual Conference and Exposition, 2021*.
- [22] E. Aronson, N. Blaney, C. Stephin, J. Sikes, and M. Snapp, *The Jigsaw Classroom*. Beverly Hills, CA: Sage Publishing Company., 1978.
- [23] CRAN, “R Software,” 2019. [Online]. Available: <https://cran.r-project.org/>
- [24] I. Feinerer, K. Hornik, and D. Meyer, “Text mining infrastructure in R,” *Journal of Statistical Software*, vol. 25, no. 5, pp. 1–54, 2008.

Appendix

A Survey

The following pages illustrate the complete survey given to the students.

CS 341 Survey

This survey is part of a research study conducted by Drs. Ben Tribelhorn and Andy Nuxoll, Tim Doughty, Heather Dillon, and Nicole Ralston from the University of Portland. We hope to learn about the effectiveness of various projects and educational experiences in this course. If you agree to participate, please complete the survey below. If you do not want to participate, please do not complete this survey.

All data will be kept in a password protected computer without any link to your name. There are no anticipated risks to your participation in this survey. Participating in this research will help improve the courses and projects in the curriculum, and may be published anonymously in a conference or journal paper. However, we cannot guarantee that you personally will receive any benefits from this research. Your participation is voluntary, and your decision whether or not to participate will not affect your relationship with University of Portland. If you decide to participate, you are free to withdraw your consent and discontinue participation at any time without penalty.

If you have any questions about the study, please feel free to contact Nicole Ralston at ralston@up.edu. If you have questions regarding your rights as a research subject, please contact the IRB (IRB@up.edu).

Name: _____

Instructor: Tribelhorn _____ Nuxoll _____

Major: ME ___ CE ___ EE ___ CS ___ or CS Minor: ___ or Non-ENG Major (List): _____

1. For the learning objective listed below, please rank the extent to which your capacity increased **during this class**. (circle one) Then, please elaborate with a specific example.

Question	Not at all	Very Little	To a Small Extent	To a Moderate Extent	To a Great Extent
To what extent has your ability to design, build, and test real world engineering systems increased during this class?	1	2	3	4	5
Describe a specific example of how your ability to design, build, and test real world engineering systems increased in this class.					

2. For the learning objective listed below, please rank the extent to which your capacity increased **during this class**. (circle one) Then, please elaborate with a specific example.

Question	Not at all	Very Little	To a Small Extent	To a Moderate Extent	To a Great Extent
To what extent has your ability to identify links between course knowledge and real world systems increased during this class?	1	2	3	4	5
Describe a specific example of how your ability to identify links between course knowledge and real world systems increased in this class.					

3. For the learning objective listed below, please rank the extent to which your capacity increased **during this class**. (circle one) Then, please elaborate with a specific example.

Question	Not at all	Very Little	To a Small Extent	To a Moderate Extent	To a Great Extent
To what extent has your ability to self-reflect on your own ideas increased during this class?	1	2	3	4	5
Describe a specific example of how your ability to self-reflect on your own ideas increased in this class.					

4. For the learning objective listed below, please rank the extent to which your capacity increased **during this class**. (circle one) Then, please elaborate with a specific example.

Question	Not at all	Very Little	To a Small Extent	To a Moderate Extent	To a Great Extent
To what extent has your ability to understand ramifications of design decisions increased during this class?	1	2	3	4	5
Describe a specific example of how your ability to understand ramifications of design decisions increased in this class.					

5. For the learning objective listed below, please rank the extent to which your capacity increased **during this class**. (circle one) Then, please elaborate with a specific example.

Question	Not at all	Very Little	To a Small Extent	To a Moderate Extent	To a Great Extent
To what extent has your ability to identify and evaluate sources of information increased during this class?	1	2	3	4	5
Describe a specific example of how your ability to identify and evaluate sources of information increased in this class.					