# Case Studies in Applying Design Thinking to Course Design in Computer Engineering

Diane T. Rover, Henry Duwe, Phillip H. Jones, Nicholas D. Fila, Mani Mina
*Electrical and Computer Engineering*
*Iowa State University*
Ames, IA USA
{drover, duwe, phjones, nfila, mmina}@iastate.edu

***Abstract*— This Innovative Practice Full Paper describes case studies from an instructional design process based on design thinking, illustrating tools used during stages of design. Instructional teams investigated the potential relevance of design thinking in engineering course design in electrical and computer engineering. Two teams of educators used a design thinking process in the redesign of two computer engineering courses, one in embedded systems and one in computer organization and architecture. The process of applying design thinking methods and tools was led by a facilitator with expertise in design thinking and electrical and computer engineering. The process leveraged specific tools and collaboration. This paper presents examples from each course, focusing on the design thinking tools used by the instructors and team members, highlighting what design thinking looks like when applied in this setting, and giving specific examples. The purpose is to suggest strategies and provide information and guidance for educators to use tools in their own course design efforts.**

***Keywords—design thinking, instructional design, processor design, embedded system***

## I. Introduction

We have been exploring collaborative course (re)design strategies in the Electrical and Computer Engineering Department at a large university in the midwestern United States. Cross-functional teams referred to as x-teams were formed for each course being redesigned. Strategies used by x-teams borrowed from the rich set of design thinking processes and tools. As such, the teams not only included the instructors teaching each course, but also a design thinking facilitator to aid the faculty in adapting and applying design thinking practices to engineering course (re)design. This process was followed in two core computer engineering courses over five plus years. While the team compositions ebbed and flowed, the focus on using design thinking processes remained.

Prior works cover various aspects of the design thinking process. However, there is little work that describes how design thinking can be operationalized in (re)designing engineering courses as a team—i.e., what does design thinking "look like" in a course. In practice, engineering educators may struggle with using key design thinking practices such as empathizing with users, framing design problems based on user needs, ideating a variety of design concepts, and engaging in frequent cycles of low-fidelity (lo-fi) prototyping and user testing. The teams struggled with these challenges and, at the start, did not even have a clear understanding what the "product" they were designing was and who the "users" of that product were[1].

This process was not without friction and setbacks—not all the many different prototypes actually integrated into the courses worked out and most required iterative refinement. However, through open-minded persistent collaboration with a cross-functional team, including other instructors, teaching assistants, and a design thinking facilitator, the process has proved effective from the instructors' perspective. This paper specifically highlights successful threads of design thinking that weave their way across the stages of design thinking using a diverse set of design thinking tools adapted to our context. The key insights come from how the design tools were adapted and linked together throughout a truly iterative design process. Our intent is to provide concrete inspiration for future users of design thinking in course (re)design by suggesting strategies and providing concrete information on how they can be applied.

## II. Design Thinking Background and Framework

Design thinking is a user-centered, creative, and iterative framework that can be used to address complex and challenging problems and create change for users and communities [1][2][3][4]. Design thinking is based on the mindsets and practices of expert designers [3][4]. Recent decades have seen efforts to translate these mindsets and practices into a variety of organizations and fields [5][6]. While successes have been reported, including in applications in education [7], scholars have noted differences in discourses and framings between these practical applications and the empirically grounded work in the design community [6]. Such dissonance creates challenges for both applying design thinking in new domains in authentic ways and understanding the concept of design thinking across domains. For example, Johansson-Skoldberg and colleagues [6] note that, in the management domain, competing discourses have contributed to a framing of design thinking that emphasizes creativity but minimizes other key aspects and presents design

This material is based upon work supported by the National Science Foundation (NSF) under awards 1623125 and 2144757. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

[1] Ultimately, the teams reached a loose consensus that the "product" was a course experience for students as the primary "users."

thinking as a de-contextualized toolkit that may ignore the expertise necessary to effectively use such tools.

Toolkits have been a prominent way of translating design thinking to education and curriculum design [7][8]. Such toolkits seem to heed Johansson-Skoldberg and colleagues' findings by (1) organizing tools by the design practices, which are situated within overarching methodology, and (2) imbuing the toolkit with the mindsets that guide the practices and tools. Gallagher and Thordarson [9] present an alternative approach. They describe five roles for school leaders, which are informed by key design thinking mindsets. These roles and mindsets are supported by practices and tools, and examples thereof. Collectively, these methods suggest a focus on and alignment across tools, practices, and mindsets when translating design thinking to education design contexts. Carlgren and colleagues [10] noted a similar framing when attempting to bridge the gap between design thinking as a scholarly concept and practical enactment.

In engineering course design, contextual and individual factors may threaten the application of design thinking [11]. For example, engineering educators may be challenged to engage in key practices, including empathizing with users (students), framing design problems based on user needs, ideating a variety of design concepts, and engaging in frequent cycles of lo-fi prototyping and user testing [12]. Such challenges may result from a complex interplay of limited design expertise; prior course design habits; and course, department, and university structures [13]. Such challenges have been mitigated through both (1) effective and practice- and mindset-aligned tool use (as suggested by other design thinking toolkits in education) and (2) careful selection, adaptation, and reflective and iterative use of such tools based on the unique contexts of engineering educators [12][13]. In this paper, we describe specific examples in which we've used such contextually-adapted tools to engage key design thinking practices and mindsets.

### *A. Design Thinking Stages*

A variety of process models exist to describe the stages of design thinking. We selected a five-stage model from Stanford's d.school as the basis for our design work for three reasons. First, the model aligns well with scholarly literature on design thinking [10]. Second, the model connected with many of our extant course design practices or offered promising new possibilities. Finally, the model incorporates most aspects of other prominent design thinking models.

Consistent with recommendations in applying design thinking to an education design context [8], we took the basic structure and essence of the five-stage model and refined stage descriptions to better align with an engineering course design context. We made refinements in response to our own initial tensions engaging in a new course design process [13], while considering unique features of the engineering course design context [11]. We present the adapted five-stage model below.

***Empathize*** *with students* – The cornerstone of design thinking in engineering course design is understanding and resonating with the experiences and perspectives of students. This stage involves committing to engage students throughout course design via interaction, observation, and artifact collection, remaining open-minded to the variety of student perspectives and experiences (regardless of differences from our own or our prior conceptions), and actively working to understand student perspectives and experiences authentically and deeply.

***Define*** *(and redefine) a design problem* – Much of the design thinking-related engineering course design process is focused through a tangible, addressable design problem. This problem should be rooted in student needs (e.g., career needs, learning experience needs, emotional needs) but also may be informed or contextualized by a variety of sources, including instructor priorities, accreditation requirements, departmental and university policies, and trends in relevant fields and industries. This stage involves identifying key user needs, translating those needs into a design problem, and reframing that problem throughout the process as new insights emerge.

***Ideate*** – Developing a solution that addresses the defined problem can be supported by ideation that emphasizes identifying a large quantity and variety of design concepts. Such ideation occurs throughout the process during both formal ideation sessions or informal conversations and may be informed by specific formats or remain open-ended. Ideation is most effective when participants keep an open mind, avoid evaluating their own or others' ideas, and build upon prior art and collaborators' ideas.

***Prototype*** *early and often* – Once design ideas form, they are refined, elaborated upon, connected, and manifested as prototypes that can be tested with users or for their suitability in addressing the design problem. Prototypes in engineering course design can take many forms and come in many degrees of resolution (including low-fidelity prototypes). This stage emphasizes frequent creation of prototypes for testing and feedback from users. Prototypes in engineering course design can focus on different course aspects (e.g., a lesson/activity plan vs. a semester timeline) and can be developed for different kinds of learning (e.g., how students might engage in an activity vs. exploring emotional state throughout a course).

***Test*** *prototypes with students (and proxies)* – Closely paired with prototyping, this stage emphasizes testing developed prototypes either directly with students or their proxies to explore how they fit in the design context and meet user needs. Some testing may occur among the engineering course design team, especially with early prototypes, and can inform new understandings of students and the design problem or help to generate new design ideas. The course itself also acts as a "final" test which can lead to new design cycles in following semesters. While testing may use traditional course artifacts (e.g., student work, course evaluations, and reflections), more immersive, interactional, or observational artifacts may lead to better results.

## III. Course Design Thinking Examples

Effective application of design thinking is supported by using design thinking tools adapted for the unique context. We applied design thinking tools in the context of x-teams for two courses — an embedded systems course and a computer organization course. The x-team model was developed and used as part of an NSF Revolutionizing Engineering Departments (RED) grant [12][14] [15][16]. An x-team includes and supports the instructor of a course. Importantly for this work each team had a facilitator with expertise in design thinking and a background in ECE. The facilitator educated and led the team in the process of using design thinking, including fluidly moving through the design thinking stages and attempting to apply an appropriate tool at an appropriate time. Critically, the facilitator allowed flexibility in both (mis)applying the tools and abandoning tools that were not a contextually good fit. Throughout these two x-teams, we have often applied tools in both courses and sometimes repeatedly in the same course. In this section we describe which tools we applied for which design stages in particular situations for each x-team. We include the specific context and connectedness of the process with the intent that it acts as a guide for others to apply design thinking tools to their own courses.

## *A. ECE Courses*

Our x-teams covered two core computer engineering courses commonly taken by computer, cybersecurity, electrical and software engineering majors—an introductory embedded systems course and a computer organization course that follows it. Each x-team operated over five offerings of their respective course.

Our embedded systems course introduces students to hardware and software aspects of embedded systems including microcontrollers, memory, input/output interfaces, embedded programming in C, initialization and configuration of peripherals, polling and interrupt processing, and mobile robots. The course starts with foundational concepts and skills, then concentrates on understanding and using microcontroller peripherals, and finishes with a project in the lab for an autonomous vehicle application. The final project is introduced early and phased in through class and lab activities. This is a sophomore-level course required in our computer, cybersecurity, and electrical engineering programs and elective in software engineering.

Our computer organization course includes processor design and assembly-level programming. This course has a core experiential lab component where students design, test, program, and optimize a processor from basic logic gates all the way to a platform that can execute nearly arbitrary assembly code for a commercial instruction-set-architecture. The course has the reputation as being intense—in time, tedium, and personal growth. This is a junior-level course required in our computer engineering program and elective in other programs.

In both courses, activities were designed or improved through the application of design thinking by x-teams. Some results of the design efforts have been published, including increasing student autonomy and innovation in the embedded systems final project [17]; integrating reflective activities in both courses to enhance learning and professional development [18]; and supporting a debugging mindset in both courses [19]. The work presented in these earlier papers has a focus on the pedagogical approaches and prototypes implemented in the courses. In this paper, we are focused less on what was implemented in a course and more on how – the process of using design thinking and applying design thinking tools. In the following sections, specific examples of applying tools are presented.

## *B. Empathy Stage Examples*

The first design thinking stage listed above is to empathize with students. In this section, we present examples illustrating the use of two tools that support this stage, **personas** and **empathy maps**.

In the first example, we describe using **personas** as a tool in the empathize stage. Early in the instructional design process, the team developed a set of goal-directed personas representing students in the embedded systems course. A case study presenting the development of the personas is described in an earlier paper [17]. The final set of personas are shown in Fig. 1.


Fig. 1: Example personas from embedded systems course.

These personas focused on the goals and motivations of students in the course. Persona development was informed through observations from course instructors and student work including written reflections, feedback, surveys, and assignments. Personas were then used in various course design activities by the team, from informal discussions to ideation, prototyping and testing stages. Personas were used to create a journey map of student experiences in the course. The personas served to diversify team member perspectives and considerations about students.

Team members were involved in all aspects of persona creation, from data collection and user research to artifact development. The personas were created as goal-directed personas, focusing on three statements: what the student wants to do in the course, how the student wants to feel during the course, and who the student wants to be by taking the course. The team began by compiling a variety of user research. This included student artifacts from the course (written reflections, surveys, and assignments), firsthand and secondhand observations from the instructors and teaching assistants, secondhand observations from other team members (e.g., discussions with students about the course through informal interactions and research interviews), and prior analyses of selections of these course data. The team reviewed the data individually and then met to complete an empathy map to summarize their observations about students. Each team member contributed their observations until the team reached saturation. Then the team collectively identified themes across the empathy map, which formed the basis of individual personas. The team described these initial personas, and one member refined and summarized the personas.

In the second example in the empathize stage, we describe using **empathy maps**. The team used the empathy map tool to better understand the student experience of debugging in both the embedded systems course and the computer organization course. Case studies presenting summary empathy maps for the courses are described in an earlier paper [19]. A summary empathy map for the computer organization course is shown in Fig. 2.

An empathy map organizes observations of users based on four aspects of an experience: (1) what they do or say, (2) what they see or hear, (3) what they think, and (4) what/how they feel. These aspects provide a brief, multi-faceted overview of an individual or group's experience. In this example, empathy maps were used to portray the experience of debugging within a course based on several data sources, including student reflections, instructor observations, and other course artifacts. The empathy maps provided information to better understand how students develop debugging mindsets.

Team members were involved in the process of creating the empathy maps, starting with data collection. A subset of team members compiled, reviewed, and analyzed data for the embedded systems course, and another subset, the computer organization course. Data collection focused on identifying observations and data excerpts potentially related (directly or indirectly) to students' debugging experience. Each team member individually placed their data within their own empathy map. For each course, the sub-team members then collaborated to merge the individual empathy maps into a composite empathy map for the course. Key themes were identified and summarized in a final empathy map. These steps are performed iteratively to refine information. The resulting empathy map for the computer organization course is shown in Fig. 2.

| **Do/Say** | **See/Hear** |
|---|---|
| • Overcame (debugging) challenges to complete tasks<br>• Recognized root causes of failing debugging tasks or taking too much time on debugging tasks, especially after large time-lost error<br>• Frequently sought external help (perhaps valuing perspectives)<br>• Frequently put off systematic debugging tasks to end<br>• Spent much of time debugging without robust mental engagement<br>• Debugging forced (independent) learning | • Debugging was contextual – needed to learn tools/methods specific to task<br>• Actual debugging time longer than expected<br>• System-level integration debugging challenging/time-consuming<br>• Teammates significantly affected debugging success/time<br>• Testbenches and debug reports (two parts of debug infrastructure) points of reflection/consideration |
| **Think** | **Feel** |
| • I need stronger background knowledge to support debugging<br>• Debugging tasks can be simplified by design / implementation choices<br>• Debugging and testing is a time-consuming, but inherent part of the process<br>• If the whole system is working, other tests or warnings are irrelevant<br>• Systematic and holistic debugging approaches can help debugging success<br>• Many tasks associated with a systematic testing/debugging approach are wasted work (either for project or learning) | • Excited at the experience of debugging, especially working through strange/unique errors<br>• Frustrated at time and effort of debugging<br>• Debugging is pain<br>• Anxious at errors, especially in a time crunch<br>• In control once errors are understood and corrected<br>• Confused, helpless, or floundering during the process of debugging |

Fig. 2: Summary empathy map for debugging from computer organization course.

*C. Define Stage Examples*

The second design thinking stage listed above is to define (and redefine) the design problem. In this section, we present an example illustrating the use of two tools that support this stage, **big rocks** and **abstraction laddering**.

In this example, we first describe using **big rocks** as a tool. The big rocks tool is based on the "Big Rocks" concept popularized by Stephen Covey, in which the big rocks are put in the jar first so that there is room for other materials to fit in the jar; if other materials are put in the jar first, there isn't room for the big rocks [20]. The big rocks represent major priorities and values. They concretely articulate a small number of key goals to focus on. One benefit of the big rocks approach is that it supports achieving the most important goals as well as some other goals that may be less important but still of interest.

The team for the computer organization course used instructors' prior observations of and interactions with students in past courses to establish a preliminary understanding of the student experience. From this, they proceeded to define and refine what aspects of the course experience should be focused on and redesigned. Initial discussions yielded over 11 main objectives and many smaller ones. The course instructors identified that the intensive term project lab experience was a core feature of the course important for subsequent courses and students post graduation (supported by student comments in later courses, senior portfolios, and e-mails from graduates). Ultimately, the two major goals identified surrounded student perception of interest in the course content and activities:

1. Help students appreciate modern computer architecture.
2. Improve students' experience in lab while retaining relevant challenges.

The next tool used by the team was **abstraction laddering** [21]. Abstraction laddering starts with an initial problem statement and provides a structured approach to framing the problem at different levels of abstraction. Asking "why" leads to more abstract problem statements, and asking "how," to more concrete statements.

The team repeatedly asked "why" in relation to the initial goal and each underlying answer. Fig. 3 shows the why half of the ladder for the initial goal of having students appreciate modern computer architecture. Most statements pertain to students' abilities to communicate with others and to gain knowledge by reading articles, knowing what they don't know, and appreciating system complexity. The computer organization course is the last required course about computer architecture for computer engineering majors. The team used abstraction laddering to refine and gain consensus on the meaning of "appreciate." The underlying motivations given by the why statements led to ideas in the next design stage (e.g., read and review technical article homework question).

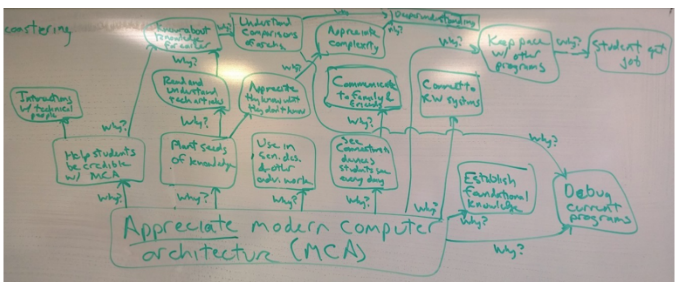


Fig. 3: Abstraction laddering for "Appreciate Modern Computer Architecture."

*D. Ideate Stage Examples*

The third design thinking stage listed above is to ideate. In this section, we present examples illustrating the use of two tools that support this stage, **Lotus Blossom** and **heuristics**.

In the first example, we describe using the **Lotus Blossom** tool in the ideate stage. The team for the embedded systems course used this tool to redesign the lab project. A case study highlighting the use of this tool and prototypes and testing resulting from this stage are described in an earlier paper [17]. Two Lotus Blossoms are shown in Fig. 4.

A Lotus Blossom starts with a core idea at the center of a grid. New ideas that are generated are positioned in surrounding spaces, working outward. Each new idea becomes the center for the next round of idea generation.

In this example, we describe using the Lotus Blossom during redesign of the final lab project in the embedded systems course, which was initiated with the purpose of promoting student professional formation and innovation. For the lab project ideation activity, core ideas for Lotus Blossom sheets were

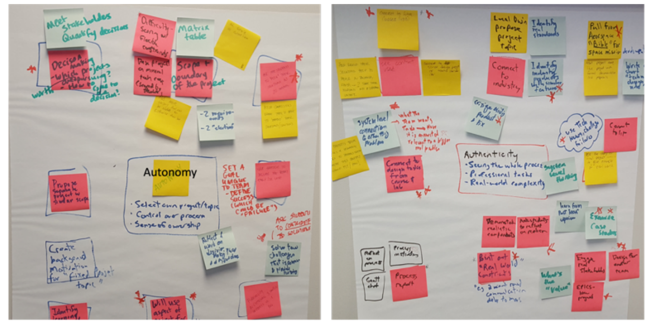


Fig. 4: Two Lotus Blossoms created on poster sheets using sticky notes and markers. Note: Different colors of sticky notes and markers correspond to different team members.

selected from research on engineering student innovation [22] [23]. This research found five constructs that describe how engineering students characterize their innovation project experiences: authenticity, autonomy, support, interest, and novelty. Thus, the tool was set up with sheets for each of the five constructs. In other words, the team started with five large poster sheets, and the center box of a poster sheet (Lotus Blossom) contained one of the characteristics. Autonomy and authenticity Lotus Blossoms are shown in Fig. 4.

Team members then started to write ideas for the lab project on small sticky notes. They placed the sticky notes on poster sheets corresponding to which characteristic would be supported by the idea. Team members continued to generate ideas to fill each poster sheet, adding "blossoms" as needed. These ideas then were then used to create prototypes for the lab project. The goal of the redesign was to support these five characteristics to enhance the student experience, innovation, and professional formation, while retaining some aspects of the prior project.

In the second example for the ideate stage, we describe using **heuristics** as a tool for ideation. An initial set of instructional heuristics was identified in a study of instructional design practices used by the team [13]. The study found a set of 22 instructional heuristics that describe how educators explore and iterate upon the problems and solutions in course design. These heuristics were grouped into categories. In preparation for prototyping for the next semester, the team reviewed categories and heuristics when ideating what aspects of the course should be focused on in the prototype. This ideate stage drew on previous work in other stages including personas. In this ideation activity, the team did not select a specific heuristic and instead used the heuristics structure as a guide. The team considered potential opportunities to extend elements of the course in each category. The team listed elements of and ideas for the course under categories of heuristics.

For example, under the *category of contextualized course content*, heuristics are focused on making course content more meaningful and applicable to students by situating topics within broader professional engineering contexts. Examples of heuristics in this category include: connect to the real world, promote professional formation, expose students to multiple contextual elements, and demonstrate connections between topics. Course elements associated with this category included activities to connect topics in this course with a prerequisite course, with engineering work in a company, and with system sketches. There was also an emphasis on connecting lecture topics with weekly lab work.

As another example, under the *category of use prior art*, heuristics are focused on applying previously developed solutions into the current course design environment. Examples of heuristics in this category include: translate past experiences and introduce evidence-based practices. The heuristic to translate past experiences considers an approach taken in another course. Motivated with these heuristics, the team identified student reflection as a potential element of the course to expand on based on reflective activities used in other ECE courses taught by team members and research about learning cycles and reflection. The team found that considering heuristics across categories and seeing specific goals to be accomplished helped the team to converge on elements and activities of the course to focus on in the prototype. Prototypes resulting from this stage are described in an earlier paper [18].

### *E. Prototype Stage Examples*

The fourth design thinking stage listed above is to prototype (early and often). In this section, we present examples of approaches to rapid prototyping: **Slide decks**, **sketches**, and **timelines**. Initially, we struggled to conceptualize and define what concrete design artifacts or prototypes exist for course design. Specifically, we recognized that the course experience may be the closest design object to a product or prototype [11]. In the educational domain, course materials such as lab manuals, template programs, presentations, activity handouts, or quizzes are often a natural prototype. However, design thinking encourages quick, rapid, low-fidelity (lo-fi) prototypes that can be tested by the design team. Therefore, we created our own tools and methods to represent course design prototypes.

In the first example, we describe using a **slide deck** as a tool in the prototype stage. The team for the computer organization course used slides to prototype a new learning cycle for labs based on reflective activities. Results from this prototyping are presented in an earlier paper [18]. A sample slide is shown in Fig. 5.

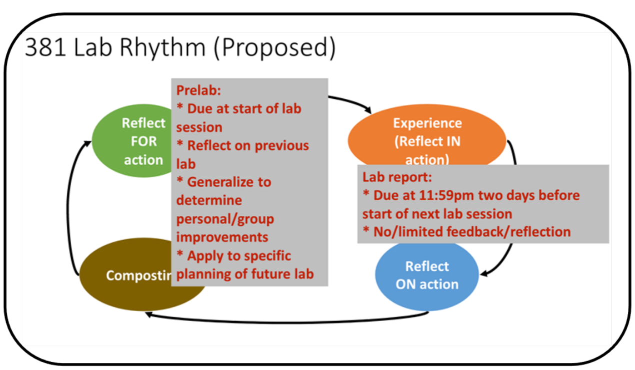


Fig. 5: Sample prototype slide of lab rhythm.

The prototype for the lab learning cycle specifically accounted for the timeline and rhythm of the lab experience. Fig. 5 shows a prototype slide that illustrates the new rhythm for labs where every two weeks students complete a modified Kolbian cycle that includes reflection for action (i.e., completing a pre-lab using composted reflection from previous lab/course activities), reflection in action (completing the lab activities), and reflection on action (producing the team lab report as well as individual feedback and evaluations about the process and experience of the lab) [18].

An important aspect of the prototype slide decks was the inclusion of connections to other stages of the design thinking process. The tangible connection helped the team track why a prototype was developed, which resulted in more intentional prototypes, such that intent could be more clearly communicated to students. For example, one prototype for the computer organization course included a concept map photo from the define stage annotated with prioritized goals for three objectives targeted by the prototype – team formation, structured design and testing approach, and accountability.

For objectives targeted by the prototype, the instructor included one or more slides with details about the support for the objectives. For team formation, the prototype included mechanisms of forming teams for initial labs, allowing some autonomy (e.g., initial partner selection), but also some more prescribed approaches (e.g., final teams of partners) and what considerations would be used (e.g., schedule availability, course goals, and background skillsets). For structured pairing roles in lab [24], the prototype included a **sketch** of what students would be doing in their assigned roles as shown in Fig. 6. For accountability, the prototype delineated essential elements needed for a lab-specific team contract template, mechanisms for individual assessment, and team dissolution.

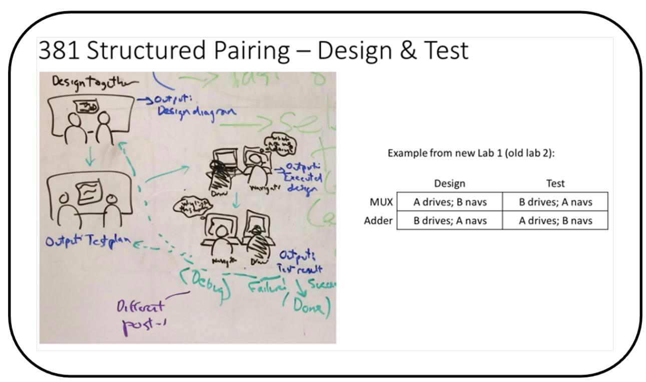


Fig. 6: Sample prototype slide of structured pairing.

In the final example for the prototype stage, we describe using a **timeline** as a tool. The team for the embedded systems course used timelines as a prototyping tool when redesigning the lab project experience. The ideation stage corresponding to this lab project redesign is described earlier in section D, which provides context for this prototyping example.

The team redesigned the lab project and its processes to incorporate characteristics associated with environments that foster innovation [22][23]. Many of the ideas generated through ideation were incorporated in a timeline-based prototype, which allowed the team to weave the project milestones throughout the overall course structure. Fig. 7 shows a photo of the initial whiteboard sketch of selected milestones mapped to a timeline. The timeline sketch was then refined using drawing tools on a slide for use in the course. The instructor uses the timeline to introduce students to the project during the first week of the semester and walk them through the project in relationship to weekly labs during the semester – helping students see the course structure and flow.

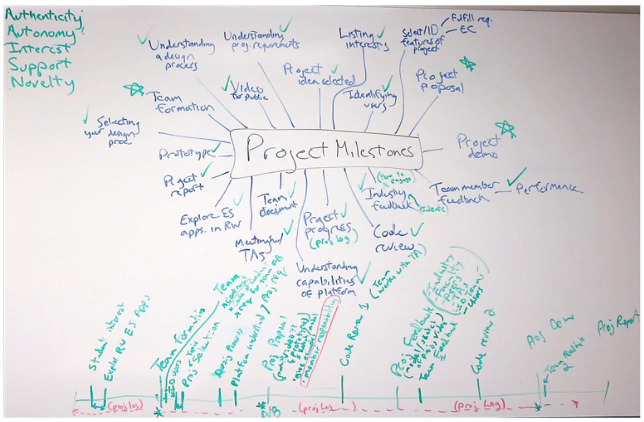


Fig. 7: Prototype timeline for lab project milestones; timeline is drawn at the bottom.

*F. Test Stage Examples*

The fifth design thinking stage listed above is to test prototypes with students. In this section, we present examples illustrating the use of a **journey map** as a tool supporting this stage. Journey maps are used to understand a user's experience with products, systems, and services over time [25]. The use of personas and journey maps in course design in the embedded systems and computer organization courses was studied in a previous paper [26].

In the first example, the team for the computer organization course created a journey map to visualize the experience that students went through during the semester. After making changes to the lab manuals, teaming process, and template designs, the team mapped responses from the same assignments into previously developed course personas. The journey map was augmented with actual student comments. This process was done during the summer of 2020 when collaboration was virtual due to the pandemic, thus the journey map was generated collaboratively in online meetings using Google Draw, as depicted in Fig. 8. Student quotes were copied into textboxes. The journey map let the team concretely visualize how and when the changes did or did not help overall. One interesting aspect was that many student personas went through an initial or middle period where the lab experience had a negative impact on resonance with the course. However, for several personas, the end of lab – when things came together – was a positive experience. Students also noted that pain or tedium in the early part of the lab, for example, learning tools that seem inefficient,

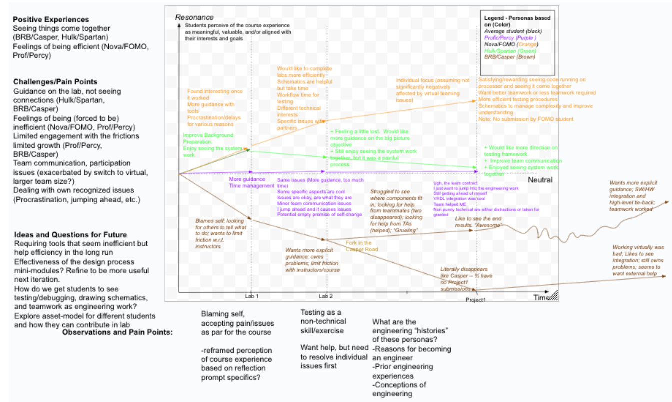


Fig. 8: Example journey map used to test the student experience in the computer organization course. Y axis is "resonance" with the course and x axis is time in lab assignments.

ultimately saved them time in the end. This testing led to further understanding of students and ideas for refining the lab experience.

In the second example, the team for the embedded systems course created a journey map to visualize student engagement in the lab from week to week throughout the semester. The laboratory experience is a major aspect of the embedded systems course. The purpose of the labs is to give students hands-on experience with concepts they learn during lecture. The team decided to use a journey map to more systematically evaluate how students experience and engage with the current labs. In addition, the journey map creation process generated information that was useful in helping the team better empathize with students and identify lab "pain points," which helped more concisely define what aspects of labs to target for change.

The team used a journey map to specifically test the engagement of six student personas with labs as they progressed through the semester. These personas were developed as part of a separate design activity by the team, as presented earlier in section B. To create the journey map, team members embodied the personas, each taking a persona and portraying that persona's engagement with each lab over time. Each team member marked on a scale their persona's engagement level for each lab, in chronological order, and labeled their response with a word or phrase that captured how that persona felt during the lab. While creating the journey map, team members discussed the reasoning behind their persona's responses. The initial journey map was drawn on the whiteboard as a collabortive team activity. Fig. 9 is a cartoon version showing three of the six personas in a more readable format.

Seeing and discussing each persona's lab journey helped the team identify when "pain points" or difficulties occurred during their journey, and gave a sense for why. Often difficulties were found to be associated with situations during the semester in which students struggled in some manner. For example, feeling overwhelmed, feeling lost in the details, not seeing how a given lab supports the larger goals of the course, and/or not seeing how

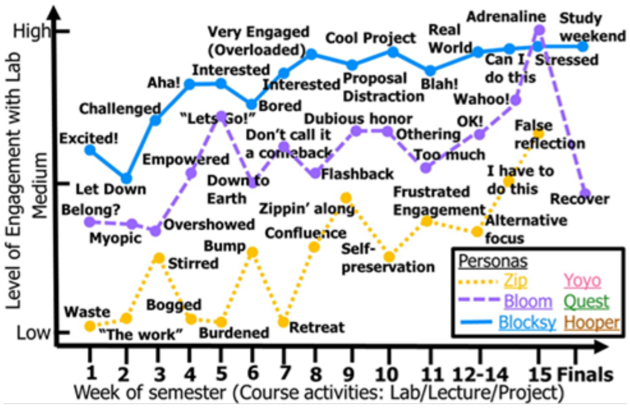


Fig. 9: Example journey map used to test student engagement in the lab in the embedded systems course. Y axis is engagement in the lab and x axis in time in weeks. Note: This is a "cartoon" version of journey map that was originally created on a whiteboard.

labs connect with individual goals. These "pain points" formed a basis from which the team focused their efforts with respect to improving the student lab experience. For example, focusing on students feeling lost in the details and not seeing how a given lab supports the larger goals of the course, the team developed ways for students to experiment with the larger system earlier in the semester (e.g., with pre-compiled functions that hide some low-level details until later in the course). Working with the larger system gives students a greater appreciation for the details they will later implement and their relationship to the whole system.

In addition to using journey maps for testing prototypes and informing design decisions, the teams also tested prototypes through their own inspection and through use in the course itself.

For example, for the computer organization course, the prototype slide decks would be presented at team meetings where members (other instructors, TAs, etc.) would provide feedback on how they believed students would experience and react to the changes given their experiences and student proxies such as personas. Essentially this feedback from team members was an early test stage. Iterations occurred until prototypes were integrated into the next offering of a course as actual course materials and activities.

Each course offering then represented an actual test environment with students taking the course. During and after the offering, the team would inspect student work, feedback, and reflections to understand how the prototype impacted the student experience in the course. For example, for the computer organization course, team members browsed through and discussed the periodic student reflections from the lab rhythm described above and examined how students experienced the lab activities.

## IV. Concluding Discussion

In conclusion, we applied design thinking practices (i.e., processes and associated tools) to the (re)design of two core computer engineering courses using collaborative, cross-functional instructional teams. A key observation was the need to flexibly use design thinking tools to achieve the desired end in the context of the course. While this process was significantly aided by our design facilitator, we hope that the examples provided here may inspire others to engage with design thinking even in the absence of a facilitator. Another key observation was the centrality of developing empathy with our students and leveraging this empathy to understand how students would experience a change in the course. Finally, we observe that many of the positive impacts from the design process occurred after multiple iterations of a prototype and can only be accomplished with the mechanisms of a persistent (re)design cycle.

## References


[1] T. Brown, "Design thinking," *Harvard Business Review*, vol. 86, no. 6, p. 84, 2008.

[2] T. Brown, *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation.* New York, NY: Harper Collins, 2009.

[3] N. Cross, *Design Thinking: Understanding How Designers Think and Work*. Oxford, UK: Berg, 2011.

[4] B. Lawson, *How Designers Think: The Design Process Demystified*, 4th ed. Oxford, UK: Architectural Press, 2006.

[5] J. Liedtka, D. Azer, and R. Salzman, *Design Thinking for the Greater Good: Innovation in the Social Sector*. New York, NY: Columbia University Press, 2017.

[6] U. Johansson-Sk¨oldberg, J. Woodilla, and M. C¸ etinkaya, "Design thinking: Past, present and possible futures," *Creativity and Innovation Management*, vol. 22, no. 2, pp. 121–146, 2013.

[7] IDEO, "IDEO Toolkit – Design Thinking for Educators," *IDEO*. [Online]. Available: https://www.ideo.com/post/designthinking-for-educators. [Accessed: May 20, 2023].

[8] K. Tschimmel, D. Loyens, J. Soares, T. Oraviita, A. Jacinto, A. Barroca, B. Marinova, J. Santos, M. Carenzo, and S. Manzini, *D-Think Toolkit. Design Thinking Applied to Education and Training*. Matosinhos: Erasmus+/ESAD, 2017.

[9] A. Gallagher and K. Thordarson, K., *Design Thinking for School Leaders: Five Roles and Mindsets that Ignite Positive Change*. Alexandria, VA: ASCD, 2018.

[10] L. Carlgren, I. Rauth, and M. Elmquist, "Framing design thinking: The concept in idea and enactment," *Creativity and Innovation Management*, vol. 25, no. 1, pp. 38–57, 2016.

[11] N. D. Fila and C. Schimpf, "An engineering course as a design object," in *2022 ASEE Annual Conference & Exposition*. Minneapolis, MN:ASEE Conferences, August 2022, https://peer.asee.org/41745.

[12] N. D. Fila, S. McKilligan, and K. Guerin, "Design thinking in engineering course design," in *2018 ASEE Annual Conference & Exposition*, no. 10.18260/1-2–30271. Salt Lake City, Utah: ASEE Conferences, June 2018, https://peer.asee.org/30271.

[13] N. D. Fila, D. T. Rover, H. Duwe, M. Mina, and P. Jones, "Course design thinking: Navigating tensions at the intersection of design thinking and engineering course design," in *2023 ASEE Annual Conference & Exposition*. Baltimore, Maryland: ASEE Conferences, June 2023, https://strategy.asee.org/42773.

[14] N. D. Fila, S. McKIlligan, and S. J. Abramsky, "How engineering educators use heuristics when redesigning an undergraduate embedded systems course," in *2018 ASEE Annual Conference & Exposition*, no.10.18260/1-2–30579. Salt Lake City, Utah: ASEE Conferences, June 2018, https://peer.asee.org/30579.

[15] N. D. Fila, D. T. Rover, M. Mina, and P. H. Jones, "Cross-functional team course design project in engineering," in *2020 ASEE Virtual Annual Conference Content Access*, no. 10.18260/1-2–34358. Virtual Online: ASEE Conferences, June 2020, https://peer.asee.org/34358.

[16] S. McKilligan, N. D. Fila, D. Rover, and M. Mina, "Design thinking as a catalyst for changing teaching and learning practices in engineering," in *2017 IEEE Frontiers in Education Conference (FIE)*, 2017, pp. 1–5.

[17] D. T. Rover, N. D. Fila, P. H. Jones, and M. Mina, "Introducing autonomy in an embedded systems course project," in *2020 IEEE Frontiers in* Education Conference (FIE), 2020, pp. 1–9.

[18] D. T. Rover, H. J. Duwe, M. Mina, N. D. Fila, P. H. Jones, and L. S. Sleeth, "Learning and professional development through integrated reflective activities in electrical and computer engineering courses," in *2021 IEEE Frontiers in Education Conference (FIE)*, 2021, pp. 1–9.

[19] H. Duwe, D. T. Rover, P. H. Jones, N. D. Fila, and M. Mina, "Defining and supporting a debugging mindset in computer engineering courses," in *2022* IEEE Frontiers in Education Conference (FIE), 2022, pp. 1–9.

[20] S. R. Covey, *The 8th habit: From Effectiveness to Greatness*. New York, NY: Simon and Schuster, 2013.

[21] Interaction Design Foundation, "Why-How Laddering," *Interaction Design Foundation*. [Online]. Available: https://www.interaction-design.org/literature/topics/why-how-laddering. [Accessed: May 20, 2023].

[22] N. D. Fila and S. Purzer, "Exploring connections between engineering projects, student characteristics, and the ways engineering students experience innovation," in *2017 ASEE Annual Conference & Exposition*, no. 10.18260/1-2–28326. Columbus, Ohio: ASEE Conferences, June 2017, https://peer.asee.org/28326.

[23] N. D. Fila, J. L. Hess, and S. Purzer, "Instilling new ways of understanding the innovation process: Evidence-based heuristics from student innovation experiences," *The International Journal of Engineering Education*, vol. 36, no. 2, pp. 633–649, 2020.

[24] N. D. Fila and M. C. Loui, "Structured pairing in a first-year electrical and computer engineering laboratory: The effects on student retention, attitudes, and teamwork," *International Journal of Engineering Education*, 2014.

[25] T. Howard, "Journey mapping: A brief overview," *Communication Design Quarterly Review*, vol. 2, no. 3, pp. 10–13, 2014.

[26] N. D. Fila, D. Rover, H. Duwe, and M. Mina, "Considerations for the use of personas and journey maps in engineering course design," in *2022 ASEE Annual Conference & Exposition*. Minneapolis, MN: ASEE Conferences, August 2022, https://peer.asee.org/41746.