Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
ASEE
Paper ID #45427

Ambiguity as a key experience acceleration mechanism in a sophomore systems engineering course

Dr. Alejandro Salado, The University of Arizona

Dr. Alejandro Salado is an associate professor of systems engineering with the Department of Systems and Industrial Engineering at the University of Arizona. His research focuses on unveiling the scientific foundations of systems engineering a

David Herring, The University of Arizona

Ambiguity as a key experience acceleration mechanism in a sophomore systems engineering course

Abstract

This paper presents an innovative educational approach used in a sophomore systems engineering course, where LEGO Mindstorms robots are integrated to accelerate the understanding of complex systems engineering concepts. While hands-on learning tools like LEGO Mindstorms are often used in engineering education, our approach uniquely emphasizes the unpredictability and complexity inherent in real-world systems engineering. Rather than focusing solely on technology or project completion, we incorporate controlled disruptions during exercises, such as modifying project requirements, changing team compositions, or removing key components from the kits. These disruptions simulate dynamic environments, requiring students to adapt, manage resource limitations, and navigate evolving constraints. This approach bridges the gap between theory and practice, allowing students to rapidly prototype, test, and observe the impacts of their engineering decisions in real time. This paper describes the instructional approach and focuses on how students responded to the learning activities as described in their reflective journals.

Keywords

Systems engineering, project-based learning, real world applications, student reflection

Introduction

The notion of scar tissue is well known among systems engineering practitioners. It represents the knowledge gained from navigating the ambiguity and unpredictability of projects. You've been there, you've suffered the consequences of your own decisions and the decisions of others. You've recovered and taken away the knowledge those experiences provide. Each setback leaves a scar, but from those scars forms protective tissue, a reminder of what went wrong and clues to avoid similar mistakes in the future. These scars improve judgment, enhance resilience, and guide how to approach new challenges [1].

For many years, the need for such depth of knowledge drawn from life experience was used as a justification for why systems engineering could not be taught to undergraduate students. However, programs such as those at West Point and Loughborough University have shown the value of introducing students to systems engineering principles early in their academic journeys. At West Point, the goal of their introductory systems engineering course is to inspire students to embrace the discipline, "anxious to grab their ball and glove and go out and have a catch – learning and practicing their engineering skills" [2]. Similarly, Loughborough University recognized the importance of developing a "systems thread" that integrates specialist areas and fosters accelerated learning of key skills, such as recovering from setbacks and working effectively in ambiguous situations [3]. In general, Muller and Bonnema [4] suggest to focus at the undergraduate level on creating awareness of system considerations, providing insights into available methods, and understanding "the ill-defined and multi-dimensional nature of system problems with uncertainties, unknowns, ambiguities, dynamics, and conflicting needs and goals".

To help students learn these skills, many programs have developed different types of active learning environments, whose benefits are well established [5], given that there is a significant delta between systems engineering concepts and their application [6, 7]. Approaches like project-based learning provide opportunities for students to internalize systems engineering concepts by applying them in real-world contexts. As Valerdi and Zonneshain note [8], many consider systems engineering to be a "contact sport," best learned by doing. This sentiment underscores the importance of immersive, hands-on experiences that allow students to grapple with uncertainty and build their problem-solving capabilities. Jonassen highlights the gap between the structured, convergent problems students solve in classrooms and the ill-structured, ambiguous problems they encounter in the workplace [9]. He emphasizes that solving textbook problems alone does not adequately prepare engineering graduates to address the complex challenges of real-world projects, which often involve conflicting goals, collaborative systems, and non-engineering constraints.

In this paper, we argue and show that many of these experiences can be safely and effectively simulated in an introductory systems engineering course at the undergraduate level, where students can begin to develop the *scar tissue* that they will need in a controlled environment. This approach helps students to better handle uncertainty, apply systems engineering practices in real world contexts, and prepare for the complex realities of the workplace. Within a safe learning environment, students learn the importance of these skills and internalize their importance in practice.

The course

The course, SIE 250, is an introductory systems engineering course that is mandatory for all undergraduate students majoring or minoring in systems engineering at the University of Arizona. It is a 3-credit hour course spread over 1 semester. The course is at the sophomore level, and students generally take it in the first semester in the major, as the freshmen year is common to all engineering students. The course introduces students to the discipline of systems engineering. Students engage in different lectures and hands-on activities to (1) experience the need for systems engineering within an engineering endeavor, (2) learn about the different concepts, methods, and processes that systems engineers can use to support an engineering endeavor, and (3) apply some of them to a realistic engineering project.

The learning outcomes are all at the awareness level, as it is intended to provide the necessary context for students to understand the *what* and the *why* of systems engineering. Specifically, having successfully completed this course, students are expected to be able to:

- *Explain what systems engineering is*. This includes explaining fundamental concepts of systems theory and systems science, fundamental concepts of systems thinking, foundations of systems engineering, lifecycle and lifecycle models, and taxonomies of kinds of systems.

- *Explain why systems engineering is important*. This includes explaining how the rise in system complexity gave birth to systems engineering, why the development of systems should be driven by stakeholder needs, why technical, business, and programmatic concerns must be addressed at once to facilitate development success, why problem definition, system architecture and design, and interface design and control are important, and why anticipating system integration, verification, validation, transition, operation, sustainment, and retirement are important.
- *Explain the role that systems engineering plays within an engineering effort*. This includes describing the relationship between systems engineering and other engineering disciplines, describing the relationship between systems engineering and other non-engineering disciplines, listing the different life cycle processes of concern for systems engineering, and explaining the roles of decision management, risk and opportunity management, planning, monitoring, and control, configuration and information control, and modeling and analysis within an engineering effort.
- *Explain what a systems engineer does*. This includes mapping the previous learning outcomes to the tasks of the individual systems engineer within a team.
- *Describe the fundamental concepts, methods, and processes that form the systems engineering discipline*. These include problem definition, system architecture and design, interfaces, system integration, system verification, system validation, and system transition, operation, sustainment, and retirement.

Satisfaction of the learning outcomes listed above lead to academic equivalency with INCOSE's Systems Engineering Professional (SEP).

The course is structured in three main phases.

Phase 1. This phase takes place during weeks 1-3 of the semester. The purpose of this phase is to let students experience the hurdles of engineering projects that make systems engineering valuable. This phase gives students an understanding, a context in which to reason about the application of systems engineering. It is a way to let them develop "scar tissue" in a safe environment. While no learning outcome is attained in this phase, it provides the underlying frame to enable them to learn systems engineering. Students are set in teams and tasked with an engineering problem. Each student is assigned a role within the project. The problem is ill- and vaguely defined. The engineering problem requires the design, build, and deployment of a mine-detection rover. The LEGO Mindstorm EV3 is used. During the project, several hurdles are injected into the team's work, as will be described later.

Phase 2. This phase takes place during weeks 4-11 of the semester. The purpose of this phase is to present students with systems engineering concepts. The phase is divided into three main blocks. The first block covers defining systems engineering, its importance, and the role of the systems engineer (Weeks 4-5). The second block covers executing systems engineering activities (Weeks 6-10). The third block covers managing the technical effort (Weeks 10-11).

Phase 3. This phase takes place during weeks 12-15 of the semester. The purpose of this phase is to let students experience the benefits of systems engineering. Students are set in teams again and asked to work on a similar engineering project as in Phase 1. This time though, (1) they are able to apply what they have learned about systems engineering and (2) they work within an MBSE/DE environment. In-class activities include discussions and additional lectures to strengthen the material presented in Phase 2, this time in the context of their own projects. Students are assessed via a personal journal where they are asked to document their experiences, reflections, and insights each week in the course. Particularly:

- During Phase 1, students report on their experience during the project: things that work, things that did not work as expected, things that they did not anticipate, etc.
- During Phase 2, students report on how they think the material they are learning in every class could have helped them in the project in Phase 1.
- During Phase 3, students report on their experiences during the project in relationship to both Phases 1 and 2.

The results and analysis provided later in this paper are based on student reflections during Phases 1 and 3.

The experiential activity

To enable scar tissue through the experience, surprise is a key aspect of the learning activity. We want students to experience ambiguity and its consequences, not just that someone tells them about them. To maintain surprise, two actions are taken. First, students agree to an informal Non-Disclosure Agreement (NDA), by which they commit to not discuss what happens in class with any students that have not taken the course yet. Of course, this cannot be enforced, but there is not much more that can be done to control the flow of information. Second, all class material, such as the detailed topics that will be covered in the course, are hidden from students during the first phase of the course. This is being done to prevent students from learning about systems engineering. We believe that the memorability of the experience increases the more naïve they are about real-life engineering problems. Once Phase 1 is completed, all material is open to the students.

In Phase 1, students are divided into groups of 5-6 students and given the assignment in the box below. The objective that the robot needs to achieve is defined vaguely intentionally. No other direction is given. The only constraint is the working dynamics, which are rigid to mimic the work of a large organization. This is done by limiting communication flows and the kind of contributions they can make. A clarification note is important here. While teams can work in a more fluid manner, we are not trying to represent here such a team. Instead, each student is effectively representing an organization of several individuals, where everyone working at once is (1) infeasible -hundreds or thousands of people- and (2) silos exist.

> During the next 3 weeks, you are asked to work on an engineering project in teams. The following instructions are critical to have an effective learning experience, so please read attentively and follow them tightly.
> I will organize you in teams. As a team, your objective is to design and build a rover that detects yellow sticky notes on the ground and emits a sound when detecting one of them. You will be limited to a LEGO Mindstorm EV3 kit that I will hand to you in class.

Each individual in the group will have a given role, and they will not be able to execute any task beyond their role. These are the roles:

- *Designer*. This person will design the robot made of Lego bricks.
- *Software developer*. This person will design the software that will be installed on the robot computer.
- *Manufacturing engineer*. This parson will build the Lego robot.
- *System integrator and tester*. This person will install the software in the Lego robot and test that it meets the objectives stated above.

You must follow these rules when working in the project:
1) You can only work in the classroom if not otherwise specified.
2) You cannot talk to each other in the classroom or outside of the classroom about the project if not otherwise specified by me.
3) Only the Designer is allowed to have access to the kit and the bricks. The rest of the team cannot have access to the kit. (At least yet.)
4) The Designer is not allowed to use any software to aid the design process. (E.g., no use of Lego Digital Designer or similar.)
5) The Designer cannot provide real-time feedback or instructions to the Manufacturing Engineer to build the robot.
6) The Software developer cannot provide real-time feedback or instructions to the System integrator and tester.

In between classes, you must document your insights, thoughts, and ideas in your Individual Journal, and submit it before each class. Please, use the template provided in this assignment. In this phase, you will not be evaluated on how well your robot works or how well you operated as a team. This is a baselining exercise. I am interested in you capturing your reflections during the experience. However, it is paramount that you still push to achieve the best solution to the problem. Just without cheating. The goal is that you play this as a board game or friendly competition, where winning is important, but cheating is not worth it.

The objective that the robot needs to achieve is defined vaguely. The activity ends with an operational test defined by the instructor at the time of conducting the test. No details about the operational test are provided to the students if they do not ask about them. No intermediate milestones were defined, and students were not tasked to plan and/or schedule their work explicitly. Lego Mindstorms are central to the activity. They consist of a central computer unit that can be controlled externally or through software and offers four input interfaces and four output interfaces. All mechanical interfaces are compatible with Lego bricks. As inputs, the Lego Mindstorm ev3 kit comes with external sensors and receivers, such as color sensors, light sensors, push buttons, and infrared receivers. As outputs, it offers different kinds of motors, light patterns, and sounds. The Lego Mindstorm can operate on batteries or plugged in, and it can be programmed through a cable or via Bluetooth.

The Lego Mindstorm EV3 was chosen as it offers opportunities to easily and affordably replicate key aspects of realistic engineering contexts [10] and has been successfully employed to teach systems engineering concepts (e.g., [3, 11-13]). Particularly for this course, the following capabilities and flexibility were deemed important:

- Most solutions require integrating hardware and software.

- It does not require domain expertise. For example, students do not need to have knowledge of electric circuit theory or dynamics to design and build working solutions. Even software functionality can be coded through a graphical interface without requiring knowledge of any programming language.
- There are ample, publicly available training materials.
- It does not require external tools to design and build the solutions, other than a regular computer to code the software and transfer it into the Lego Mindstorm's computer.
- There are no special needs for safety precautions during both building and use of the solutions.
- The solutions can be used in multiple environments, e.g., inside and outside of the classroom space.
- Because of the underlying Lego platform, building and unbuilding can be done at a fast pace, avoiding spending long times on non-value-added activities, such as those that domain engineers and technicians would do. For example, for learning systems engineering it suffices that the student experiences that a part must be rebuilt and that such a process takes time. It is not necessary for the student to spend a significant amount of time in the process of building and unbuilding.
- Huge flexibility to create solutions and decent capabilities to address challenging problems.
- Legacy platforms can be incorporated into the solution space, as there are different variants of the Lego Mindstorm, its software design environments, and its drivers, all with different levels and/or states of supportability by Lego.
- There is supporting software that can be used to aid the design and construction process, such as dedicated CAD tools and/or libraries, all of which are inexpensive.
- There is very low dependency on parts of the kits. Other than the main computer, losing (through loss or damage) a few parts (i.e., bricks, sensors, or drives), does not jeopardize the ability of the students to create solutions. This would just result in a more heavily constrained solution space.

During the work of the students, two categories of disruptions are introduced: extrinsic and intrinsic. Extrinsic disruptions are injected by the instructor, who changes the external conditions of the project. While the instructor has a predefined list of disruptions that can be injected, injections are decided on the spot for each team to maximize its learning impact. Students are not advised previously about these disruptions being injected. Intrinsic disruptions emerge naturally from the development effort and the integration of the work of the different engineers.

Since students are not accustomed to disruptions, particularly extrinsic ones, a safe environment (with regards to grades) is emphasized during the activity. Success in the technological solution is not evaluated and does not affect students' grades. To avoid disclosing information about the experience, as stated before, this "safety" is communicated by letting students know that whatever they do in Phase 1 does not contribute to their grades in the course, other than them submitting their written reflections. In Phase 2, students know what to expect in general terms. In this case, students are told that the success of the solution in meeting the technological requirements is not a factor in their grades, but their ability to use SE during the process, planning for and reacting to disruptions.

This is a major point of departure with respect to traditional experiential learning activities in engineering education. Most of them focus on students building solutions that solve a problem with set parameters or practicing iterative design skills; they do not try to replicate the normal and unpredictable disruptions that occur in practice, as described in the Background section.

However, in this activity the focus is on the process, the engineering journey. The quality of the results is not measured on how well the system met its original objectives, but how well the teams did SE work to overcome the disruptions that they faced. It is important to clarify that this is not to state that meeting technological objectives is not important; of course it is. But the goal of the course is to make students aware of what systems engineering is and why it is important; it is the living and experiencing of the scar tissue that the activity is seeking. The purpose of the activity is to afford students the opportunity to experience the continuous joy of failure until something may finally work.

As a final note, students use a Digital Engineering environment to support their work in Phase 2.

Methods

To explore this research question, we collected qualitative data in the form of student reflection journals. The purpose of this exploratory study was to reveal how students respond to solving an introductory engineering problem while encountering ambiguity and challenges that arise in real world engineering scenarios. We were also interested in how this course structure would impact students' feelings about systems engineering. IRB approval has been obtained for this study.

We chose to analyze reflection journals because they offered a direct view into students' firsthand experiences with the class structure. The journals allowed students to document their thoughts, challenges, and growth in their own words, providing qualitative data that reflected their thinking and feelings throughout the course. The journals provided ongoing reflections over the semester. This made them especially useful for studying how students adapted to new challenges and learned to apply systems engineering concepts over time.

Setting, participants, and data collection

Students had to consent to participate in the project. Consent was requested after the course was completed and all grades posted. Students did not know about the research project while taking the course. The journals of all students that consented to participate in the project, a total of 11, were analyzed in this study. The journals captured students' experiences across all three phases of the course, except for one journal, which included reflections only from Phase 3, the second robot-building project. The journals were anonymized to protect student identities before analysis.

Data Analysis

The student reflection journals were analyzed after the course ended, so there was no opportunity to ask follow-up questions or clarify responses. The coding framework was developed based on topics and themes that emerged during an initial review of the journals and guided by the course goals. The framework focused on the challenges students faced (external and internal problems), their emotional responses, their approaches to organizing work and applying systems engineering concepts, and their broader reflections on the building process. These codes were chosen to align with the study focus on how students adapt to ambiguity and apply systems engineering principles. We applied descriptive codes to phrases and longer related segments, such as sentences or groups of sentences. The coding framework with definitions and examples is presented in Table 1.

Table 1: Coding Framework

Code	Definition	Operationalized Definition	Example
External Problems	Disruptions introduced intentionally by the instructor or caused by external factors.	Includes missing parts, changing requirements, environmental challenges (e.g., gravel terrain).	"There was a structural integrity issue with our tracks so we couldn't use them."
Internal Problems	Challenges arising from team dynamics or interpersonal interactions.	Includes communication issues, role conflicts, absenteeism.	"Group member with the robot was missing."
Feelings	Expressions of positive or negative emotional responses.	Positive (e.g., excitement, confidence, optimism); Negative (e.g., stress, doubt, confusion, overwhelm).	"That was a great act of systems engineering on my end" (confidence) "I'm having a hard time verbalizing what's going on inside my head." (struggle)
Organization of Work / Application of Learning	Actions or strategies related to task management or collaboration, including evidence of systems engineering concepts being applied	Includes parallelization, task delegation, use of tools (e.g., Jira, shared documents), verification/validation, baselining, testing protocols.	"In our group, we have four separate roles." "I finished creating all of our tasks in Jira."

Reflections on Process	Students' reflective insights into their learning or project experience.	Includes realizations about systems thinking, improvements over time, or broader lessons learned.	"I really learned how to be okay with ambiguity, got better working with a team, and am not afraid to ask questions."

After the initial coding was applied to entries from Phase 1 (January 12-January 26) and Phase 3 (April 15-May 1) and the students' final reflections on the class, we performed separate frequency tracking on the results from Phase 1 and Phase 3. We also included specific subcategories for each code, which are listed in Table 2.

Table 2: Frequency Tracking

Code	**Subcategories Tracked**
External Problems	Missing/Unusable Parts, Changing Requirements, Environmental Challenges, Changing Team Members, Unexpected Product Tests, Hardware Issues, Software Issues
Internal Problems	Communication Issues, Role Conflicts, Absenteeism
Feelings*	Positive and Negative Feelings in the categories of Setbacks/Work, Class Structure, Systems Engineering
Organization of Work / Application of Learning	Serialization, Parallelization, Task Delegation/Scheduling, Use of Tools (e.g. Jira, Violet), Verification and Validation, Testing and Testing Protocols, Direct Instruction (during the build cycles), Tutorials, Baselining, Revision of Plans
Reflections on Process	Better Communication, Focus on Learning (in the class), Dealing with Ambiguity, Time Management, Team Structure, Systems Thinking

*Positive feelings: resilience, optimism, confidence, enjoyment/excitement, pride, comfort; negative feelings: overwhelm, stress/struggle, doubt, worry, confusion, pessimism, annoyance, sadness

Results

The data analysis revealed a few interesting insights about how students approached the two robot-building assignments differently, how they felt about the class itself, and how the class influenced their opinion of systems engineering.

In both Phase 1 and Phase 3, all students encountered and identified problems during the building process. In Phase 1, external problems were common, with 90% of the students recording issues such as changing team members, noted by more than half of the students, and missing or unusable parts, mentioned by almost half. Student 3 described their frustration with these disruptions, stating, "I did get a bit overwhelmed during class with all of the constraints…. I'm still confused as to why [the instructor changed the Lego kits we received]" (1/19/24). Internal problems related to team dynamics were also common, with 70% of students reporting challenges. Communication and absenteeism were most frequently cited, each appearing in the reflections of four students.

In Phase 3, 82% of students identified at least one external problem. Software issues were the most common, which seven students mentioned, followed by changing requirements, which four students noted. The increase in software-related issues is partly the result of students using a new software platform with limited documentation during this phase of the project. The introduction of this software as part of a larger adoption by the engineering program presented students with an authentic real-world challenge. Internal problems were even more widespread, with 91% of students reporting difficulties. Communication issues, mentioned by six students, and absenteeism, identified by five, remained the most frequently cited challenges during this phase.

While the types of problems students faced in both phases were similar, their responses changed after instruction on systems engineering concepts. In Phase 1, students described organizing their work in basic terms, with eight students highlighting the sequential nature of task completion, even without explicitly using the term serialization. Most students (90%) discussed task delegation, and six mentioned testing their robot. Student 8 reflected on their team's approach, stating that the results of the first build "showed us a lot about how communication and reciprocation of ideas and the overall objective could have been done more efficiently" (1/24/24). By Phase 3, students not only continued to discuss task delegation and testing, now cited by seven and five students, respectively, but also incorporated advanced practices. Five students described using tools like Jira, Violet, and shared documents to support parallelized task delegation and address issues such as absenteeism and communication breakdowns. Additionally, six students employed verification and validation processes, and two introduced baselining. None of these concepts were mentioned in Phase 1. These changes demonstrate how students applied the systems engineering skills they had learned to address the challenges more effectively.

When reflecting on the class and the process of completing the robot builds, more than half of the students emphasized the importance of better communication for achieving success. Six students highlighted communication as a critical factor, while four noted the significance of ambiguity. These students either expressed greater comfort working in ambiguous situations or recognized the value of developing skills to navigate such environments effectively. As Student 5 explained, "This has been a trial-and-error learning process. The frustration mixed with uncertainty has been a little stressful, but at least I know I work well under stress and pressure" (4/17/24). This reflection underscores how the course fostered resilience by allowing students to engage with ambiguity in a supportive environment.

Some of the most insightful findings from the study came from how students expressed their feelings during the process. In Phase 1, the majority of students (80%) expressed at least one positive emotion, resulting in a total of 16 positive expressions. Negative emotions were far less common, with only three students mentioning them, contributing to a total of six negative expressions. Importantly, no student expressed only negative emotions during this phase: those who reported negative feelings also described positive experiences. In Phase 3, students were generally more expressive, and the balance between positive and negative emotions shifted. Ten students (91%) conveyed positive feelings, accounting for 24 instances, while nine (82%) reported negative feelings, with 26 instances overall.

The reasons behind this shift are not entirely clear. It is possible that students felt more comfortable sharing emotions later in the semester, or that specific aspects of the second project produced more frustration. External issues, such as the pressures of the semester's end, could have also influenced these changes. This shift should be explored in future research. However, it is noteworthy that most students (82%) expressed optimism about the project in Phase 3, and only one student reported negative feelings without also describing positive experiences.

Student reflections on the structure of the class and systems engineering as a discipline were overwhelmingly positive. Among the eight students who expressed feelings about the class structure, all conveyed positive sentiments, frequently mentioning excitement and enjoyment. They particularly appreciated the emphasis on learning over grades and the supportive environment that encouraged exploration without fear of failure. As Student 10 noted in their final reflection, "This class has sparked a significant interest in the learning process for systems engineering and I will keep in mind the 3 takeaways that our professor gave us, which I think can be applied beyond our major: embrace ambiguity, don't stop asking questions, read as much as you can. In conclusion, this was an incredible class and I feel it has given me the blocks to build the proper mindset and attitude to succeed." While two students initially reported negative feelings, both feeling overwhelmed and one also worried at the start of the class, both described these emotions being replaced by positive feelings as the semester progressed.

Student perceptions of systems engineering as a discipline were unanimously positive among the nine students who shared their feelings on this topic. Most expressed excitement and optimism about the field, highlighting a newfound appreciation for its principles and applications. Two students initially reported feeling confused at the beginning of the course, but both noted that their understanding improved significantly by the end of the semester.

Limitations

This study has several limitations that should be considered when interpreting the results. Most significantly, the dataset is relatively small, consisting of only 11 journals from 11 students, with one journal missing data from Phases 1 and 2. The small sample size limits the generalizability of the findings and may not capture the full diversity of student experiences. While the exploratory nature of the study allowed us to gain valuable insights, future research should try to collect data from larger cohorts or examine data from successive classes to strengthen the validity of the results.

The guidelines provided to students for their journal reflections were intentionally broad, asking students to document their experiences, reflections, and insights during the course. While this open-ended approach encouraged authentic and diverse responses, it also resulted in variability in the level of detail and consistency across the journals. For example, some students focused only on the facts of what they accomplished in each class while others reflected in detail on their own feelings and the feelings of their teammates.

Another limitation of this study is that the journals were collected and anonymized after the course ended. Because of this, there was no opportunity to ask follow-up questions or clarify ambiguous responses. Without the ability to interact with participants directly, it was difficult to explore specific reflections in greater depth or gather additional context, which may have limited the richness of the qualitative data.

Also, the reflective journals were the only source of data for this study. Incorporating other data sources, such as observations of student interactions, interviews, or quantitative surveys on specific topics could provide a more comprehensive understanding of how students responded to the inclusion of real-world challenges and applied systems engineering concepts to solve those problems.

As noted above, the study did not control for potential external factors, such as other academic or personal sources of stress, that may have influenced students' experiences and reflections, particularly near the end of the semester. These factors could have contributed to the increased expression of both positive and negative emotions observed during Phase 3.

Despite these limitations, this study provides valuable preliminary insights into how students engage with ambiguity and real-world challenges in an engineering classroom. These findings suggest several avenues for future research, including exploring how more structured reflection prompts, larger datasets, and additional data sources could enhance our understanding of student learning and adaptation in similar educational settings.

Discussion

This study shows how an introductory systems engineering course can simulate the ambiguity and unpredictability of real-world projects, providing students with opportunities to develop the “scar tissue” essential for professional success. Through deliberate disruptions and challenges, students experienced setbacks that required them to adapt, learn, and apply systems engineering principles. By Phase 3, students demonstrated growth in their abilities to manage complexity, using tools like Jira and verification and validation processes to address challenges. Such experiences align with findings that real-world engineering problems are often complex, ambiguous, and shaped by organizational dynamics [14]. These challenges are essential to bridge the gap between academic preparation and professional practice.

Experiential learning also emerged as a key driver of student development. Hands-on, project-based environments have been shown to foster critical engineering skills such as teamwork, communication, and problem-solving [15]. In this course, students engaged with realistic, open-ended challenges that demanded collaboration, iteration, and reflection, an approach consistent with the principles of effective engineering education [15]. These activities not only integrated theoretical knowledge with practical application but also prepared students for the realities of professional engineering.

The emotional shifts observed across the phases further support the value of this approach. While frustration and uncertainty remained, students increasingly expressed confidence and optimism as they applied new strategies and accepted ambiguity. These reflections highlight how a safe, supportive environment allows students to recover from setbacks and internalize key lessons without fear of failure. However, the study also presents future opportunities to further explore these findings, such as refining reflection prompts to capture more consistent insights.

Conclusion

This study highlights the potential for systems engineering education to simulate real-world challenges, enabling students to begin developing the "scar tissue" essential for professional success. By engaging with ambiguity and unpredictability in a controlled environment, students learned to recover from setbacks, anticipate the unexpected, and apply systems engineering practices to address these challenges. The positive reflections on the course structure and systems engineering as a discipline show the value of this approach in preparing students for work outside the classroom.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. CMMI-2205468.

References

[1] A. Pyster, N. Hutchison, and D. Henry, *The Paradoxical Mindset of Systems Engineers*. Hoboken, NJ, USA: John Wiley and Son, Inc., 2018.

[2] D. J. McCarthy, W. J. McFadden, and M. L. McGinnis, "5.3.2 Put Me in Coach; I'm Ready to Play!: A Discussion of an Evolving Curriculum in Systems Engineering," *INCOSE International Symposium,* vol. 13, no. 1, pp. 493-501, 2003, doi: https://doi.org/10.1002/j.2334-5837.2003.tb02635.x.

[3] S. Goodlass, "5.2.1 Can Systems Engineering be taught at Undergraduate Level?," *INCOSE International Symposium,* vol. 14, no. 1, pp. 945-955, 2004, doi: https://doi.org/10.1002/j.2334-5837.2004.tb00547.x.

[4] G. Muller and G. M. Bonnema, "Teaching Systems Engineering to Undergraduates; Experiences and Considerations," *INCOSE International Symposium,* vol. 23, no. 1, pp. 98-111, 2013, doi: https://doi.org/10.1002/j.2334-5837.2013.tb03006.x.

[5] S. Freeman *et al.*, "Active learning increases student performance in science, engineering, and mathematics," *Proceedings of the National Academy of Sciences,* vol. 111, no. 23, pp. 8410-8415, 2014, doi: doi:10.1073/pnas.1319030111.

[6] T. McDermott and A. Salado, "A perspective on systems thinking, architecting, and art," *Systems Research and Behavioral Science,* vol. 36, no. 5, pp. 648-655, 2019, doi: 10.1002/sres.2622.

[7] A. Salado, T. McDermott, K. Davis, and A. Moral, "Why Not Teaching Systems Architecture as a Studio Art Class?," Cham, 2019: Springer International Publishing, in Systems Engineering in Context, pp. 267-278.

[8] R. Valerdi and A. Zonnenshain, "Teaching Them How to Fish: Industry-Focused Student Projects in Systems Engineering," *INCOSE International Symposium,* vol. 22, no. 1, pp. 2188-2195, 2012, doi: https://doi.org/10.1002/j.2334-5837.2012.tb01466.x.
[9] D. H. Jonassen, "Engineers as Problem Solvers," in *Cambridge Handbook of Engineering Education Research*, A. Johri and B. M. Olds Eds. Cambridge: Cambridge University Press, 2014, pp. 103-118.
[10] E. Danahy, E. Wang, J. Brockman, A. Carberry, B. Shapiro, and C. B. Rogers, "LEGO-based Robotics in Higher Education: 15 Years of Student Creativity," *International Journal of Advanced Robotic Systems,* vol. 11, no. 2, p. 27, 2014, doi: 10.5772/58249.
[11] P. T. Grogan and O. L. d. Weck, "LEGO Product Design and Manufacturing Simulations for Engineering Design and Systems Engineering Education," Tampa, Florida, 2019/06/15, 2019. [Online]. Available: https://peer.asee.org/33056.
[12] H. A. Hahn, "What Can You Learn About Systems Engineering by Building a Lego™ Car?," *INCOSE International Symposium,* vol. 28, no. 1, pp. 158-172, 2018, doi: https://doi.org/10.1002/j.2334-5837.2018.00474.x.
[13] C. B. Nielsen and P. Adams, "Active learning via LEGO MINDSTORMS in Systems Engineering education," in *2015 IEEE International Symposium on Systems Engineering (ISSE)*, 28-30 Sept. 2015 2015, pp. 489-495, doi: 10.1109/SysEng.2015.7302802.
[14] S. R. Brunhaver, R. F. Korte, S. R. Barley, and S. D. Sheppard, "Bridging the gaps between engineering education and practice," in *Engineering in a global economy*, R. Freeman and H. Salzman Eds. Chicago, IL: University of Chicago Press, 2016.
[15] A. J. Dutson, R. H. Todd, S. P. Magleby, and C. D. Sorensen, "A Review of Literature on Teaching Engineering Design Through Project-Oriented Capstone Courses," *Journal of Engineering Education,* vol. 86, no. 1, pp. 17-28, 1997, doi: https://doi.org/10.1002/j.2168-9830.1997.tb00260.x.