





STRATEGIC PROTOTYPING TO LEARN IN STANFORD UNIVERSITY'S ME310 DESIGN INNOVATION COURSE

L. Domingo , D. Moore, D. Sirkin, G. Toye, L. Leifer and M. Cutkosky

Stanford University, United States of America

 ldomingo@stanford.edu

Abstract

Through a strategic learning process, prototypes unveil design directions. We provide a review of prototyping methods for novice designers to study and pedagogical practice for capstone design course faculty to juxtapose. Stanford University's ME310 graduate-level project-based learning course introduces students to various prototyping design techniques, such as Needfinding and Benchmarking, and prototyping methods, such as the Critical Experience Prototype, Critical Function Prototype, Dark Horse Prototype, Part-X is Finished, Funky System Prototype, and Functional System Prototype.

Keywords: design education, design learning, prototyping, engineering design, design pedagogy

1. Introduction

To experienced engineering designers, the design process can feel natural and intuitive. However, design depends on tacit knowledge, and, as a result, defies systematization, and even a clear definition. There are many different approaches to design including design thinking, systematic design, Syntectics, and TRIZ, and just as many ways to teach design. For students from technical backgrounds that emphasize analytical thinking, teaching a design process presents many challenges, such as how to approach undefined, open-ended problems (Dym et al., 2005; Buchanan, 1992).

Despite the popular moniker of design *thinking*, the design process itself advocates design *doing* and is most successful when accompanied by a mindset open to exploring and expressing new ideas in physical forms. Design thinking and other approaches provide overarching frameworks such as Express-Test-Cycle (Tang, 1989), but rarely prescribe specific actions for teams to take. Educational curricula and company design cultures can then offer specific steps to explore user needs and prototype solutions.

This paper summarizes the pedagogical prototyping method that Stanford University's graduate-level design methodology course, ME310, uses to teach design methods and navigate the design process. In doing so, we juxtapose Stanford's methodology (which has evolved over more than 50 years) with existing models such as Houde and Hill (1997) and Nielsen (1989). While we highlight one model, we acknowledge that many effective methodologies exist across institutions, and countries. ME310 primarily has a mechanical engineering focus, and we acknowledge that different disciplines will have different prototyping emphases. For example, aerospace engineering emphasizes simulation techniques, and software development emphasizes numerous parallel and iterative prototypes (Dow et al. 2010).

Across disciplines, prototypes are a central tenet of learning to unveil design directions, and not just to demonstrate decisions already made. Prototypes fulfill various purposes in the engineering design and

development process (Houde and Hill, 1997; Nielsen, 1989; Lande and Leifer, 2009; Brereton and McGarry, 2000; Bohmer and Kayser, 2017) such as to:

- Test novel ideas (discovery)
- Answer questions that arise throughout the exploration process
- Create a shared medium for communication for intended users and internal design teams
- Elicit user needs
- Strategize technical development through strategic subsystem testing (decision-making)

Often in pedagogical practice, students learn the different prototyping techniques in succession, as presented throughout this paper. However, in design practice, these techniques need not be used in a strictly linear fashion, and instead should be used whenever the design team finds appropriate. In reality, the insights from the preceding prototypes and the timeline of the project should guide which prototyping methodology should be utilized next. For example, there may be a time to revisit previously abandoned ideas, which can only occur after a team has pursued one or more ideas first. We review the prototype themes from the ME310 course pedagogy as currently taught with the goal of (1) stimulating discussion among the design education community on how the nature of prototypes changes over time, (2) better situating the ME310 approach to prototyping with respect to other design cultures, and (3) providing a snapshot of the current methodology for future designers to reflect on as ME310 evolves in the coming decades.

2. Related work

To characterize the broader context in which the ME310 prototyping process fits, we first highlight various features in our concept of prototyping: the purpose a prototype has in a design process, the general quality of a prototype, and the process of implementing prototypes.

2.1. Prototyping: Purposes

Successful prototyping and testing clarify design decisions that guide the next stages of a design process and inform a development strategy by navigating ambiguity and assessing risk (Camburn et al., 2015). Prototypes embody theories exposed to reality to be proved or disproved through technical development. Several models offer ways to prototype and evaluate different aspects of a design. Houde and Hill (1997) outline four prototyping dimensions, and highlight the purpose of each: implementation (evaluate technical functionality), look and feel (explore user experience), role (assess value added to the user), and integration (role, look and feel, and implementation in tandem).

When designing complex systems with many potential functions, Nielsen (1989) offers three kinds of prototypes: horizontal prototypes, vertical prototypes, and scenario prototypes. Horizontal prototypes test many possible design features at low resolution and fidelity to triage which ideas are best to pursue. They are typically used early in a design project to facilitate divergence. Vertical prototypes emphasize deep functionality of a few ideas to further advance idea resolution. They are typically used at a later stage of design project maturity and facilitates convergence. Scenario prototypes have a few functions with minimal functionality and balances both horizontal and vertical prototyping methods.

In the education context, Bohmer and Kayser (2017) studied prototyping in a design for manufacturing course where undergraduate students manufacture a self-scoped project. The authors found that different prototyping techniques served as tools for discovery or decision making and that certain prototyping methods were more beneficial for some student's projects than others.

Lande and Leifer (2009) distinguish between design thinking (DT) and engineering thinking (ET) prototypes. DT prototypes, such as sketching, conceptual prototyping, and Wizard-of-Oz experience prototypes emphasize role and look and feel. DT prototypes generally engender divergence to facilitate idea generation and problem definition by bringing a concept into reality that can be shared. ET prototypes, such as CAD models and physical prototypes, embody technological concepts, and facilitate convergence that brings design specifications to a more concrete resolution.

2.2. Prototyping: Qualities

Each prototype during development can be characterized by its resolution and fidelity. A prototype's resolution is described as the amount of detail. A prototype's fidelity is described as the closeness to the final product (Houde and Hill, 1997). When answering questions with user testing, the prototype must be a good enough approximation with a believable degree of resolution and fidelity for users to be able to engage appropriately (Houde and Hill, 1997). Presenting a rough (or inadequate) prototype to an ill-informed audience can elicit inaccurate, negative feedback, where novice design audience members may have difficulties seeing past the prototype's roughness and contextualizing the prototype in the intended environment and time point in the design process. As a result, all user testing requires strategic introduction (Schrage, 1996).

Whether a cardboard mockup, a sketch, or a sophisticated CAD model, the designer's chosen prototyping approach must balance several factors, and ideally answer a specific question. Lower resolution prototypes can quickly explore a specific design space and better elicit user needs early in the design process. Higher fidelity prototypes are better suited for feature refinement and insight into design specification (Edelman, 2011). However, low resolution prototypes can negatively affect user perception during testing and impede the usage flow. Moreover, complex integrated prototypes can distract users from the core question to be answered and leave more room for troubleshooting slowing down the development process. Furthermore, subsystems of an integrated prototype are not isolated making it difficult to tease out what feature may influence a certain outcome (Brereton, 2000). Integrated prototypes typically require higher resolution to assist with interconnecting into a larger whole.

2.3. Prototyping: Styles

Different stages of a design process call for different approaches depending on the question to answer or problem to solve. With an intention for each prototype developed in mind, designers can use different prototyping styles to explore the design space (Brereton, 2000; Edelman, 2011).

Iterative prototyping (Schön, 1938; Camburn et al., 2015) seeks to apply lessons learned through a "cycle of building testing and improving a single design concept" (Camburn et al., 2015, p. 3). Iterative design can be optimal when changes are small and the final desired outcome (i.e. requirement or specification) is well defined requiring primarily refinement. With each newly developed prototype or representation, a designer develops their repertoire of expectations, images, and techniques (Schön, 1938). Iterating and sharing multiple prototypes within design teams and amongst users facilitates building ideas to be built on one another towards a higher fidelity (Bushnell, 2013; Houde and Hill, 1997; Yang, 2005).

Rapid prototyping (Adams, 2005; Carleton and Cockayne, 2009) involves making or expressing low resolution prototypes to quickly explore a design space, generate a shared communication medium, and explore potential user requirements. This approach can be used to concretely simulate user interactions or abstractly express end goals for decade long development cycles. Rapid prototypes developed early on can serve as objects imbued with social or cultural knowledge that can evoke ideas explored in the past and the subsequent lessons learned.

Parallel prototyping (Nielsen, 1989; Camburn et al., 2015) involves the development of multiple prototypes at the same time to explore the feasibility of many ideas. Parallel prototyping methods facilitate risk mitigation to guide a process of developing low cost, resolution, and fidelity prototypes to quickly abandon design directions that demonstrate little promise and to save on resources (i.e. time and money) to further develop more promising ideas. Furthermore, the learning from a breadth of prototypes explored, whether abandoned or pursued, can complement the development of converged upon prototype ideas. Parallel prototypes can also facilitate the exploration of design assumptions triggering the exploration of design spaces previously perceived infeasible (Bushnell et al., 2013; Edelman, 2011; Camburn et al., 2015).

3. Course context

ME310 is a year-long project-based learning (PBL) graduate-level course with projects sponsored by industry partners (Dym et al., 2005). Project prompts can vary from developing or applying a novel

technology to designing for open-ended wicked problems (Buchanan, 1992; Rittel and Weber, 1973) across a wide range of domains including cosmetics, wearable interactive devices, and satellite construction processes. After an introductory design cycle to gain familiarity with the course's process, where students rapidly build a vehicle out of cardboard, students explore their industry sponsor's open-ended design space, develop numerous prototypes, and create a final concept that they present to industry sponsors and the broader design community through presentations and a trade show. Project teams comprise 3-4 Stanford students partnered with comparably sized student teams from a global university partner that also follow an ME310-like methodology (Wiesche et al., 2018). Project teams share a prototyping budget of approximately \$16,000 (USD). Through learning design methodology, students develop new engineering skills, strengthen their creative confidence, and participate in an extended community of designers to develop a tangible and polished final prototype. The student learning cycle of ME310 applies Kolb's experiential learning cycle of concrete experience (needfinding and prototype testing), reflective observation (meetings with the teaching team), abstract conceptualization (applying engineering principles), and active experimentation (prototyping) with each new prototype developed and tested (Kolb, 1984). As a course final deliverable, students complete a final report to document the project's problem exploration through to its eventual final design development process for future teams and industry partners to share in the lessons learned.

One element of the course that makes ME310 unique within engineering is that the students develop and iterate on project requirements and specifications, rather than just building a system with pre-set requirements. This openness is often challenging for engineering students who need to learn to "dance with ambiguity" (Leifer and Steinert, 2011). Part of ME310's pedagogical intent is to guide students to navigate the degree of resolution and fidelity a prototype should have to elicit feedback for project progress. Moreover, the functional requirements for every system must be informed by user research and testing with real (potential) users outside of the classroom.

For ME310, the relevant audience members are the: end user, student design team, and project sponsor organization, with one additional audience member being the teaching team. The teaching team provides students with an allocentric perspective and feedback through design critique sessions, where they provide coaching and technical support that guides teams throughout the school year. The teaching faculty has found that the studio feedback model, like in architecture and art, where completed work qualitatively critiqued on a case-by-case basis, helps to nurture a growth mindset (Dweck, 2008) and develop confidence in wayfinding ambiguity (Edelman, 2011). Edelman described Wayfinding as the (Edelman, 2011, p. 103):

...in situ determination of a [design] route based on perceptual cues often in the context of narrative...Rather than a top down approach of determining what is wrong with the product and following a program for fixing it, [designers] 'feel' the object and the environment as the scenario unfolds.

The prototyping methods scaffold students' wayfinding through the divergence of design exploration and the convergence of design execution.

4. Design methods

As summarized in Figure 1, we will briefly describe each prototype taught to, and used by, students throughout the academic year in ME310. For each prototype, we also share how a student team implemented that particular prototype for their project.

4.1. Benchmarking and Needfinding

Benchmarking explores current technology and trends for students to avoid reinventing the wheel. As a part of the Benchmarking process, students research market trends, products, and key industry players in order to understand competing technologies, and to understand the evolution and shortcomings of existing technologies. Benchmarking emphasizes understanding where technologies may have succeeded due to new features and markets or failed due to previous limitations in manufacturing techniques. Students benchmark both obviously relevant and analogous technologies and trends. Part of the project budget is allocated for students to purchase current market products and reverse engineer mechanisms of how current market products work in order to more deeply understand the functionality

of a current technology. Students use historical trends to understand future trends and apply foresight thinking methods (Carleton and Cockayne, 2009) to scope technology trends relevant for their project's development. Though taught early in the course, benchmarking continues throughout the school year.

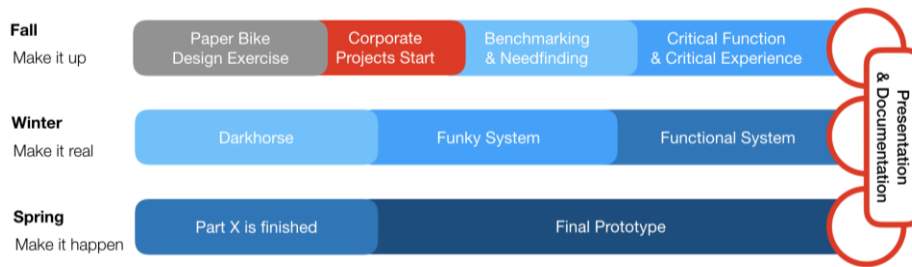


Figure 1. A caricature of the ME310 design project timeline

Needfinding is the process of observation and identifying needs of a user group. According to Rolf Faste, “A need is a perceived lack, something that is missing... In order to find and articulate a need, this missing thing must be seen and recognized by someone” (Faste, 1987, p. 1). The Needfinding process borrows from the art community emphasizing a change of perception to see the world in ordinary, new ways. “Needfinding is the act of discovering people’s explicit and implicit needs so that designers can create appropriate solutions” (Patnaik and Becker, 1999). Physical prototypes can also serve as a way to elicit user needs not previously identified (Houde and Hill, 1997). Through Needfinding, student designers can uncover users’ deeper, root problems instead of just addressing symptomatic issues. Benchmarking and Needfinding are complementary activities that promote efficient and effective use of time and resources. In any design project, the most limited resource is time, and reinventing old ideas wastes resources. Proper benchmarking contributes to a design team’s efficient use of time. Needfinding emphasizes that the proper problem should be addressed given the scale and scope of a design challenge and thus emphasizes the effective use of time and resources to develop products and prototypes that add value for users. While the pedagogy teaches Benchmarking and Needfinding techniques early in the academic year, students should (and often do) repeatedly perform Benchmarking and Needfinding activities throughout the entire duration of the project to explore different design directions.

4.2. Critical Function Prototype (CFP)

Designing new, complex, engineered systems often requires combinations of subsystems. The *Critical Function Prototype* encourages designers to build and test the most ambiguous or challenging element of a complex system first that the rest of the system is contingent upon, rather than building easier subsystems first. Conceptually, the CFP is the minimum essential functional component to build credibility of a solution idea, and the insights from the CFP elucidate whether the imagined design direction is worth pursuing. The teaching team has found that students often struggle with developing a sufficient CFP as a first assignment for many reasons including time limitations and experience difficulties identifying the most critical element of a system before it is built. However, the learning goal of identifying and testing a critical technical function and knowing about the possibility of a CFP nurtures student’s confidence in developing future prototypes and a more concrete design vision. The CFP also emphasizes the quantification of a system’s capabilities in order to determine a system’s specification, calibration, and technical feasibility.

CFP Case Study

Context: Team Microsoft wanted to simulate the different somatic feedback sensation to simulate the experience of touching the warmth of another person’s hand.

Prototype: The end goal experience required mechatronic development where the critical function was to create a specific heat and motor sensation like touching another’s hand. This system was the most novel part of a communication system design and thus the riskiest technology that required prototyping.

Outcome: Team Microsoft applied functional prototyping techniques to create an experience and had to decompose a complex output into prototyping components such as a vibration motor and Peltier device. Team Microsoft developed self-efficacy using the different hardware components gaining more mechatronics experience in the process.

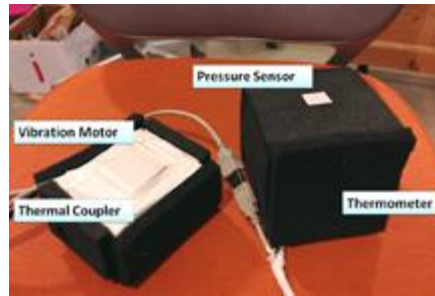


Figure 2. Team Microsoft's hand comforting CFP

4.3. Critical Experience Prototype (CEP)

The *Critical Experience Prototype* can elucidate whether a suggested prototype or concept adds value to a user group and if a design concept carries a potential that leads to a desirable goal. Often times Wizard-of-Oz prototyping techniques are used to simulate the experience of using a new technology, implementing “smoke-and-mirrors” to create the context and story in which a user would use the technology to be developed (Sirkin and Ju, 2014). The CEP looks towards designing the minimum essential component for human perception or somatic experience of a solution idea. As such, the CEP allows designers to test the user experience of future technology using present day technology. The teaching team has found that the CEP helps student communicate and scope forecasted user experiences. CEP and CFP prototyping techniques can be concurrently used in the same integrated prototype in order to tease out both technical feasibility and user experience.



Figure 3. Team Nestle's recipe suggesting aisle CEP

CEP Case Study

Context: Team Nestle wanted to test the customer experience of interacting with an automated grocery shelf that would recommend ingredients based on a selected recipe.

Prototype: This prototype hoped to promote amateur chefs to cook more often. The implementation of the prototype was easily executed using a transistor-controlled LED circuit.

Outcome: The team learned that the experience was well equipped for individual users but not for groups since the sensors and perspective lighting was a difficult design to implement for different angles and moving bodies. Team Nestle decided to scope their project to aisle ends so that the final prototype was able to interact with one user at a time. The team realized that viewing the ingredient boxes from different perspectives was technologically difficult and scoped their project to single user interactions.

4.4. Dark Horse Prototype (DHP)

Risk aversion due to a fear of ambiguity can often stifle the creative process resulting in the exploration of old, safe ideas and an ineffective use of time. After a coherent vision has been developed, the *Dark Horse Prototype* encourages the divergent exploration of risky ideas in a finite amount of time that had otherwise been ignored due to common sense. The DHP forces designers to suspend an underlying cognitive solution fixation and to consider alternatives. The DHP applies the CFP and CEP while embracing a more divergent disposition. The learning outcome of the DHP is to question the assumptions that a design vision may hold as immutable and expand the design space. While DHPs are often highly experimental, elements of them often linger in a team's final concept (Bushnell et al., 2013).

A DHP is characterized as a prototype that explores a risky, radical, infeasible, or poorly studied solution where there is both a high risk in future feasibility and high return on learning and reward in enhanced capability, functionality, efficiency, cost, simplification, etc. Proper benchmarking is critical leading up to the DHP since students should then be able to identify gaps in technological development and understanding. Furthermore, planting a prototype in the new, unknown concept space can bear high rewards in learning and project progress. Successful DHP are technically well executed or user tested to push the design space boundary.



Figure 4. Team Volvo's construction dirt sequestering DHP

DHP Case Study

Context: Team Volvo originally explored the design space of urban mining and glass disposal and reuse. Pivoting away because of the exploratory nature of the Dark Horse Prototype, Team Volvo wanted to explore redesigning an unquestioned activity of construction site dirt storage.

Prototype: The team learned that storage and transportation of dirt at construction sites took up a lot of space and power. Team Volvo developed a construction site storage vessel that was stackable, reusable, and took up less space on construction sites than free forming piles.

Outcome: Team Volvo learned through prototyping that they needed to reinforce the interconnections between the hardwood since the binding mechanism was the failure point during failure testing.

4.5. Funky System Prototype (FKSP)

The *Funky System Prototype* begins the process of integrating all system components to evaluate a project holistically and examine the system's boundaries. The foundational prototyping techniques (i.e. CFP, CEP, DHP) used up to this point are typically focused in scope to develop or explore a function or experience. The FKSP is Funky since the subcomponents that make up the system are not required to be built to a high fidelity. Parts of the system have been "hacked" together with low resource investments in order to define the system boundary of the final prototype and elucidate system interconnections. In practice, by laying out the entire funky system, designers can determine what subsystems or subcomponents were completely forgotten altogether, require further development, or simplification. The objective is to demonstrate a desired function and bring broader system boundaries into focus. By laying out a medium fidelity system prototype, design engineers can strategize what subsystems to prioritize and what materials and resources will be needed in the future.

FKSP Case Study

Context: Team Xylem began their convergence by developing a FKSP to layout what systems they planned to develop and tease out what other subsystems to develop for the final prototype.

Prototype: The team integrated two solenoid valves, a pH sensor, and temperature sensor to test out the efficacy of an integrated control looped mechatronic system to respond to changes in water quality. In parallel, the team began developing the piping geometries and configurations that would later be integrated with the sensor system. Components were integrated in such a way that enabled disassembly for further subsystem refinement.

Outcome: The team learned that they also needed to source metal plumbing components and interconnections. The team also decided to build an irrigation system that would require purchasing a water reservoir that took two weeks to manufacture and deliver.

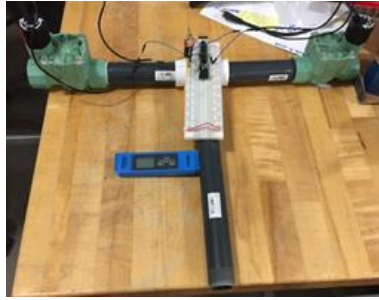


Figure 5. Team Xylem's water distinguishing FKSP

4.6. Functional System Prototype (FCSP)

The *Functional System Prototype* is a medium-high fidelity prototype that marks a major shift away from exploration to exploitation. The FCSP provides a clearer scope in terms of the final physicality, materials used, packaging, etc. Wizard-of-Oz prototyping techniques should be used to a minimum if at all and the FCSP should be complete technically. Through focusing on technical implementation, students can begin discussions to gauge whether they need to hire vendors or outsource work to create a polished final prototype's standards on time. The FCSP often refines the technical specifications and user experience of the final system and creates opportunities for realistic user testing. Pedagogically, students learn about detail design for how to develop a prototype to a higher functional standard, mitigate integration challenges, and refine a system's capabilities. The FCSP begins to emphasize the refinement of the prototype-context story to portray how the world could be with the prototype in the world.

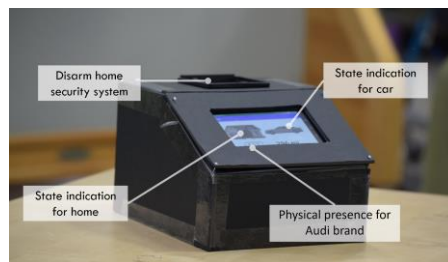


Figure 6. Team Audi's vehicle and home security FCSP

FCSP Case Study

Context: Team Audi observed that many home security systems go unused, as homeowners neglect to arm the system when they leave. The team prototyped a system for a car key fob to arm and disarm a home security system quickly and securely to bring the Audi brand into the smart home.

Prototype: Their functional system prototype aligned all of the key system elements: sensors, interface design (including key functional requirements for integration into a smart home ecosystem), and electronics in the key fob (RFID for demonstration). Aesthetics were not a focus of this prototype, as indicated by the rough foam core, tape, and nail construction.

Outcome: The team user tested several different physical forms, converging on a bowl design that would detect the key's presence and deactivate the user's security system. The ergonomic display on the front of the box allowed users to interact with their smart home and vehicle simultaneously. Users were delighted by the easy interaction and smart home system overview, as well as the affordance of the implicit motivation to consistently store keys in the same place so they don't get lost.

4.7. Part-X is Finished (PXF)

Part-X is Finished is a high-fidelity component that will eventually become part of the final design system. During the development process of a system prototype, system components are in an ambiguous flux state due to component interdependency resulting in a developing impasse. Forcing the design team to build one critical component, PXF, of the system design to a high fidelity that implements key parameters, such as size, scale, or capacity, helps break the circular chain of uncertainty. The primary pedagogical intention of PXF is to encourage student design teams to "put a

stake in the ground”. The completion of PXF is intended to boost student design team motivation and morale to build momentum for completion of the rest of the project.

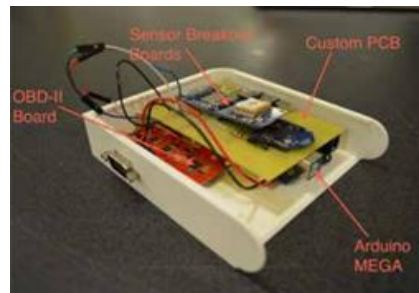


Figure 7. Team EMN's electric vehicle pedal Part-X is Finished

PXF Case Study

Context: Team EMN designed a system to reduce driver anxiety for electric vehicle adoption. The final system included a user interface, hardware subsystem, and software subsystem. The team converged on the hardware system and PCB as a relevant system to develop to a high fidelity to build from.

Prototype: Team EMN completed a custom PCB board and housing as a PXF for their final project. The team was then focused on the software, user interface, and integration for the rest of the project duration.

Outcome: Team EMN's PXF required minor tweaking for aesthetics but remained as a shared medium that the collaborating teams could use to ground their conversations.

4.8. Final prototype

The *Final Prototype* is a high fidelity, high resolution system that integrates fully functional prototyped and refined subsystems. Ideally, users test the Final Prototype, and the design team crafts a story to convey both surprise and delight. The prototype should have a well-developed appearance, technical implementation, and contextual story (role). As a part of the Final Prototype package, students are expected to develop a business case for adoption by the industry sponsor, as well as a coherent narrative on how the product will improve users' lives. The culmination of ME310 is a presentation of the Final Prototype's context, an interactive exhibit at a trade show known as EXPE, and a final document outlining the lessons learned throughout the project duration.

5. Conclusion

We reviewed the pedagogical approach of the Stanford University ME310 course and gave a snapshot view of the prototypes in use today to scaffold student learning. This paper does not cover every element of the course, nor exhaustively review every kind of prototype student teams have used. Rather, it presents the course's primary prototypes as a toolkit, where the appropriate prototyping methodology used by each team, and on each project is highly context specific. Every step in the prototyping process informs the next, and students constantly iterate and revise their plans based on user and teaching team feedback. Even though teams often struggle to meet the prototyping assignment's desired resolution on their first attempt, they still learn the prototype's underlying purpose, and are able to adeptly apply knowledge of prototypes' purposes later on in the course and beyond the course's completion.

The outcomes of this approach vary with each project. Some projects will emphasize user-experience and will need to prioritize user testing, while other projects will emphasize technical feasibility and require more functional testing to prove efficacy and future potential. It is up to the teaching team to guide the design process and adapt the pedagogy to the project's maturation process. Additional learnings from student's experiences of the course such as project management and intercultural communication skills may not reveal themselves until well after the course concludes. No design culture can persist for decades in isolation, without incorporating new concepts. We hope this snapshot stimulates conversation between different design cultures and pedagogical approaches. We also hope this paper codifies the current approach for future ME310 students and researchers, who will inevitably iterate on the course to changing times and student needs.

References

- Adams, J.A. (2005), "September. Human-robot interaction design: Understanding user needs and requirements", In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 49 No. 3, pp. 447-451, SAGE Publications, Sage CA: Los Angeles, CA.
- Böhmer, A.I. et al. (2017, June), "Prototyping as a thinking approach in design Insights of problem-solving activities while designing a product", In *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, IEEE, pp. 955-963.
- Brereton, M. and McGarry, B. (2000, April), "An observational study of how objects support engineering design thinking and communication: implications for the design of tangible media", In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, pp. 217-224.
- Buchanan, R. (1992), "Wicked problems in design thinking", *Design issues*, Vol. 8 No. 2, pp. 5-21.
- Bushnell, T. et al. (2013, September), "Using A 'Dark Horse' Prototype to Manage Innovative Teams", In *3rd International Conference on Integration of Design, Engineering and Management for Innovation*.
- Camburn, B. et al. (2015), "A systematic method for design prototyping", *Journal of Mechanical Design*, Vol. 137 No. 8, pp. 081102.
- Carleton, T. and Cockayne, W. (2009), "The power of prototypes in foresight engineering", In *DS 58-6: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 6, Design Methods and Tools (pt. 2)*, 24-27 August 2009, Palo Alto, CA, USA, 24-27 August 2009.
- Dow, S.P. et al. (2010), "Parallel prototyping leads to better design results, more divergence, and increased self-efficacy", *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 17 No. 4, pp. 1-24, <https://doi.org/10.1145/1879831.1879836>
- Dweck, C.S. (2008), *Mindset: The new psychology of success*. Random House Digital, Inc.
- Dym, C.L. et al. (2005), "Engineering design thinking, teaching, and learning", *Journal of engineering education*, Vol. 94 No. 1, pp. 103-120.
- Edelman, J.A. (2011), *Understanding radical breaks: media and behavior in small teams engaged in redesign scenarios (Doctoral dissertation, Stanford University)*.
- Faste, R.A. (1987), "Perceiving needs (No. 871534)", *SAE Technical Paper*.
- Houde, S. and Hill, C. (1997), "What do prototypes prototype?", In: *Handbook of human-computer interaction*, North-Holland, pp. 367-381.
- Kolb, D. (1984), *Experiential Learning: Experience As The Source Of Learning And Development*.
- Lande, M. and Leifer, L. (2009), "Prototyping to learn: Characterizing engineering students' prototyping activities and prototypes", In *DS 58-1: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 1, Design Processes*, 24-27 August 2009, Palo Alto, CA, USA, 24-27 August 2009.
- Leifer, L. and Steinert, M. (2011), "Dancing with Ambiguity: Causality Behavior, Design Thinking, and Triple-Loop-Learning", *Information-Knowledge-Systems Management*, Vol. 10, pp. 151-173, <https://doi.org/10.3233/IKS-2012-0191>
- Nielsen, J. (1989, September), "Usability engineering at a discount", In *Proceedings of the third international conference on human-computer interaction on Designing and using human-computer interfaces and knowledge based systems (2nd ed.)*, Elsevier Science Inc, pp. 394-401.
- Patnaik, D. and Becker, R. (1999), "Needfinding: the why and how of uncovering people's needs", *Design Management Journal (Former Series)*, Vol. 10 No. 2, pp. 37-43.
- Rittel, H.W. and Webber, M.M. (1973), "Dilemmas in a general theory of planning", *Policy sciences*, Vol. 4 No. 2, pp. 155-169.
- Schön, D.A. (1938), *The reflective practitioner*, New York, 1083.
- Schrage, M. (1996), "Cultures of prototyping", *Bringing design to software*, pp. 191-205.
- Sirkin, D. and Ju, W. (2014, October), "Using embodied design improvisation as a design research tool", In *Proceedings of the international conference on Human Behavior in Design (HBiD 2014)*, Ascona, Switzerland.
- Tang, J.C. (1989). *Listing, drawing and gesturing in design: A study of the use of shared workspaces by design teams*.
- Wiesche, M. et al. (2018). *Teaching Innovation in Interdisciplinary Environments: Toward a Design Thinking Syllabus*.
- Yang, M.C. (2005), "A study of prototypes, design activity, and design outcome", *Design Studies*, Vol. 26 No. 6, pp. 649-669.