International Journal *of* Designs for Learning
2018 | Volume 9, Issue 1 | Pages 49-79

DEVELOPING MULTI-DISCIPLINARY SKILLS THROUGH A COURSE IN EDUCATIONAL SOFTWARE DESIGN

Marisa Exter, *Purdue University*

This design case covers a graduate course in educational software design that focuses on semester-long projects in response to client requests. The course was intended to address the need for professionals across disciplines, such as instructional design, computer science, and human-computer interaction design, to usefully collaborate on educational software projects. The ability to work on a multi-disciplinary team was fostered in several ways: through recruiting students across multiple majors; providing readings and student presentation topics related to language, processes, and techniques used by each discipline; and by scaffolding the work of multi-disciplinary student groups in a major semester-long project.

Marisa Exter is Assistant Professor of Learning Design & Technology at Purdue University. Her research focuses on design and technology education, 21st century skill development, and multi-/transdisciplinary learning experiences.

https://doi.org/10.14434/ijdl.v9i1.23413

INTRODUCTION

This design case focuses on a new course on Educational Software Design. Some of the main goals of this course include learning basic terminology, techniques, and processes used across design fields, as well as gaining deep experience working with team-members from a range of backgrounds through engagement in a realistic project. The need for this course was supported by my own research and extant disciplinary education research.

My research on the experiences of educational software designers has revealed that these professionals generally design and develop software as part of a multi-disciplinary team (Exter, 2012, 2014). Analysis, documentation, and testing are performed at multiple levels, including but not limited to appropriately identifying and scaffolding learning, creating maintainable software architecture, and designing a friendly and usable interface. Therefore, creating a complex, high quality educational software typically involves some degree of collaboration between professionals from multiple disciplines, such as instructional design, computing, and human-computer interaction. In order to effectively collaborate, even the most specialized individuals benefit greatly from fully understanding the terminology used and the abilities of other specialists with whom they interface or co-design. This is also important for all members of a team, but particularly for those with project management capabilities or who are involved in overall product design, to understand the different design processes that cross the lifecycle of a product, such as ADDIE or other ISD models in instructional design, or software engineering models used by a computing team. My own experience as a software developer, software designer, instructional designer, and day-to-day project manager aligns well with the experiences shared by my research participants.

The people who work in software design and development came from multiple formal educational paths, including holding degrees in instructional design, computer science or other computing fields, and a variety of other more eclectic backgrounds (Exter, 2012). Regardless of formal education, nearly all involved in creating educational software play

multiple roles simultaneously, and additional roles across the duration of their careers, including instructional design, project management, software engineering, software development, and quality assurance. Furthermore, most members of software design teams need to gain an understanding of the needs of clients or users and to interface with subject matter experts and team members with a variety of expertise. However, this often was not covered extensively, if at all, across formal educational paths.

Instructional designers increasingly work on cross-disciplinary teams (Koszalka et al., 2013) and spend significant amounts of time on project management and in meetings (Cox & Osguthorpe, 2003; Ritzhaupt & Martin, 2014; Ritzhaupt, Martin, & Daniels, 2010). They often are intensely involved in project conceptualization but not in project development, leaving little opportunity for them to understand the process or offer their own insights and innovation as part of a collaborative development team (Hooper, Rook, & Choi, 2015; Smith, 2008). Yet, they must make complex judgments related to multiple aspects of each design, many of which are not directly related to learning outcomes or ID theory (Gray et al., 2015).

Similarly, computing professionals must be prepared not only to work with specialists from other fields, but also either become highly knowledgeable in the domain, or be able to work closely with a variety of subject matter experts and stakeholders to understand their needs (Dieste, Juristo, & Shull, 2008; Hofmann & Lehner, 2001; Niknafs & Berry, 2012, 2017). For example, degrees in computer science and software engineering generally do not prepare students to work with users from different age groups or in contexts such as school classrooms. In practice, software design professionals recommend offering students the opportunity to engage in large scale, cross-disciplinary team projects of realistic scope and complexity (regardless of the domain) to help them gain these skills that are in demand in the workforce (Exter, 2014; Exter, Caskurlu, & Fernandez, n.d.; Exter & Turnage, 2012). Similarly, user experience design professionals find communicating with others who use industry-specific jargon or work processes to be a major hurdle when they first enter the workforce (Gray, 2014).

The goal of the graduate course in Educational Software Design described in this case was to provide exactly this type of an experience. Students from varying backgrounds participated in complex design problems that related to education or instruction. They were grouped in multi-disciplinary teams based on student skill-sets and preferences and were guided through the entire design process. Course readings exposed students to terminology, processes, and techniques used in several disciplines, including instructional design (ID), software engineering (SE), human-computer interaction (HCI), and graphic design.

Course Designer & Instructor

I am an Assistant Professor in the Learning Design and Technology program at Purdue University. My professional background includes software design, development, and project management. These experiences allowed me to expand my understanding of the knowledge obtained from a Bachelor of Science and Master of Science in Computer Science. During my seven years in industry, I worked primarily on back-end systems with minimal to no user interface requirements. My interest in improving educational systems, particularly computer science education, inspired me to seek a PhD in Instructional Systems Technology. While obtaining my PhD, I had the opportunity to become involved as a Graduate Assistant in an educational software project. Over eight years, I worked my way from the lead of design and development to project manager on the project. The project spun off into a small company and I became director of Design, Development, and Testing. I received an opportunity to experience the early stages of developing a business and gained insights into the challenges of successfully bringing such a product to market. The combination of my education and experience provided an appreciation for the complexity of designing educational software and of working on a cross-disciplinary team with members who not only spoke different languages but also looked at design processes and products through different disciplinary lenses. This in turn inspired my dissertation topic: relating the professional and educational experiences of in-practice educational software designers.

Design Inspiration

Knowing that I had benefited from my own cross-disciplinary professional and educational experiences, I was inspired as a doctoral student to contemplate designing an experience for instructional design students that would foster an appreciation for and an understanding of the language and processes of other fields; provide a sense of what it is like to collaborate with colleagues from other disciplines; and provide experience with the complexity of designing software for real clients and users. When I began my position as an Assistant Professor, I had the opportunity to think seriously about developing a one-semester course. I initially envisioned this as a course for instructional design students in which they would learn to program and go through the entire design, development, and test process. However, I was very concerned that this would not be realistic—after all, in the one-semester senior capstone course I took during my undergraduate program, computer science students barely got past the initial design phases, while two-semester capstones often result in a small prototype rather than full-fledged software. Furthermore, training instructional designers to use a software engineering process and do some programming would not give them experience in working with specialists in other fields, one of my main goals.

I had several lengthy discussions with Dr. Micah Modell, a friend with a background similar to my own. After talking through my quandaries, he suggested that I review my dissertation findings to identify the most important skills for an educational software designer. Helping instructional designers understand and respect the important roles played by others specialists was important to me. These include, but are not limited to, software designers, human computer interface designers, and graphic designers. Each of these disciplines who not only design and develop particular aspects or components of the software, but also use very different types of analysis and testing than those included in instructional design models. Students in these fields had a similar need to gain interdisciplinary experience. The real world requires interacting with specialists from multiple fields, as well as with users in particular domains such as educational software. This was the inspiration I needed to focus the course on the design process.

My own learning experiences, including reading and learning about design, conducting research work, and observing colleagues who applied a studio approach to teaching design, helped me realize that this was the pedagogy I wanted to emulate in my educational software course; however, practical constraints prohibited the full realization of a studio model (notably, the slightly less than three-hour per week time-slot allotted to the class). These experiences provided mental precedent for my adaptation of the studio model for this course. I had previously taken studio courses in human computer interface design and instructional graphics design as part of my PhD coursework at Indiana University. As a student in Elizabeth Boling's research group at Indiana University, I participated in ongoing reading discussions on literature related to design pedagogy. Later, when I accepted a faculty position at Purdue, I joined a group of faculty fellows who designed a transdisciplinary undergraduate program that relied heavily on studio-model pedagogy. As part of this effort, I worked closely with several faculty members who had deep experiences with studio teaching, including Dr. Colin Gray (human computer design and development), Richard Dionne (theatre), and Davin Huston and Amy Van Epps (technology and engineering). I had an opportunity to observe their teaching style and interactions with students within my roles as program evaluator and researcher. During this time, I co-authored articles related to the use of critique in an instructional design course and the incorporation of a studio model into an undergraduate transdisciplinary program, as well as other related topics.

COURSE GOALS

My goals for the course were to provide learners:

- A basic introduction to the language, design processes, and techniques used by several fields, including instructional design (ID), software engineering (SE), graphics design (GD), human computer interface design (HCI) and quality assurance (QA).
- An opportunity to collaborate with peers as part of a cross-disciplinary team.
- Experience working through a systematic design process while designing software for a real client and users.

CONTEXT & STUDENTS

This course took place at Purdue University, a large land-grant, research-intensive university in the Midwest. This university is well known for its science, engineering and technology programs, but also has a strong College of Education. The Educational Software Design course was offered within the Learning Design and Technology (LDT) program in the department of Curriculum & Instruction. The LDT program awards both Master's and PhD degrees.

The Educational Software Design course was offered to LDT Master's and doctoral students, as well as other graduate students from across the university. The can serve as one of the major electives for LDT students and a general elective for other students. The 11 students who completed the course came from LDT, Engineering Education, Technology, and Linguistics programs.

Although none of the students had worked professionally on educational software, several had related experience: (a) three students had significant software development experience (including one game developer, one embedded systems developer, and one who had worked in data analysis and large data systems), (b) two students had significant instructional design experience, and (c) three had graphic design experience, including one who focused on human-centered design.

All of the students had some professional experience. One was a current faculty member at Purdue, two currently previously taught at a local community college, and the others had some teaching experience, either as graduate students or in other similar capacities.

Students came in with a range of goals. Several hoped to work on a software project of their own design. Others wanted to learn more about instructional design or had a foundation in instructional design but wanted to learn more about software design. For a few students, this course merely served as a convenient elective.

This diversity of student backgrounds and goals presented a significant design challenge. While I attempted to address this challenge in the course design, I did not know exactly what to expect until I surveyed students a week before the semester began. Even then, I was surprised at how different students' backgrounds informed their understanding of, and interaction with, the material. For example, two students had significant amounts of experience with programming (one

who used it to support the use of sophisticated statistical techniques, the other to interface with embedded systems); however, both had very little experience with software design techniques or the Unified Modeling Language (UML) visual language used in software design. Another student was pursuing a computer graphics degree, but had taken coursework in instructional design. Students' goals impacted what they hoped to get from the class. For example, one student majoring in Learning Design and Technology planned to pursue a career in university administration, indicating that his primary interest in the course was to become an informed consumer of educational software consulting services. This impacted what he perceived as being the most interesting or relevant aspects of the course.

COURSE DESIGN PROCESS

Course Traits

The initial traits envisioned for the course included:

1. *Attract students from across disciplines and give them experiences working together.* This would allow students to experience working in a cross-functional team, as would happen in a real-world setting.
2. *Include a semester-long project for a real client in K-12 or higher education.* This would create a real-life situation in which students need to focus on a client's vision and desires, rather than on meeting an instructor's requirements. This would encourage students to understand clients' desires, learn the process being used, bounce ideas off of each other, share documentation that could be understood by non-specialists, and negotiate deliverables.
3. *Provide an introduction to the design processes used by several fields and require students to work through a full design process during the semester.* This would allow every student to come away with a high-level view of how specialists following multiple disciplinary design processes work together to create a piece of educational software. In the future, students would be able to tackle small projects on their own and become knowledgeable colleagues to specialists in other fields.
4. *Provide studio space and time, allowing students to engage with one another during class time.* This allowed students time and space to work on deliverables during class time (although outside work time also was required). As part of the studio set up, they also were required to give critique and serve as test subjects for their peers' interview protocols, usability tests, and other activities.
5. *Provide opportunities for purposeful iteration of concepts based on feedback through instructor and peer critique, client feedback, and user testing.* Iteration is an important part of the design process, allowing both problems and solutions to evolve over time as new understanding is gained. In addition to team brainstorming, critique, and client feedback, students would interact with target users (such as students). These interactions would serve to drive teams to revisit a problem frame and design choices, thereby inspiring design iterations.
6. *Minimize the number of lectures, reserving time for the above items.* Even before deciding on the studio model, I was certain I would like to use an active learning approach, allowing students to learn from the experience by applying concepts introduced in readings to the real design project. This would take the emphasis away from me as an instructor, and ensure we had plenty of time to focus on the actual experience of designing software as part of a multi-disciplinary team. This also was intended to subtly communicate that I was not an expert in each of the domains. We all had expertise to share and things to learn.

Course Topics and Readings

The project was central to the course. However, students needed access to the disciplinary knowledge and skills to complete their projects. To overcome this issue, I developed a list of key topics across several domains and began to collect resources for each. The following topics were selected for the course:

- Instructional design models *(ID)*
- Software Engineering design process models *(ID)*
- Learner & needs assessment *(ID)*
- Gathering/eliciting requirements *(SE)*
- Visual design languages: UML, use cases *(SE)*
- Rapid prototyping & iteration *(SE/ID/design)*
- User interface design *(HCI)*
- Graphic design basics *(GD)*
- Usability considerations *(HCI)*
- Usability testing *(HCI)*
- Types of software quality testing *(SE)*
- Formative & summative evaluation *(ID)*

After reviewing my personal library, I searched the university library and the Internet for additional resources, and approached friends for recommendations, including Dr. Colin Gray (HCDD) and Dr. Esteban Garcia (CGT) at Purdue and Dr. John Jeffry (Computer Science) from Elmhurst College. The topic list was refined based on my reading and again as I planned course activities.

Topical Presentations and Activities

Although I anticipated that many of the students would come in with skills from one or more design disciplines, I could not count on having all disciplines represented, and I did want all students to acquire some base-level knowledge

and skills in each area. While I recognized that course lectures were necessary to bring students up to speed about the processes and languages used in each field, I felt that lecturing on these topics would take away from the student-directed environment I hoped to create. I also wanted to emphasize that all students brought knowledge and skills into the course rather than elevate myself as an "expert" in all areas (which I am not). Therefore, I quickly settled on the idea of having students present key topics.

A limitation to this approach was that I might not be able to attract students with expertise in each area every semester. Therefore, I assembled a list of readings for each topic, which were accessible to students who lacked expertise. I further narrowed down the reading list in an attempt to avoid overwhelming students. I focused on those readings that provided clear, high-level overviews and examples that could be used as models for project work. I planned for student presenters to read all of the readings from each week (required and optional) and present based on these. I also made additional materials available for those who wished to use them. I told students that they were welcome to use any approach they wanted in their presentation. Although I left it up to them whether or not to create an interactive activity, nearly all of them opted to do so. I therefore cut back on some of the activities I had planned, or used the additional time to focus on activities more directly tied to project work.

Many students selected a topic that they were already familiar with from their own disciplinary background, but others chose to stretch themselves and presented on areas new to them. All students appeared to have done research

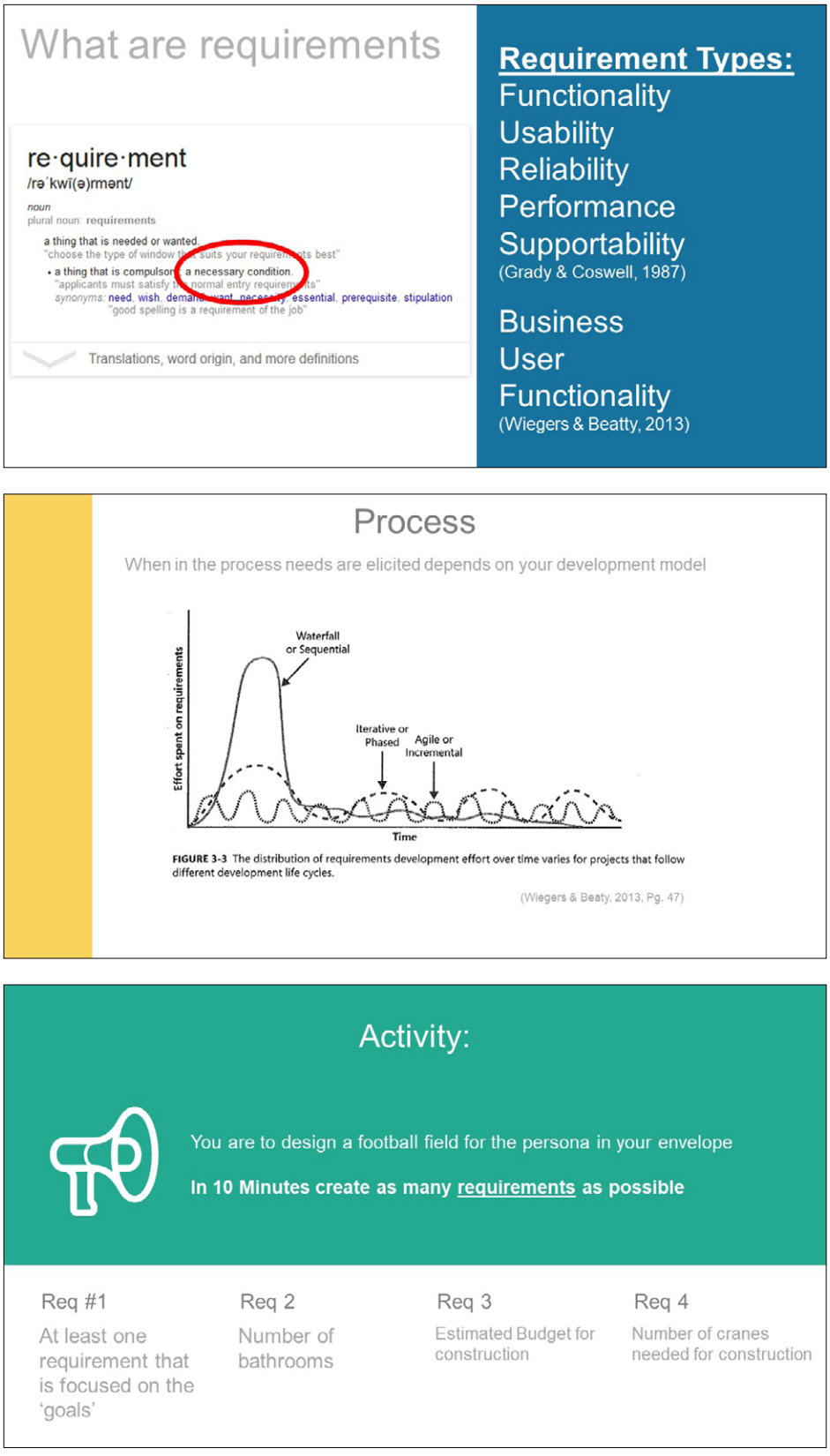


FIGURE 1. Student presentation on requirements (sample slides)

beyond the required readings and based presentations on their previous knowledge.

Student presentations varied, but generally included a PowerPoint presentation on the topic (see Figure 1 for a selection of slides from a presentation about gathering and writing requirements) and often included a related video or other resources. Students generally elected to include one or more short interactive activities related to the topic (see Figure 2, in which student pairs respond to an activity relating to visual design, and Figure 3, in which students remained in their project teams and designed their own process models after a presentation on software engineering models). These presentations were intended to be limited to 20-30 minutes, but the group often brought up questions that resulted in lengthy discussions. Each presenter would address questions, followed by additional comments by me and other students. In many cases, I had to encourage discussions to draw to a close after an hour.

I was excited to see that several students began or ended their presentations with insights about the course and course materials. One of the concepts that I aimed to impart in the course - but was concerned would be difficult for students to fully grasp - was how design processes across different disciplines were similar but had unique aspects. I also wanted to foster an understanding that these design processes overlap chronologically (as opposed to fitting the entire Software Design process model into the "development" step of the ID process, as ID process models seem to imply). I was therefore thrilled that two students began their presentation explaining their epiphany about this very topic and shared a visualization of their understanding (see Figure 4, one student's visualization of the alignment of various process models).

The activities students organized were almost always designed to get peers working in groups—sometimes in their project teams, and other times they were put into different small groups or pairs. As the course instructor, I often participated in the activity as well. Activities varied per topic. For example, one student presented basic graphic design principles and then had students work in teams to redesign poorly designed street signs (see Figure 2). Another student presented usability test techniques and had each group write a brief usability test script, and then had groups switch members to practice the short test protocols. Although some activities focused on outside materials (in one case, having us examine some of the "10 worst websites of the year"), others connected the direction to the main course projects. For example, one student presented different software design models and asked each team to talk about what model they would use for their own project. Several teams chose to adapt one of the existing models for their own purposes and shared their reasoning (see Figure 3).

FIGURE 2. Student groups present their sign redesigns (on whiteboards) as part of a student-led activity.

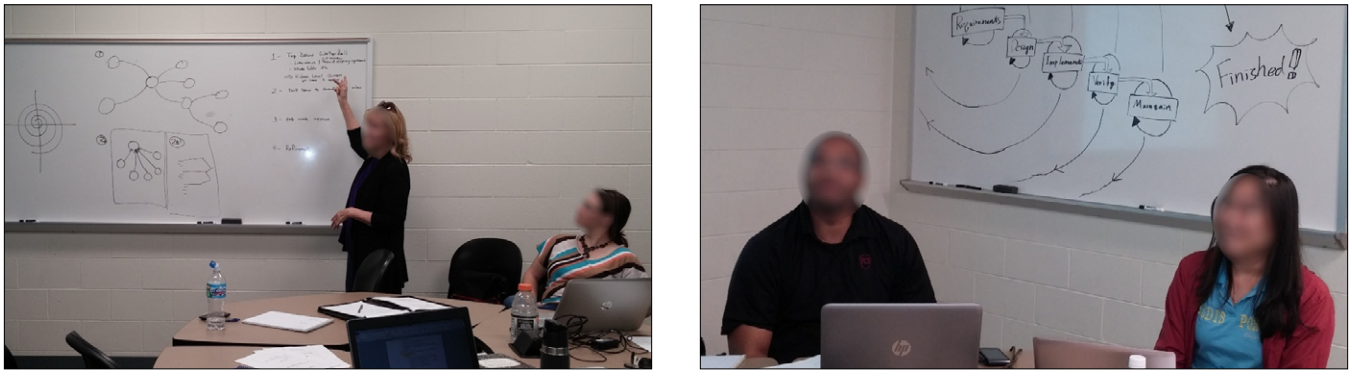


FIGURE 3. Student groups design their own process model to suit their project goals as part of a student-led activity.

Overall, I was very pleased with the student presentations. Students who expressed concerns that they were not very familiar with the material presented a high level overview based on the readings I had provided and came up with engaging activities. I tended to be more interactive during discussions on these topics and sometimes added information during the presentation if I felt it would be useful. Those who had relevant expertise tended to give a brief overview on the topics presented and then delved deeply into one or more aspects, usually introducing a depth beyond my own level of knowledge. Discussions after these sessions allowed less knowledgeable peers to gain a deeper understanding, as well as enabling the entire class to question, debate, and share relevant life experiences.

Project

Project Process Model

One of the major goals for the course was to introduce students to the life-cycle models or processes used by various disciplines, as well as some of the common tools and techniques used within the steps of those processes. Within each discipline, rather than recommending any one model, we would discuss the pros and cons of various models. This decision was driven, in part, by the sheer number of processes described in textbooks for each discipline. I was further influenced by my impression that the discipline-specific processes I had learned during my formal education (specifically the software engineering and instructional design process models) were often not directly followed in practice.

I also hoped to convey that processes used by the various disciplines overlapped with one another. For example, ID models appear to imply that the work of user interface designers and software developers falls within the "development" phase from an ID perspective. However, user interface designers and software developers must also learn about users' needs, document or at least review and discuss technical requirements to ensure they are feasible, and create designs of various types before beginning development. Each of these disciplines also has unique forms of testing that have

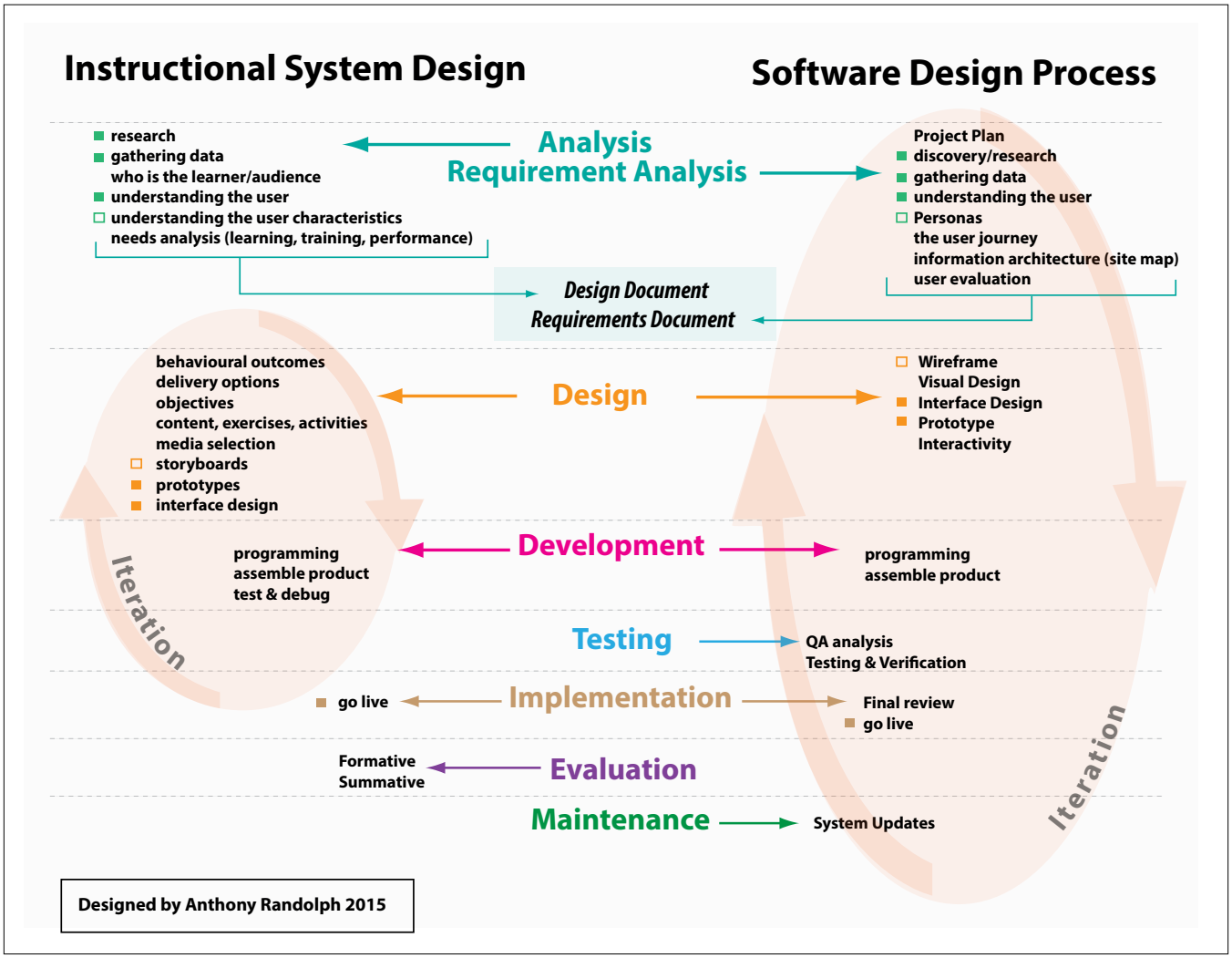


FIGURE 4. Student's vision of relationship between process models, as presented to the class.

ACTIVITY	DISCIPLINE DERIVED FROM	PROJECT DELIVERABLE
Team Forming	–	Team Charter
Learner & Needs Analysis *(including literature review and data collection from prospective users and other sources)*	Instructional Design	Design document (draft & final): Analysis
Requirements Elicitation & Requirements Writing; UML	Software Engineering	Design document (draft & final): Requirements Specification
Persona & Scenario Generation	Human Computer Interaction / Software Engineering	Design document (draft & final): Requirements specification
Interface Design	Human Computer Interaction / Graphic Design/ Usability	Design document (draft): initial design concepts; Design document (final): initial prototype (sketches/wireframes)
Prototype construction	Rapid design processes (in several fields)	Testable low-fidelity prototype (paper, PowerPoint, or web-based)
Usability Testing	Human Computer Interaction	Usability test script; Usability report
Software Quality Testing	Software Engineering	*In-class activity only*
Formative & Summative Evaluation	Instructional Design	Final report
Client meetings *(spread throughout semester)*	–	Client feedback reports; Impacts on design and design reports
Instructor, peer, and guest critique *(spread throughout semester)*	Art and Design	Instructor and peer critique reports; Impacts on design and design reports

TABLE 1. Project-related Activities, Origin, and Related Deliverables.

different purposes and that must occur at different points throughout the lifecycle of a product.

In practice, linear or cyclical models are rarely followed as documented in textbooks because design process is messy. When multiple design specialists interact, there is even more potential for those working on one aspect of the design to impact the process of working on another aspect. For example, when a software developer determines that a portion of the design is very difficult or not feasible to implement, they may recommend changes to the user interface design, even after interface designers believe their design to be complete.

Although I wanted to encourage students to consider a fluid, "no-right-answers" approach to using process models, their semester-long project had to have some structure. Otherwise, as in my own undergraduate project-based software engineering experience, students might find themselves barely getting through the first few steps of the process, resulting in students and clients' disappointment. Clearly, we could not work through a full formal lifecycle model for each discipline I intended to introduce. This was a struggle for me, because I wanted to give students experience with different techniques across disciplines, allowing them to understand the value of each component and the contributions of each disciplinary area to the overall project. Yet, I also wanted them to see the value of a shorter cycle: iterative rapid prototype model.

I decided to create a loose "hybrid" process that would allow students to experience working with some techniques from each discipline. This would guide them through the analysis, design, and early testing phases, and allow them to gain some testing experience. The goal I gave to students was to develop a prototype (which could be low fidelity) that would allow for some basic user testing.

After reviewing a variety of process models from across disciplines, I laid out the syllabus (see Appendix A) to guide students through each of the analysis, design, and testing activities, with deliverables assigned to align with each of these key steps as shown in Table 1 (see detailed schedule in Appendix B). Although due dates were specified in the syllabus, I was aware before the course began that these may need to be flexible based on teamwork and informed students accordingly as the course began. Although the syllabus, schedule, and Table 1 may indicate that the process would be linear, I encouraged students to continue to revise

their design and design documents (supported by submitting design documents multiple times—a draft, a "final" version, and finally a list of potential revisions as part of their final project submission).

Throughout the course timeline, teams were encouraged to return to earlier steps as they learned and experimented more with different design concepts. For example, after narrowing their target population or collecting data from users, I strongly recommended teams look back to research literature or other sources to help them learn more about the user group. Going through the design process also elicited more questions, which could potentially drive students to collect additional information from potential users and subject-matter experts, and to ask more thoughtful questions of their clients, after which they could reframe the problem or tighten the scope of the project. For this process, I took as mental precedent my observations of instruction offered by colleagues with significant background in studio pedagogy. These colleagues used generalized design models that allowed for fluidity between steps and multiple low-fidelity prototype iterations.

Each student group determined specific methods to use for their team projects. Learner and needs analysis were conducted through interviews, focus groups, surveys, and, in one case, walk-throughs of the current product with student and instructor users. Several activities encouraged students to come up with divergent visual design and interface design concepts (such as those shown in Figure 5) and then further refine the concepts (see Figure 6). I also stressed that they must determine underlying behaviors of the tool (see Figure 7).

Deliverables: Project Documentation

In order to avoid overwhelming students, I created templates and sample materials for each step, which were simplified versions of materials used in practice. The filled-in templates would become major deliverables for the class. While preparing for the class, I reviewed many templates found in books and online, then created a simplified version specifically for this course. For example, my learner analysis template included a table to fill in and add bullet point lists with suggestion

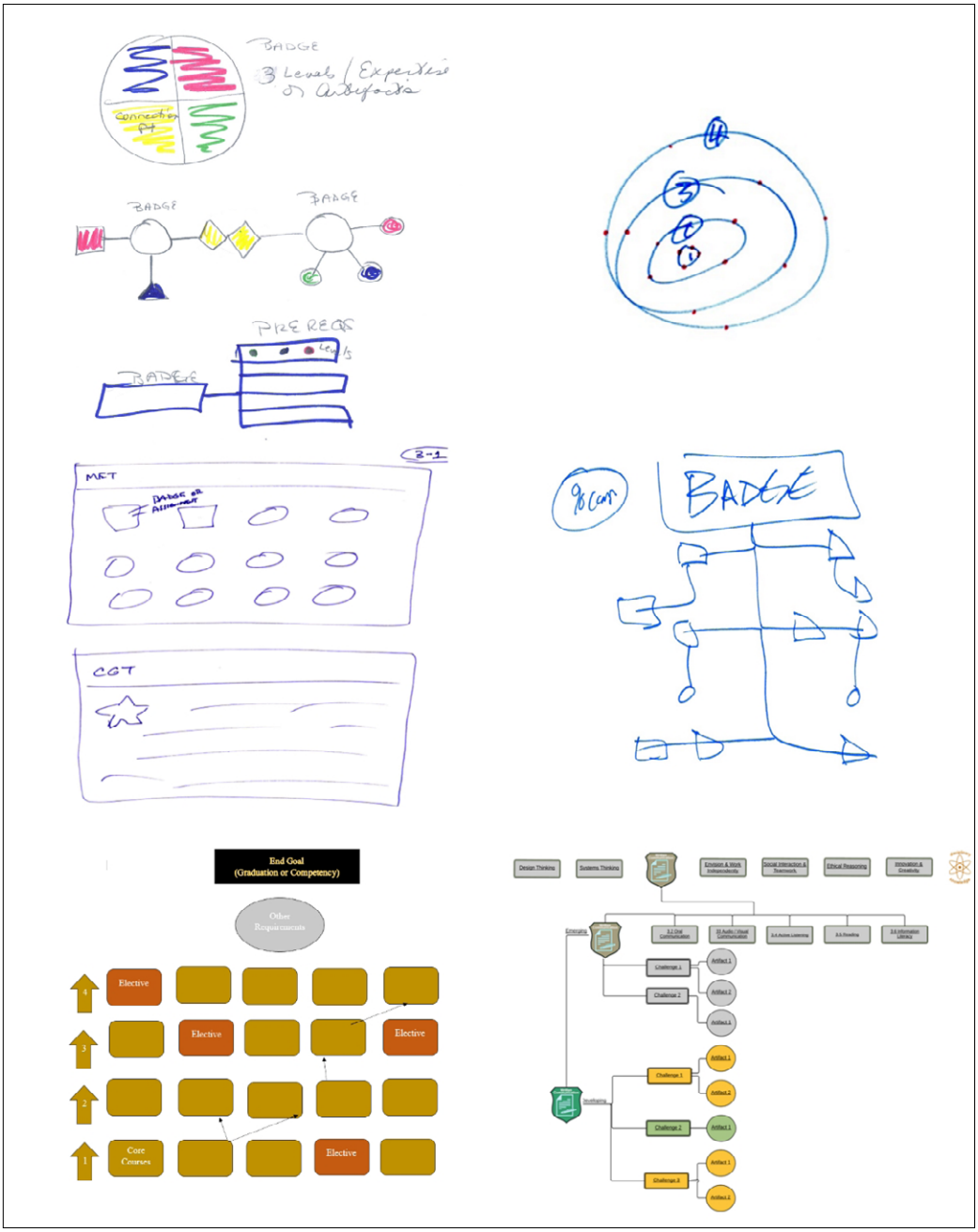


FIGURE 5. Ideation. Divergent concepts for representing related badges (Badge group)

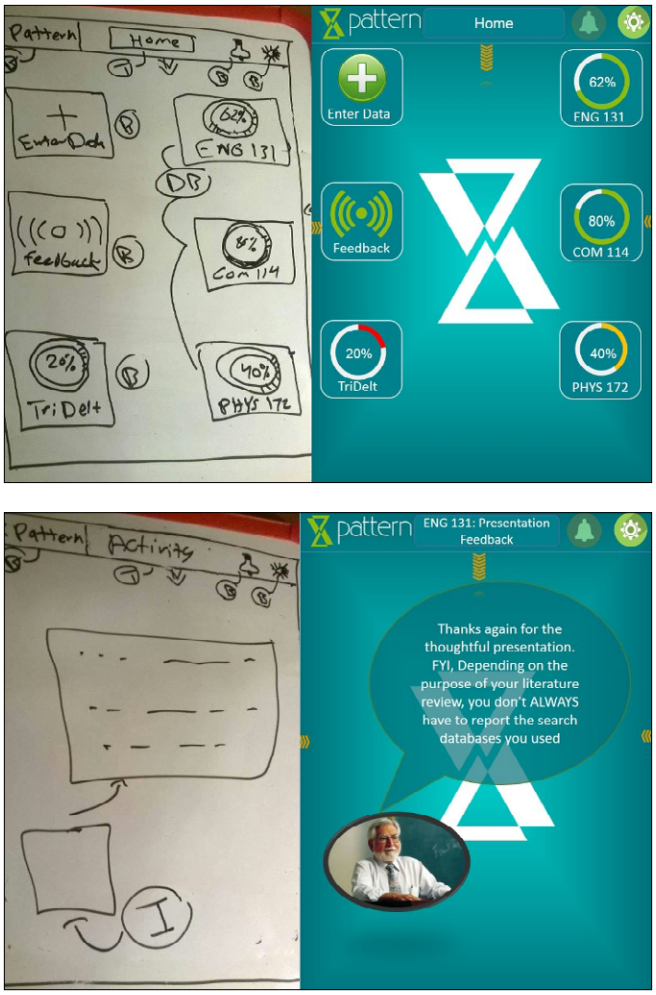


FIGURE 6. Refinement: Final design sketch translated into interactive prototype (Time management group).

about what to include in each section (See Figure 8, with a larger version in Appendix C). The requirements document template included a description of what should be included in each section (see Figure 9, requirements; larger version in Appendix C). The requirements document assignment also included a sample requirements document, including a simplified version of UML (a visual language used in software engineering for providing clear and specific documentation) for students to use as an example (Appendix D).

Groups received feedback on each deliverable and turned in a second version (although not noted on the schedule, the revised version of the usability test report was incorporated in the final document submitted at the end of the semester).

Student groups filled in the template documents directly with nearly no instances of adding additional sections. Any changes made were only to formatting. Although my intent was for students to use these documents to walk them through the process, I observed that students began to work on these documents only in class sessions immediately preceding the due dates for the deliverables.

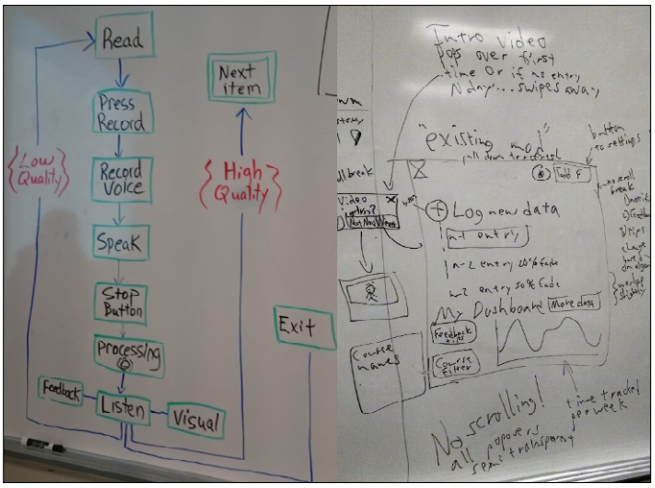


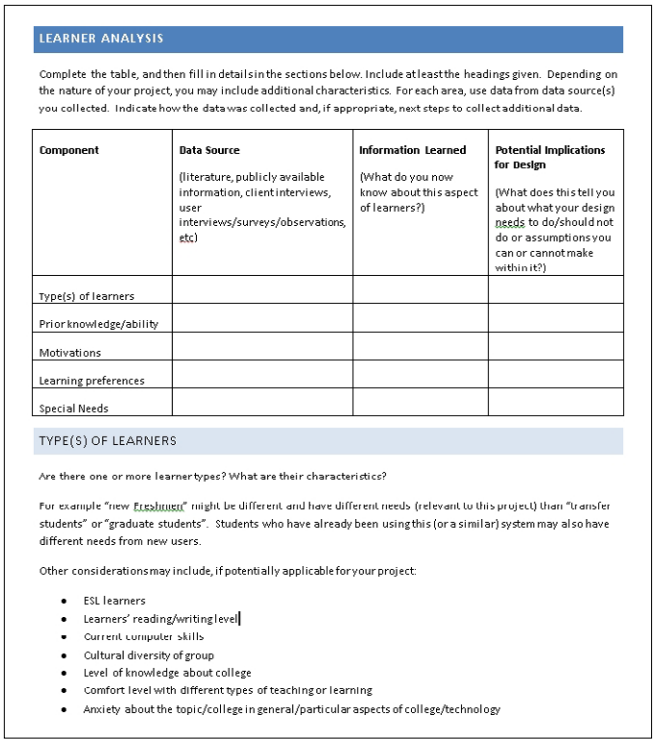
FIGURE 7. Underlying software behaviors (Japanese assessment, right; Time management, left).

Furthermore, I believed that documents were often written in a divide-and-conquer mode, based on differences in writing styles across the documents. Students did not say much about these templates at the end-of-semester debrief or in their reflection statements, other than to recommend breaking the documents up into more frequent and smaller deliverables.

I anticipated that student groups might struggle most with the technical writing components of the design documents, which were to include a simplified use-case diagram with detailed functional requirements. Despite the sample use-case diagram and a set of very simple requirements included as an appendix in the design document template, I found that I had to give a lot of feedback on these sections on students' first and even second drafts. This may be because the concept of a visual language like UML and the type of writing and specificity required to write technical requirements was completely new to the majority of students, but I believe that the learning process was further complicated by trying to use these new tools on designs that were not yet fully formed. While an experienced software designer might find creating this documentation helps students to organize their thoughts, the same activity may not do the same for my students—especially as, to my surprise, even students with significant software development experience, within their disciplines of engineering or game design, were not familiar with these types of documentation.

Clients & Project Requests

When locating potential clients, I aimed for realistic, complex design problems for students to tackle. I wanted projects focused on "educational software," but for practical purposes, I allowed this to stretch to any software that supported education in some way, allowing me to work with clients that were interested, easy to access, and did not have concerns about intellectual property issues. It was not easy



LEARNER ANALYSIS

Complete the table, and then fill in details in the sections below. Include at least the headings given. Depending on the nature of your project, you may include additional characteristics. For each area, use data from data source(s) you collected. Indicate how the data was collected and, if appropriate, next steps to collect additional data.

Component	Data Source (literature, publicly available information, client interviews, user interviews/surveys/observations, etc)	Information Learned (What do you now know about this aspect of learners?)	Potential Implications for Design (What does this tell you about what your design needs to do/should not do or assumptions you can or cannot make within it?)
Type(s) of learners			
Prior knowledge/ability			
Motivations			
Learning preferences			
Special Needs			

TYPE(S) OF LEARNERS

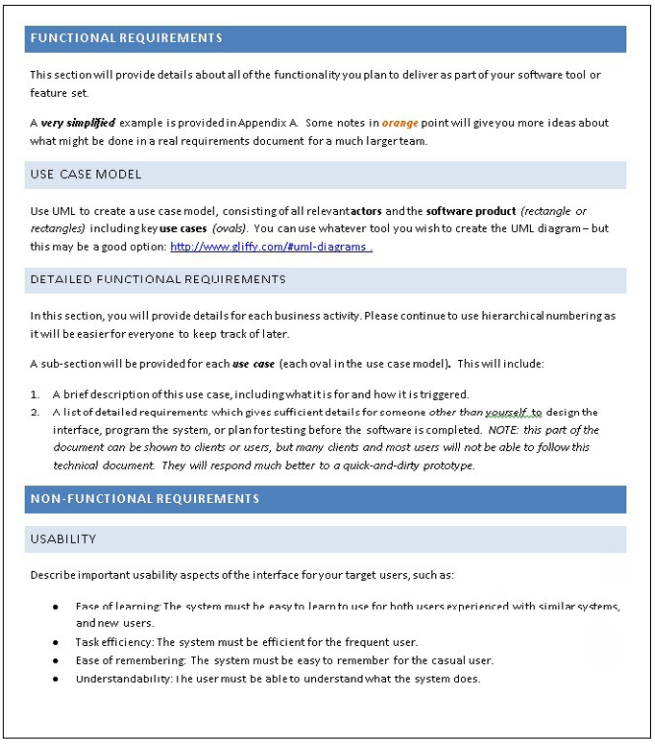
Are there one or more learner types? What are their characteristics?

For example "new Freshmen" might be different and have different needs (relevant to this project) than "transfer students" or "graduate students". Students who have already been using this (or a similar) system may also have different needs from new users.

Other considerations may include, if potentially applicable for your project:

- ESL learners
- Learners' reading/writing level
- Current computer skills
- Cultural diversity of group
- Level of knowledge about college
- Comfort level with different types of teaching or learning
- Anxiety about the topic/college in general/particular aspects of college/technology

FIGURE 8. Section of Analysis Document Template (enlarged version in Appendix C)



FUNCTIONAL REQUIREMENTS

This section will provide details about all of the functionality you plan to deliver as part of your software tool or feature set.

A ***very simplified*** example is provided in Appendix A. Some notes in ***orange*** point will give you more ideas about what might be done in a real requirements document for a much larger team.

USE CASE MODEL

Use UML to create a use case model, consisting of all relevant **actors** and the **software product** *(rectangle or rectangles)* including key **use cases** *(ovals)*. You can use whatever tool you wish to create the UML diagram – but this may be a good option: http://www.gliffy.com/#uml-diagrams .

DETAILED FUNCTIONAL REQUIREMENTS

In this section, you will provide details for each business activity. Please continue to use hierarchical numbering as it will be easier for everyone to keep track of later.

A sub-section will be provided for each ***use case*** (each oval in the use case model). This will include:

1. A brief description of this use case, including what it is for and how it is triggered.
2. A list of detailed requirements which gives sufficient details for someone *other than yourself* to design the interface, program the system, or plan for testing before the software is completed. *NOTE: this part of the document can be shown to clients or users, but many clients and most users will not be able to follow this technical document. They will respond much better to a quick-and-dirty prototype.*

NON-FUNCTIONAL REQUIREMENTS

USABILITY

Describe important usability aspects of the interface for your target users, such as:

- Ease of learning: The system must be easy to learn to use for both users experienced with similar systems, and new users.
- Task efficiency: The system must be efficient for the frequent user.
- Ease of remembering: The system must be easy to remember for the casual user.
- Understandability: The user must be able to understand what the system does.

FIGURE 9. Section of Requirements Document Template (enlarged version in Appendix C).

to find potential clients who would agree to the caveat that in exchange for their time they would receive a design but *not* fully developed software. I began by reaching out to personal contacts and asking for referrals. Unfortunately, as this was only my second year at the university, I did not have a very rich collection of personal contacts, and those whom I talked with did not have any projects in mind, or may not have had a clear vision of what my class could offer them.

Although I began conversations with several individuals, in the end the group most interested was the university's own instructional technology team. This team often works with student interns, and they indicated having many concepts for potential tools or tool enhancements that no one in their own team had time to work on. They were especially interested in the potential for my students to conduct a learner analysis and come up with a new perspective on two of their existing software tools. The third project for the semester, which was pitched to me by a student before the course began, involved developing a software tool to be used by Japanese language students and instructors. Although students brought this project to me, I required them to locate someone who could serve as a "client" so that they would have the opportunity to engage with someone else who might have different ideas about what they wanted out of the project, just like other teams. Unfortunately (in my mind), the individual identified as a client already worked very closely with one of the team members and they had a common vision of both the problem and the potential solution. Therefore, this team did not experience the creative pushback that other teams faced, nor did they need to learn to communicate with a client whose background was different from their own (especially as the initiating student became the primary point of contact for this client).

Clients were asked to present their projects on the first day of class, before students decided their project preferences or were assigned to teams. Once teams were formed, they were expected to interact directly with clients, including scheduling and conducting meetings. I provided advice upon request about how to tackle issues or quandaries but did not interfere directly between students and clients. Clients were surveyed at the end of the semester (highlights of client survey results are presented in the section on Client Feedback).

Student Team Formation

Students were asked to fill out a brief questionnaire before the end of the first week of class. In this questionnaire, students were asked to rank the projects based on their preferences and list of skills they had to offer the team. I assigned students to groups based on their preferred projects and existing skill-sets, while ensuring that each team included students from across multiple disciplinary backgrounds. Fortunately, most students got their first pick as this allowed for diverse groups, so this was not as difficult as I had expected. Unfortunately, two students dropped the class in

the following weeks, but students volunteered to move to ensure there were at least 3 students in each team.

Intellectual Property Considerations

Before advancing too far with potential partners, I met with a representative from the university intellectual property office to learn about students' and clients' (or sponsors) rights. After several meetings, he consulted with another team to create a statement of acknowledgement of students' and the university's rights per federal law and institutional guidelines, which both students and client representatives signed.

Team Tools

Students were referred to an online resource including information about team charters and sample team charters (http://www.teamlti.com/charter/index.html) and required to create their own charter. However, I gave no specific guidance about creating and using the team charter. In retrospect, I believe I made this decision from my own lack of familiarity with team charters, or their use, in class projects. Another faculty member recommended I use one, and, after a brief online search, I decided to use the *teamlti.com* site as it seemed to give clear recommendations and had a number of samples. I assumed that the graduate students would make use of this resource as they saw fit. Student teams did not appear to have difficulty creating their own team charters, but did not appear to use them throughout the semester. Students later recommended that I provide reminders for teams to review their team charters because these would have addressed team issues experienced during the semester.

Students also were required to use a self and peer feedback tool described in Modell (2013), in which they rated themselves and their team-members on a weekly basis across 13 different prompts. In addition to encouraging students to reflect on their team interactions, this was intended to keep me up to date on how well various team members were doing. I planned to pair this with my own observations during in-class and informal discussions with teams during class time each week.

Unfortunately, because of a variety of technical issues we were not able to use the tool successfully for about the first third of the semester. Surprisingly, students did not mind using the tool despite these issues, and we discussed how we were essentially beta-testing a working prototype system. However, the results were harder to read than I had anticipated. Although there were some peaks and valleys in individual performance, to a large extent students appeared to be giving themselves and their peers equal scores throughout the semester (the design of the tool required scores across all students to add up to 100%, meaning that raising the score for one would require lowering it for others). However, overall there was much less variation between students than I would have anticipated based on informal conversations throughout the semester. Although I had planned to use this tool to aid me in assigning points to each student based on their individual effort, I felt that I did not have enough data to fairly differentiate between students.

Each student was shown the graphs created by the self-and-peer feedback tool during his or her debrief session. Students indicated they were not surprised about the results, as specific dips and heights tracked with specific events they recalled (such as graphic designers putting in a lot more effort during specific weeks, or weeks during which an individual was traveling or dealing with personal issues).

Multiple Opportunities for Critique

Because the intention was to model a studio design course, I planned for multiple opportunities for critiques of course projects. This emphasis on critique was driven by my experience learning from and teaching with Professor Elizabeth Boling while at Indiana University, as well as reading about studio pedagogy.

I realized that not all students would be familiar with the practice. Therefore, I planned both structured and un-structured activities which were intended to provide a friendly environment to help students not only improve their designs but also gain experience with giving and receiving critiques. Students received critiques in several ways. Fellow students gavecritique at multiple pre-set "peer critique" sessions (some on written documents, some in person, as shown in Figure 10), as well as informally discussing their projects in class. However, the latter occurred less often than I had anticipated because students generally kept to their own teams. Students asked clients for feedback at client meetings, which generally included students updating clients on analysis and design activities, and discussing design alternatives. Another faculty member with expertise in HCI attended two class sessions and spent around half-an-hour per group giving critiques on two occasions. Several groups also reached out to potential users in various ways to get feedback on early ideas or mockups.

I regularly circulated around the room engaging in informal group desk-critique, which varied from serving as a sounding board, to providing just in time instruction, and talking with them to sort out design problems. This time also became an opportunity for me to clarify communication problems, often attempting to translate or at least point out where disciplinary language was being misunderstood without either party realizing it. I also provided detailed written feedback on each report submitted.

In order to encourage students to make well-informed use of critique, I required them to reflect and respond to formal peer, instructor, and client critiques. The template shown in Figure 11 was used to record each critique provided, how

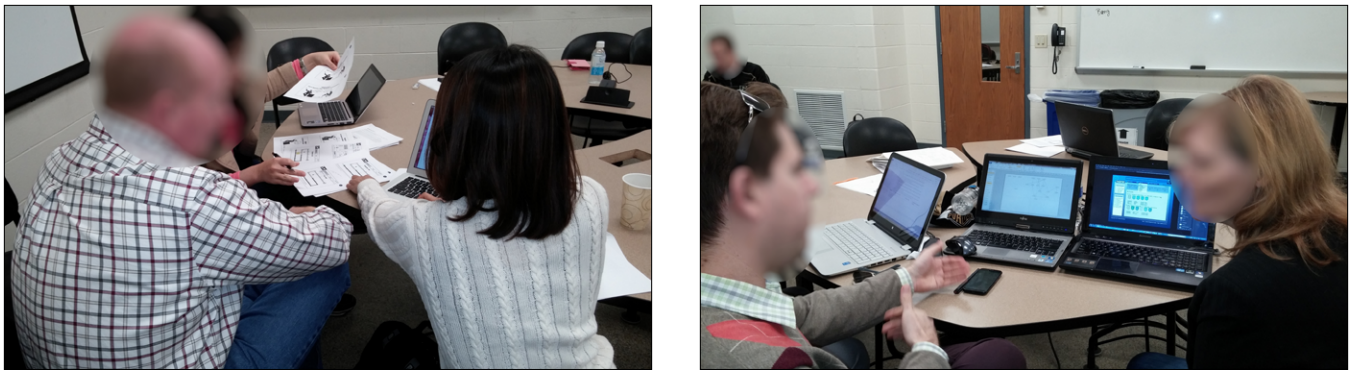


FIGURE 10. Peer critique of early prototypes.

Client Feedback Report

Overview of Client Meeting

- When and where did you meet?
- What topics were discussed (at a high level)?
- What new things did you learn? How will they impact your design process or your software design going forward?

Plan to Address Client Feedback

After each meeting with your client, you will fill out the following table. To consider as you record the client feedback:

- Each distinct item of feedback from the client should have its own row.
- "Next steps" are what you may do to follow up on the feedback. This could include doing additional research, following up with more questions, considering alternative designs, trying something out with users, etc.
- "Impact on your design" indicates changes you will make (or not make) based on client feedback. This may include:
  - Literally implementing the feedback
  - Making a change that addresses the underlying concern/problem in a different way (describe how and why you choose to do this)
  - Not implementing any changes because of lack of time/resources but adding to a list of potential future work (explain why you decided to do this)
  - Not implementing any changes because you do not feel they are appropriate/relevant to the design (explain why and justify how you are still meeting client's overall needs.)
- "Justification" of your next steps/impact on design. You will need to plan to be able to explain your decisions to clients at next meeting or upon questioning after a final presentation. Therefore, the justification should be clearly articulated.

Client feedback	Next steps	Impact on your design	Justification for next steps/impact on design

FIGURE 11. Feedback Report Template (See Appendix C for enlarged version).

the team intended to use the feedback, and the justification for their choices. Each critique received was generally reported in great detail, with every distinct comment recorded as a separate item. In retrospect, I wished I had provided more guidance on synthesizing across sources before making design judgments.

Grading

My goal was for students to achieve personal development rather than meeting a specific standard. Throughout the class I encouraged students to focus on my feedback, not grades, and therefore gave a large amount of detailed feedback. Similarly, I did not grade the quality of the final prototype. Rather, grades were based on how they went

about the process, as evidenced through their written materials and presentations. Illustrations of early designs and the final prototype were reviewed for evidence of evolution of the design based on their research, critique received, and user testing. The distribution of points aligned with this vision.

Special Guests

On the first day of class, experts were invited to present on intellectual property concerns at the university and project topics. My primary reason for this request was to ensure that students understood their rights and relationship with the clients (or sponsors) in terms of intellectual property. However, the IP experts also provided a broader overview about intellectual property rights, as well as bringing useful handouts. Students responded very favorably to the session and indicated that it was very useful in terms of explaining different types of protection for intellectual property (e.g., copyright, trademark, and patent) as well as their rights as Purdue students; most had not been aware that Purdue students retain intellectual property rights to anything created as part of a class project.

Later in the semester, an expert in HCI design was invited to attend two class sessions. In addition to presenting on wire-framing and user flow diagrams, our expert provided in-depth critique on students' interface designs.

Typical Evening in the Classroom

The course was offered from 5:30 p.m. to 8:20 p.m. on Thursday evenings, and took place in a classroom which included large, movable tables for group-work and whiteboards on every wall. A computer station projected images onto screens at the front and back of the room. Thirteen students originally enrolled in the course, but two left within the first week due to heavy course loads. This left eleven students, who worked in two groups of three and one group of four for the major course project.

On a typical course night, students generally talked enthusiastically with me or with each other until all students had arrived, usually a few minutes after the official start time. I would make brief announcements about upcoming deadlines and plans for the evening. Then, a student would present a course topic. After presentations were over, I would often introduce another activity related to group projects. Once completed, groups would continue to work on their projects. Students were encouraged to work on whiteboards and paper during design phases, although a number felt more comfortable working on laptops. During group work time, I would check in on each group one or more times, asking questions and offering comments or critique. After the class ended for the night, multiple students would stay to talk with me, either to offer additional thoughts of their own about course topics, or to express private concerns about their group's progress and other group-related issues.

STUDENT REFLECTIONS ON COURSE EXPERIENCE

Students were asked to write reflections at the end of the semester, responding to four prompts:

1. Describe a unique, surprising, perplexing, problematic, affirming or engaging event that occurred during your design process. What did you learn from it?
2. As you reflect back on the entire semester, what did you learn about the process of designing educational software?
3. What did you learn about working on a cross-disciplinary team?
4. If you had a chance to work on a similar project in the future, what would you do differently? What would you do the same?

Students also had individual debrief sessions with me during the last day of class. The quotes in this section come from these two sources.

Cross-Disciplinary Experience

Students indicated that they appreciated learning about the mindset, tools, and processes used in multiple design disciplines. One student, who had significant experience as an entrepreneur, and in teaching design, explained:

> *Over the years, I have generally become familiar with the software development process, the design process, customer analysis, and educational environments. However, this has been the first opportunity for me to really put those different things together. The biggest thing I learned is that there are processes specifically for this (this meaning… educational software design). Models like ADDIE that exist specifically to integrate these ideas are fantastic!*

Another contrasted her experiences within her technology-focused computer graphics technology program, and the emphasis on educational software within this course. "In technology they have a class called computer user interface design. In [that] class they focus on interface design, but they don't think about educational goals. They only focus on the commercial product."

A student with game design and development experience currently pursuing a degree in LDT valued both the cross-disciplinary team experience and the opportunity to design *educational* software.

> *Before I took this course, I thought it would be an easy course since software design is not a new topic for me…. However, this course was a different experience for me. It*

helped me to learn more about the process of educational software design, and it provided me with the opportunity to work with a cross-disciplinary team on a very complicated software.

The value of participating in a course that presented material and provided experience with a variety of disciplines depended on each student's personal goals. One student who has entrepreneurial and administrative interests explained, "[This course allowed me to] know more about the software developers—get in the head of who they are, and how I can better work with them. [It will] improve my relationships going forward with designers and developers."

Several students indicated that they learned a lot while preparing for their topical presentations.

I thought I had the basic knowledge that would be good enough to give a presentation, but …I decided to learn more about Quality Assurance to give a 20 minute presentation. I wanted to give something, so whenever the class is done, [fellow students will] remember what they learned [about QA testing].

The same student indicated, "I loved all the presentations. Sometimes I learned about things they didn't think I would learn… a lot that is very applicable to me in game design… [for example,] graphic design is a language itself." Early in the semester, students requested that all presentations be uploaded in the course's LMS, and one student volunteered to video record sessions.

Varied but Growing Understanding of Design Process Models

Most students came to understand the concept of a process model, and the differences and similarities between models presented, to varying degrees. One student, who had experience in several design disciplines, was able to think deeply about similarities, differences and touch-points between different process models (his visual depiction of this understanding can be seen in Figure 4). As he explained,

An important element that I learn during this semester in this design process for the Japanese educational software is the parallels the Design process has with the ID Process. I developed this chart during the class which gave me a complete understanding of how my ID expertise fits in the Design process. Once I gain this understanding I was able to see how my input contributed to the software design process.

One of his team-members (who did not come from a design field) explained the team's thinking about developing their own process model (see second picture of Figure 3 for initial brainstorm, and Figure 12 for their final depiction of their process, as documented in their design document).

I never knew that there are different types of design [process models] and each one had a unique name. While someone was giving a presentation about design [process model] types, I was thinking our project might take the mixture of waterfall and spiral design though it was not on the article nor presentation, (it was either spiral or waterfall). However, there was somebody who thought the same thing as I did, and I learned it is good to envision different ideas from articles or presentations.

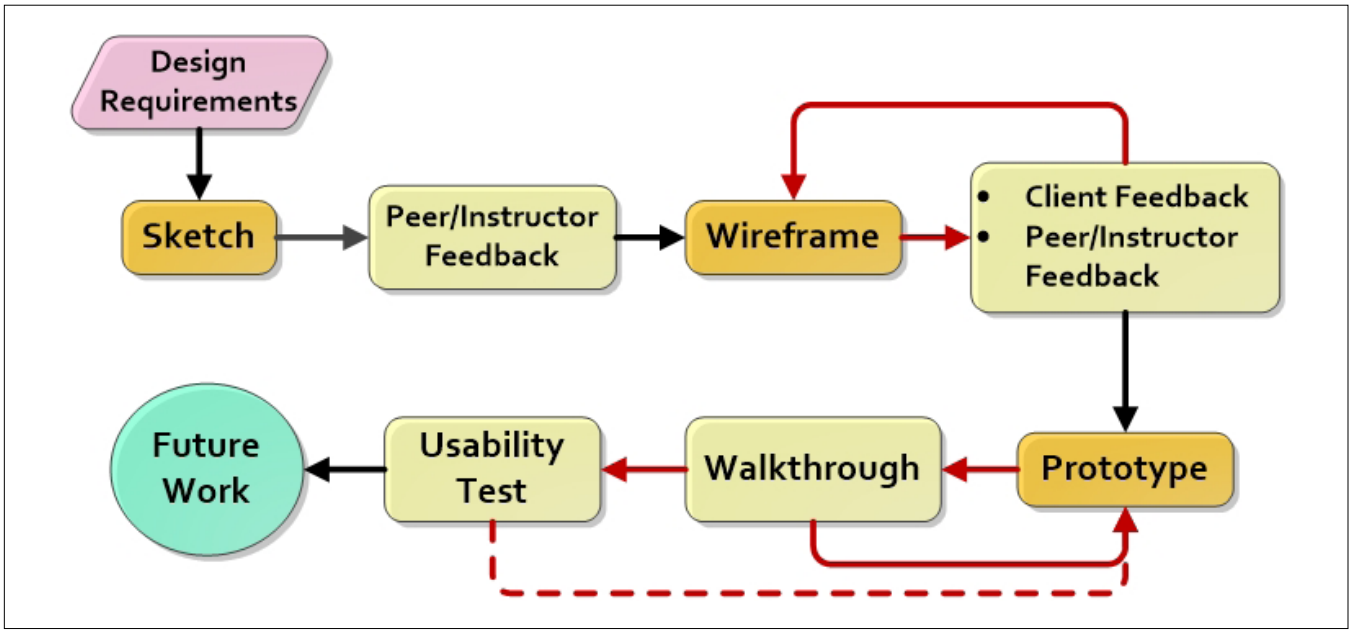


FIGURE 12. Student team's modified process, shared as part of final design document.

Even students with a background in instructional design or other design fields may not have experienced the more iterative and cross-disciplinary models encouraged in this class. As a student with teaching and practical experience in engineering design explained,

> *My department tends to treat the design of learning environments or experiences as generally a linear process. Even if I do not end up designing software, the tools I learned in my first experience in LDT will end up being useful to me in an instructional role (as a reminder that the design process I teach students really does apply to designing learning environments) and in a research role in how I look at the teaching and learning process. The frameworks for instructional design that we talked about, the UML (even though it is software specific I can see many research uses for this), and the requirements writing (which I have never considered in an educational context) were interesting new perspectives.*

Project experience

Clients & Project Definitions

Each team had unique experiences with their client(s) and with clearly defining the problem statement and scope. Student teams struggled with initial client presentations that did not clearly identify the problem and project parameters, or provided too many goals or questions without clear guidelines on what they hoped to see at the end. As one student described, "On day one, our client gave us a presentation to describe [the existing software] and the problem that they wanted us to solve. The presentation was not informative at all, and we were not sure what the problem was."

Although some students were frustrated with the client or the course design, others indicated that, in retrospect, they could have done more to work with clients.

> *I would ask for more meetings earlier with the key stakeholders. I would clarify what they are most interested in knowing/learning/solving. I would make sure I had access to a fully populated [student] profile [within the existing tool] so as to see how it works and displays.*

As described earlier, students were especially interested in the educational aspect of the software they were designing.

> *Even though we [had] defined the goals and objectives and needs for the software, one of the most different thing is that we have to consider more about how to improve learners' or instructors' experiences and performances. User Inter Design focus on the ease of use, but for educational software, we need to think more and combine learners' and instructors' perceptions with design.*

However, other students felt that this goal was not met. Across the duration of the course, I heard several comments that the software they were enhancing or redesigning was not truly educational, even though it was aimed at students and instructors (such as the digital badge system and the time management app). When speaking of her project, which involved enhancing the digital badging system, one student felt "[w]e were not really designing educational software… this was more a look at how a current product is functioning and how the interface needs to be improved."

In the end, at least one team felt uncertain about what they should provide. Should they focus on what the client appeared to want and benefit most from (learner and needs analysis) or on what was required for the course? I encouraged students to balance my requirements with those of the client by focusing on what the clients requested but also provide them a design as proof of concept. The students in this team did not appear to be satisfied with this compromise—interestingly, students seemed dissatisfied notbecause of workload, but because they felt that producing a design could mislead the client into believing the team endorsed the use of this tool. This concern was exacerbated by a sense that they could not convey their concerns fully to the client.

> *I would have liked to be more honest with [the client] about our groups' feelings on [the project] and the potential it has. Even in our final presentation I feel we were trying to make it look better than it is. False hope over honestly didn't feel right to me. When I presented this to my group it was rejected out of fear of getting a bad grade.*

(Team) Process

Analysis

In many ways, the teams did more than I expected them to do, especially in the area of analysis. I had anticipated that students would interview 3-5 students and instructors similar to those for whom the software was designed. To my surprise, the Japanese language group conducted a number of interviews and followed up with a survey distributed to all students currently enrolled in the course (46 responded), as well as all teaching assistants who taught the course and who were particularly interested in the development of this too (nine responded).

The time management app team were able to add related questions to focus groups that already had been planned for another purpose, allowing them to recruit 60 undergraduate students, as well as interviewing 3 faculty. The digital badge system group interviewed instructors, undergraduate students, and graduate students across three different programs. They later returned to a smaller number of students to conduct a think-aloud protocol on the current tool.

Analysis became a key phase for each of the student groups. As one student explained,

> *One of the biggest 'take aways' for…the project is from our last client meeting. [The client] said the two most valuable things to him were (1) The data we gathered about students/users and (2) the program map because it reflected a real/actual program. Out of all the design phases and process I thought it was powerful that he felt the front end analysis/data was one of the most impactful things for him.*

Students returned to analysis related activities throughout the semester, by accessing existing secondary research as well as returning to potential users (instructors and learners) and clients to learn more about the expectations and motivation of each.

Design Process & Creation of Early Prototypes

Students were encouraged to use information provided by the client and analysis data to inform their initial designs. They were encouraged to create "multiple diverse concepts" (ideate), then focus on one or two general designs and refine these (iterate). Subsequent iterations were informed by feedback from clients, peers, the instructor, and special guests. Coming out of a period of being focused on analysis, students were eager to charge forward to a 'solution'. However, as one student recommended when requested to write guidance for future students, "Embrace the process including ups and downs, creation and revision, agreement and disagreement. Rapid prototype is your friend but you can't be married to your ideas."

One team never seemed to diverge significantly from the initial concept that they agreed on very early in the semester. A team member described minor changes over time, based on feedback received. "There were many times I felt that our group came back to the same spot where we used to be along the designing process, but it was not completely the same as before." Another team member found that the process of trying out different alternative that nearly fully returned to the original design was an affirmation on the quality of their design.

> *One thing that I would characterize as an affirming event is that the nature and scope of the project returned to my initial conception of the project. Initially, I thought of the pronunciation practice module as an adjunct to existing courses. Nevertheless, I kept an open mind and we considered alternatives (including a self-study use). However, in the end we returned to the beginning. Moreover, it was not due to some Machiavellian manipulation on my part.*

He went on to describe why his team reverted to the original plan: It appeared to meet the needs of the clients and focused on an aspect of the problem not currently targeted by currently available commercial software packages. As the instructor, I could understand how the problem lent itself to the design produced, but had hoped they would branch out into more "wild" alternatives. The largest changes made were on the navigation page, not the main page that students spent most of their time interacting with.

Testing the prototype

Each team created a low-fidelity prototype to be used for usability testing. Although I recommended paper prototypes, all three at least initially began by creating interactive digital prototypes (see Figure 13). After testing these in class, two of the groups decided to print out each screen and use these printouts as a type of paper prototype. As the graphic designer in the Japanese Language team explained, they quickly moved from a somewhat interactive, on-the-screen prototype to paper printouts after initially testing the prototype with peers in class:

> *When it came time to implement it with actual testers, the prototype became big concerns. The mock web-based prototype looked realistic and I thought it would trigger testers' actual movement rather than PowerPoint or paper prototypes. However, it [would] stake some time to create a mock web-based prototype and it was not accommodating as much as other prototypes. On the other hand, the paper prototype which we did was easier to amend even after we found small issues, and testers had no problem imagining the real application.*

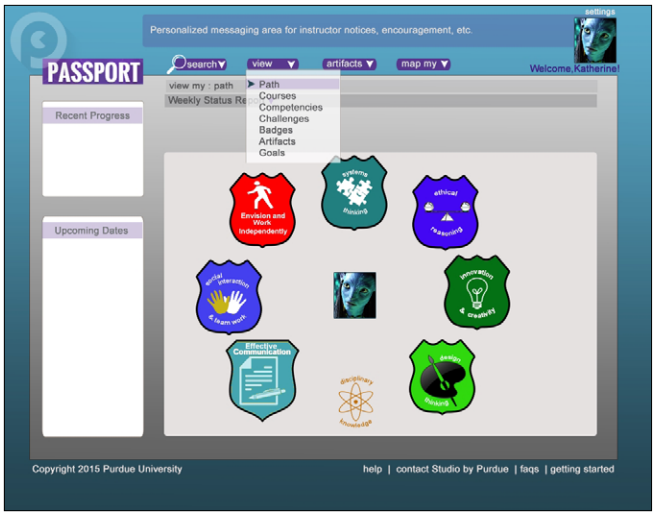
Expectations regarding development work

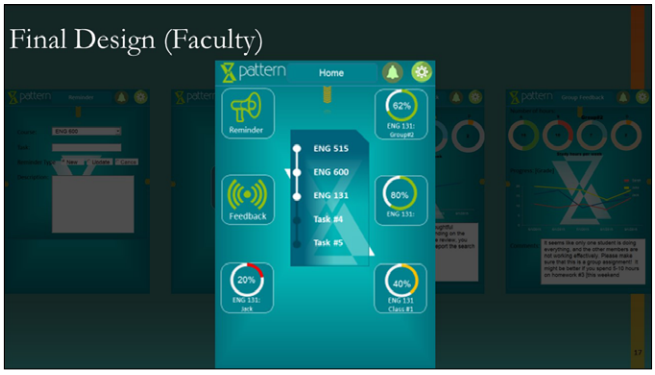
Although the course was advertised as "software design" with "no development skills required," a number of students had goals of actually implementing software during the course period. The students who had development experience expected to do development work and believed there would be time during the semester to complete it. Those who did not had expressed anxiety about their lack of knowledge and experience, something I reassured students about several times early in the semester, and this probably led to at least one student dropping the course early.

The students in the Japanese Language group intended to continue their work after the semester ended. They had wanted to further develop their project, which was computationally complex. However, as far as I am aware, the team members had not progressed on their project because of other priorities, including the emphasis on research for the PhD students.

Need for Project Management

Many student comments related to a need for project management within their teams, a topic that was not explicitly covered in the course. One student indicated that this was "honestly the biggest learning experience." I had requested





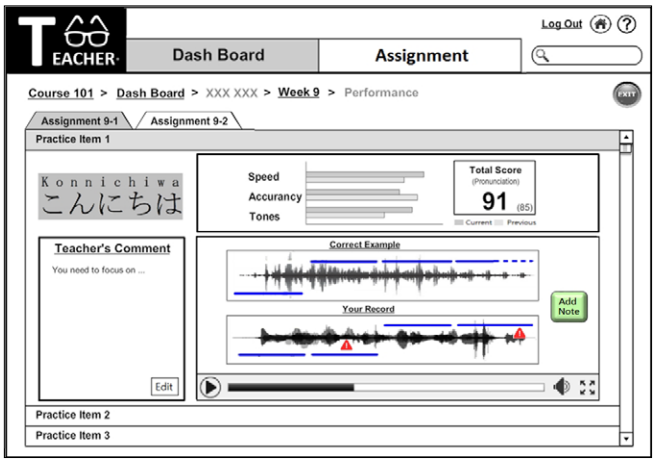


FIGURE 13. Final Prototypes (top to bottom: badge system, time management assistant, and spoken Japanese feedback & assessment).

that this team-member take on some of the organizational role because of difference in work styles and interpersonal friction between other group members. As one member of the same team reflected,

> *[We had] too many chiefs and not enough Indians—I tried to step back from that role purposefully. [One team mate] was doing a good job of pushing for deadlines. I think [he] and I have done a good job working together, but at the same time, we have really butted heads. Times when I wanted to finish working on wireframes and [he] was saying no, we have to get this report done. I said if we get the wireframes done we can write the report…*

In other teams, members indicated that they were not as comfortable with one member taking on a primarily management role. One student suggested:

> *We need a shared schedule document. Everyone can edit what they did and what they will do and their progress on it. Therefore, team members can clearly know the progress of project and who needs assist[ance], and avoid repeating…the same work.*

Being a member of a Cross-Disciplinary Team

Several students specifically called out the value of participating in cross-disciplinary teams including members with expertise in several domains.

> *I'm glad the class was setup with a cross-disciplinary team. Our Japanese Language Software Team consisted of individuals with varying expertise which was Key to the software design process. The expertise on the team consisted of expert Japanese Language instructor, Software Programmer, Visual Designer, and Instructional Designer. Without these contributing experts' there would have been a void on the team.*

Others explained that it also could be a struggle to work with team members from such different backgrounds. As one student explained,

> *I have worked on many 'cross-disciplinary' teams, but they have had a couple of core caveats: (1) predominantly engineering team members (and maybe a random non-engineer) or (2) some form of linear structure (3) overlapping expertise). This was the most 'diverse' team I have worked on in terms of life experience and shared experiences to build from.*
>
> *The situation was new to me in the level to which we spoke a different language. Our prior experiences fundamentally informed out views, biases, and prior knowledge. [One team member] spoke the language and knew the process…he knew software backwards and forwards. [Another] knows Purdue, and the players, and the broader picture. I know some of the design and motivation space. The big learning was vocabulary. I would use a word one way, [the second team member] would have no clue what I was talking about (not his fault) and [the first] would think I was saying something different.*

Interestingly, a very similar sentiment was described by a student who had recently begun a Master's program and expressed her concerns about her lack of general design

experience early in the semester. By the end of the semester, she reflected:

> *It's one of the great paradoxes of life: it's the best of times and it can, in some situations, be the worst of times. There were moments where our team was absolutely phenomenal, productive, and engaged in the design process. It was at this point we made the most progress in the design of our prototype. There were other times when differences of discipline created philosophical views or gaps in understanding/knowledge that made creating a synthesized project difficult.*

When reflecting on her own personal performance, she explained:

> *It terms of what I personally would do differently: (1) Make sure to have relevant and insightful contributions to each team meeting. (2) Work to understand the functional requirements, wire framing, etc. elements of the process better and not let more experienced team members create the prototype but do it together. (3) Manage differences of opinion more effectively and ensure all team members feel heard and understood.*

Each of the groups used a divide-and-conquer approach at times, partitioning work based on their individual skills. While I discouraged dividing up work on major aspects of the design, because I wanted everyone to experience each phase in the design process, one group in particular persisted in what they felt was an efficient and satisfactory model.

> *Although we initially attempted to learn collaboratively the multiple elements, we soon fell into the habit of splitting work according to the abilities and strengths of the team. Having group members that could function in the needed roles was useful. For me, in particular, it was useful to have someone who could rapidly create graphics (user interfaces). Having a subject matter expert with connections to potential users for data collection and interface testing was also helpful. For us, it worked reasonably well this term. Had there been substantial skills gaps or an imbalance causing some to do much more than others, there might have been problems.*

However, another member of this team described a misunderstanding, which persisted across for a large portion of the semester.

> *…One time that major argument happened. 2 of our group members had the different understanding from the other two and it was crucial for our designing process. 2 members already had the final image of the application and they thought they explained to the other two, however, about a month later, it turned out they had been misunderstanding. Therefore, I learned I have to be more careful when I explain something common in my field to others who are not familiar with the topic.*

Personal Responsibility

Interestingly, students did not place blame on the course, or me as the instructor, for team or client issues. Students generally indicated that they felt these were part of the real-world nature of the course and, in the case of team issues, were under their own control. One student explained, "as in any group-based work, there were some issues that affected the quality of the work. For example, it seems that management and professionalism were important skills that we should have paid enough attention to…"

When asked if there was anything I, the instructor, could have done to prevent or assist with these issues, students generally indicated that there was nothing I could or should have done differently. One student explained, "[Like] in the real world, we [referring to himself and fellow instructors in a technology program] tell students you will be in a group project with people you don't see eye to eye with and you have to suck it up and deal with it."

Critique

Although students appeared appreciative of critique provided by peers, a guest faculty member, and the instructor and incorporated much of the feedback received in some way into their projects across the semester, they did not frequently call this out in the end of semester reflections, debrief, or end-of-semester survey. The feelings about critique received from clients across the semester varied. One group felt they needed to balance receiving feedback from clients and what they saw as a need to inform clients about the benefits of this analysis and design process. Two groups were frustrated by client contacts changing several times. Although they appeared to receive useful feedback from each, difficulties setting meetings or even determining who to set them with were added frustrations. The group with a student-initiated project had less difficulty scheduling and generally found the "client" in agreement with their designs—possibly due to prior mutual agreement on the core problem and potential solutions between this client and the team member with subject matter expertise. The key area of discussion between these group members and the client related to the degree to which the product being designed for the class would be directly tied to their immediate need and tools already in use at the university or made more generally viable as a stand-alone tool. Ultimately the decision to work on a stand-alone product was made for pragmatic reasons after several consultations with me.

Workload and use of Class Time

Students' reported time spent on this course ranged from 10-15 hours per week, to one student who reported 30-40

CRITERION	SA	A	N	D	SD
The instructor has met my expectations	3	2	-	-	-
The team has met my expectations	2	3	-	-	-
The team has been respectful of my time	5	-	-	-	-
Team members have listened carefully to what I have to say	5	-	-	-	-
Team members have seemed to understand who the user/learner group is and what their needs are	3	2	-	-	-
The team has made modifications to their design based on my feedback*	1	2	-	-	-
The team has come up with novel approaches to the problem	1	3	1	-	-
The team appeared to have internal disagreements that were reflected in our meetings or other communication*	-	1	-	1	1
Team members all appear engaged with the project (even if some speak more than others)	1	4	-	-	-

TABLE 2. Client feedback survey responses.

Scale: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree
**2 responses included no answer to the items, as the individual (who responded to two teams) only interacted with the teams at the initial project introduction and the final report*

hours per week. I had anticipated that students would spend around 12-15 hours/week most weeks, which aligns with the common expectation of 3 hours outside of class plus one hour inside class for every credit hour (that is, around 9 outside hours and 3 in class hours each week). Although 12 hours/week may be high as an ongoing average for most classes, I felt it was reasonable for a class advertised as being project-oriented.

When asked about their workload during debrief sessions, most students said they worked much more than they had anticipated this semester, and several said that this course required more time than any other course they had experienced during their time in graduate school at Purdue. However, others indicated it was just about the right amount of time. Only one student called out this aspect of the course within the anonymous end-of-course feedback survey, indicating that the workload was "too much for the timeframe." In contrast, one student explained during the debrief that he specifically did not mind the workload, because of the level of feedback received at each phase in the process, compared to the amount of written material produced by the group.

Other Feedback & Suggestions

Positive Responses

Anonymous, end-of-course student survey results showed that students were generally positive about the course and the instructor. Within open-ended comments, several individuals indicated that they appreciated the opportunity to work with real clients, others would have preferred to work on their own projects or a less open-ended project I created just for this class. They indicated they especially appreciated the special guests. During the debrief as well as in the survey, individuals mentioned that they liked the readings and had stored them away for later use, and two indicated that they had already used what they had learned in another course.

Areas for Improvement

Some individuals indicated they would have preferred working on their own ideas, or working on a new project rather than augmenting or refining an existing project. Others stressed that as the instructor, I should screen the projects better, to ensure that projects are "do-able" and interesting to the students in the class. Individuals commenting on the group experience recommended my allowing them to pick their own teams and projects and providing a "plan B" for groups that "really [go] sour in the first few weeks." Individuals made a variety of requests, including incorporating material related to the business side of software development; ensuring that student-facilitated topical presentations are timed to match with course activities; allowing students to work on multiple projects or "consult" with other teams on areas of their expertise; and creating a second class for those who wish to further develop their projects.

CLIENT FEEDBACK

I sent a questionnaire to clients at the end of the semester. Two clients provided feedback for each of the badge system and time management projects, while one client responded to the Japanese language project. Clients indicated they spent 3-4 hours during the semester interacting with students and were satisfied with their overall interactions with myself and the students, as shown in Table 2.

The clients' open-ended feedback was very positive. When asked whether the students' designs met their original goals and expectations, the client representatives expressed enthusiasm about the teams' analysis report, citing specific findings that they found interesting or surprising. For example, "Yes! Their findings were valuable about how students perceive privacy" and "One valuable piece of feedback was that we may need to consider different views of traditional vs competency-based degrees instead of a one-size-fits-all approach." They also mentioned the design, although this seemed less important to them ("I think the design was very well developed. I am interested in the actual outcome of this design.").

Interestingly, clients seemed most impressed with the group that indicated they struggled the most with how to communicate their concerns with the client. For example, one stated, "These guys were a blast to work with. Very passionate about what they do, and very eager to share." Several examples followed about the value and relevance of the students' analysis. The same group of clients reviewed another team, indicating that although the final design was rather "simplistic," they thought the investigation done by the team was very useful.

> *This team attempted to solve a VERY challenging problem. They may have gotten into the weeds with redesigning [the] interface which was not part of the original specifications. Once they started working with [one of the programs that used their software] to wrap their minds around the specific issues facing a program, they began to obtain more valuable feedback. They did a good job with the time they were given and we are very grateful for their time spent!*

These quotes illustrate a difference between the clients' expectations and my own, which I will discuss in the following section.

INSTRUCTOR'S REFLECTIONS ON THE COURSE DESIGN AND IMPEMENTATION

Now that I have had time to reflect since the course ended and re-review my in-class observation notes, artifacts of the students' analysis, and design process, as well as student and client reflections, a few key areas stood out.

Expectation Mismatch

One theme, apparent from the beginning of the course, was a mismatch between my students' and the clients' expectations. This was despite what I had thought was clear goal-setting on my part when I met with each client before the semester began. Although the students themselves were expected to set goals with clients, my intention was to limit the scope appropriately and ensure that tasks were in line with what could be accomplished in this course before the semester began. Despite these meetings, I was surprised at

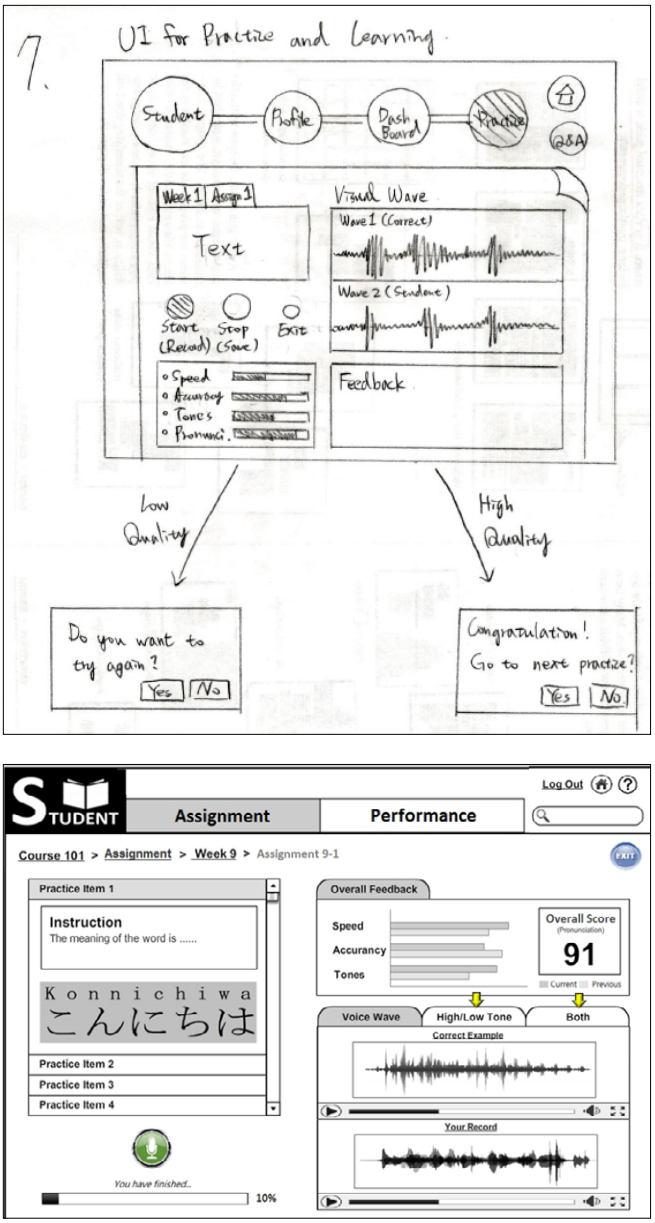


FIGURE 14. Relatively few changes between early design (top) and final prototype (bottom).

what clients presented when they visited the class to pitch their projects to students. The main discrepancy arose from my belief that clients would have a specific design 'problem' to solve or a new feature they wanted my teams to design. This turned out to not be the case for all three projects.

The student-initiated, oral Japanese feedback & assessment project was proposed by two of the four students that formed the project team. One of these students was a subject-matter expert (a Japanese language instructor), and the other had taken her course and was interested in the computational problems that would be involved in creating an automated feedback system. These students had a clear vision for what they thought was needed, and the team's

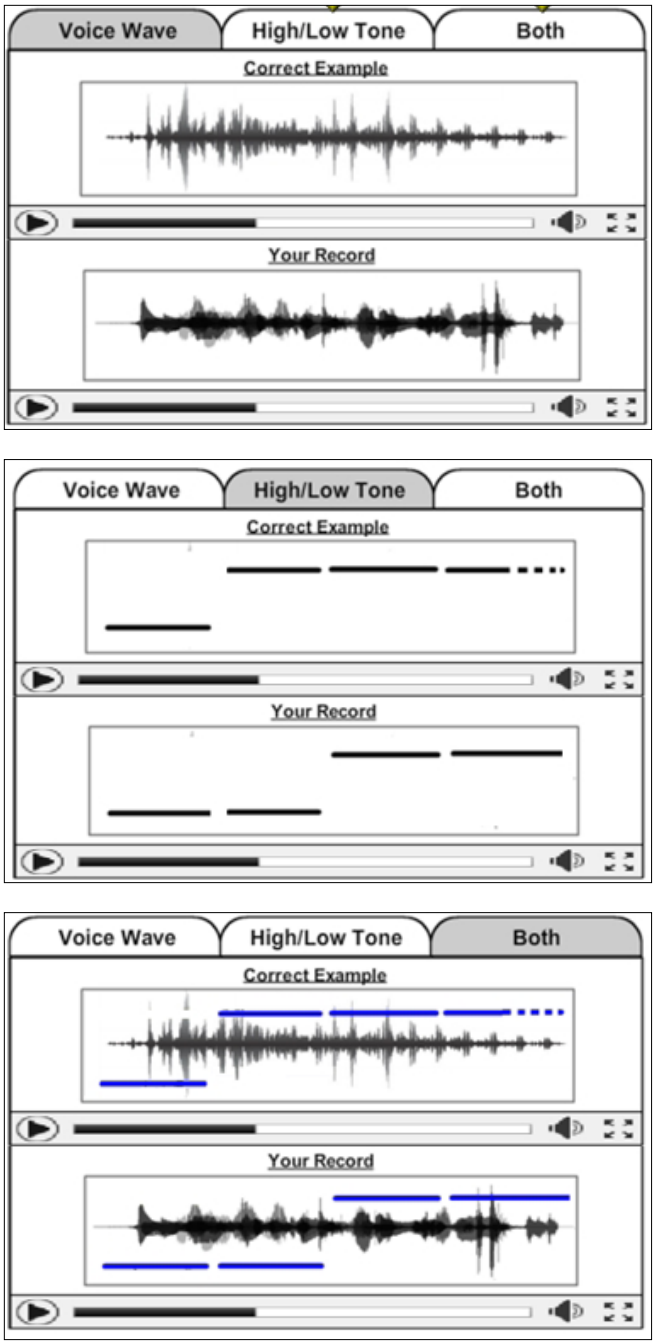


FIGURE 15. Differences in the details: Representation of sound allowing students to self-correct: Waveforms (top), Up-Down (middle), and Both (bottom). The team designed a survey instrument to get feedback from students currently enrolled in a Japanese language class.

design work was clearly heavily impacted by this initial vision. I requested that these students locate someone to serve as a 'client,' one who was not a member of the team. The identified individual was someone who was also enthusiastic about the proposed idea and saw the same project needs as the two team members had proposed. Therefore, the students in this group did not experience the need to integrate different visions and priorities of individuals outside of their own design team, as would be typical when working within a larger organization or when developing software for internal or external clients. Despite the encouragement to "think outside the box" and come up with a variety of designs, this group's final concepts did not diverge as much as I had anticipated (see Figure 14). However, some of the most important changes made were based on input from potential learners and instructors. For example, as part of a survey sent to current Japanese language students, the team mocked up three potential options and determined which was best understood (see Figure 15).

In the case of the time management project, initiated by the university's instructional technology development organization, the clients' initial presentation to the students focused on several open-ended questions they would like to have answered about students' motivation to use their tool and how the tool could be used to motivate students to develop better study skills. Thus far, the university had seen very low use of the tool, even when instructors offered extra credit to students who used it. In essence, the project description amounted to a request that my students conduct a learner analysis for an existing product. Once my students began looking at the existing software and conducted their analysis, they began questioning whether the software met the needs of the target learners. As one explained,

> *The most problematic event that came up for our group … was coming to the realization, through our learner analysis, that no matter the design, students are still not interested in tracking their academic behavior and productivity—even with major incentives. Therefore, it was a little discouraging for our group to come up with a great design knowing all along that the learners, and even ourselves, would never use such an application.*

One of the key problems this group needed to tackle was the seeming disconnect between what they were learning about student motivation and the implicit assumptions that appeared to be made by the existing software design. They struggled with *how*, and *how much* to tell the client. Some team members feared that such information would not be well received.

During the semester, the students and I negotiated with the clients on deliverables. I told the students that they should meet clients' needs as closely as they could, and so that I could be somewhat flexible about deliverables and due-dates; however, I also wanted them to gain experience designing and developing a low-fidelity prototype. Therefore, they needed to complete all course deliverables. Students felt that this did not align with the clients' strong emphasis on the value of the analysis document. While the clients seemed to find the prototype systems my students designed interesting, they did not have a need for someone to develop a prototype.

> *By the end of the semester, I saw the prototype as more of an artifact to communicate the learner analysis in a concrete way rather than a fully fleshed out proposal of how we thought the system should work. It was intended to highlight differences in underlying assumptions rather than demonstrate functionality*

Although our agreement was that the organization could 'take inspiration from our designs,' it was unclear how, or if, the prototype itself was used or adapted by the IT group. This may have disappointed some students who were excited about the opportunity to design something that would actually be used by learners.

Student Workload

As discussed earlier, students indicated that the workload for this class was higher than that of other classes, and some felt that it was unreasonable. While I did anticipate that they would spend a fair amount of time outside of class with course readings, individual work, and team meetings, I also had planned to offset some of the workload by offering in-class studio time for students to work in their groups. However, more time than I had anticipated was taken up by student presentations, activities, and, especially, follow-up questions and discussions. Although students largely found these experiences useful, the result was much less "studio" time then I had planned on. This meant not only less time for me to observe the teams and offer critiques, but also more time had to be set aside outside of class.

In addition, some students reported spending more time outside of class than I had anticipated, and I sensed (although I had no direct evidence other than from participation in class discussions) that many students did not complete all or even any of the readings during the latter part of the semester.

I have continued to try to address these issues in subsequent years. In the next year, I held more strictly to my own guidelines on the duration of student presentations. In the following year I experimented with replacing in-class presentations with online discussions. I also continue to review readings, searching for shorter and more practitioner-oriented readings, as well as reviewing websites and videos, to replace densely written texts. Finally, I broke up deliverables into smaller chunks and realigned expectations. For example, instead of requiring UML, I gave students the option to use one from among many types of visualizations of their software, including flow-charts and site-maps. I also expected each student to write one set of requirements rather than cover all aspects of their design.

Student Growth

My intention for this course was to use a personal growth model rather than normative assessments or mastery learning approaches. Based on the end-of-course reflections, each student learned from this course, and the majority indicated that they grew in areas that aligned well with my goals for the course, as well as some areas that I had not intended (this is discussed further in Exter et al., 2018). From my perspective, each student grew in some way, although some appeared to gain more from the course than others. For example, it became obvious to me that some students had read the materials and applied them in their course work and in-class discussions but for other students this was less obvious due to the largely collaborative nature of the course.

Team and Individual Effort

I realize now that I was not fully prepared for some aspects of the facilitation of the teamwork process. The most obvious issues were between individual team members. Students themselves indicated to me that there was nothing more I could, or should, have done to address interpersonal issues between them, stating "we are all adults". However, there was one aspect that students did not appear to be concerned about, but was most important to me: The team that was most satisfied with its collaborative efforts indicated that they primarily used a divide-and-conquer approach. While this allowed teams to build on their respective strengths, it did not encourage students to learn from each other (at least, not in much depth) or encourage everyone to gain at least some experience in each of the areas I had hoped they would. I recognize that my rather vague instructions to "collaborate and make everyone involved in each step of the process, but also lean on team members' strengths" did not give sufficient guidance. I am still exploring ways to achieve, or even articulate, the balance that I had intended, which would involve by leveraging students' own strengths while still engaging all team members in all parts of the process.

The differentiation in grading between individual team members that I had planned for did not occur because I could not be certain how much work the various team members had done. This was despite my informal attempts to determine what each team member was doing each week, and the use of a self-and-peer feedback tool provided little differentiation among students overall. Therefore, everyone got full points for "individual effort." Since that semester, I have read more about team- and group-work, but I am still searching for guidance and, ideally, a tool that best suits a fully collaborative work model.

In the third iteration of the course design, I experimented with a "flipped" classroom model, in which students engaged in online conversations each week. The guidelines for students' initial discussion posts are designed to allow each individual to practice the skills they are learning (e.g., asking students to sketch multiple design ideas or create a process model their team might use). Guidelines for the

follow-up discussions aim to give students opportunities to critique one-another's work and engage more deeply in each activity. Students are required to integrate what they have learned from course readings in both initial responses/designed artifacts, and ongoing discussions. I believed this would encourage students to be more consistent about reading throughout the semester, as well as foster more substantive discussion and deeper learning. It also ensured that every student exercised the desired skills at least once prior to working as a team or relying on the student with the most expertise to create a polished product.

Critique & Feedback

For this studio course, I planned for critiques to play a major role and designed opportunities for students to receive formal and informal critiques from peers, clients, guest experts, and me. I was pleased at how this turned out, and I believe some credit goes to the scaffold used to enforce reflection and planning based on each formal review by peers (scheduled rotation of students to provide feedback on one another's work), clients (through client meetings), and me (written feedback on deliverables), as well as feedback received from prospective users during user testing. Compelling students to reflect on each critique and determine how to apply it (if useful) appeared to help students become more thoughtful about applying changes. This activity indicated a comprehension of underlying design tensions, or areas for improvement, in their design work. In retrospect, I was fortunate that students were able to provide well thought out and relevant feedback and recommendations to one another. I believe this was largely due to the prior experience of each student in the class. This may be something I will need to scaffold more in the future, based on my more recent experiences using a similar approach with groups that include members with no prior design experience.

Students received a large quantity of written feedback from me on all deliverables, especially on the analysis and design documents. One student continues to tease that I provided more written feedback than the amount of text contained in the original deliverable. For the most part, the feedback seemed to be utilized, although in some cases students appeared to be making changes because I told them to rather than students' own perception of the value of the feedback to inform the later design.

In subsequent semesters, I have incorporated additional rounds of formal critiques, which have been well received and used by students. This is paired with a move towards a more purposefully iterative design model, incorporating peer, instructor, and client feedback for every 3-week round. This model also provides more focus on continuous review of problem framing and scope, which is guided by what is learned from critiques, as well as a collection of primary and secondary information from, and about, learners.

Grading Considerations

Some students expressed concern about earning grades and points throughout the semester and also appeared to be especially concerned that the quantity of written feedback I provided on drafts (which did not receive grades) indicated that they would receive a low grade for the course. I assured them I was much more concerned with their own learning, and that I preferred they experience all of the ups and downs of a realistic project experience and submit materials that reflected this rather than try to produce something that appeared 'perfect.'

In subsequent semesters, I have continued to give feedback on an initial draft (generally incomplete) and grade a second (complete) draft, but I also allow students to resubmit their projects. This encouraged them to think through and apply feedback intended to help them grow in their understanding and use of the various techniques and types of documentation. This might be a constraining factor in the degree to which this course could scale up, but this is not a concern for me at this time, especially since feedback is given at the group level and even a larger class would likely not exceed five team projects.

Tie-in with Prior Experience

Although some students were very experienced in one of the disciplinary areas covered, none had professional experience specific to educational software design. This was not an issue, as I had intended to bring together students from different disciplines and give them a new experience. Students brought their own areas of expertise to the class, in topical discussions and within their working teams.

However, I was surprised to discover that students' related disciplinary experience did not prepare them fully for specific elements of the course. For example, several students had prior programming experience as part of their role as engineers, statisticians, or game designers. However, even these students did not show a complete understanding of UML or how to write requirements for a software project. I realized that I would need to spend more time understanding the backgrounds students bring to the course rather than assuming that their disciplines are "similar."

I expected that students who had a professional background working with clients on complex projects within one or more design disciplines would be more prepared for this aspect of the course than others. Although this was true in many ways, it seemed that students needed to consciously draw on their experience from their own disciplinary work to deal with clients, as well as deal with very ill-structured problems introduced in this course. Interestingly, in their final reflections and end-of-course evaluations, several students noted that although working with clients could be

frustrating, they recognized that this is a typical aspect of design work and therefore a valuable experience.

In subsequent years, I have been surprised at the impact of the make-up of the student body enrolled in my course. Each year, students vary not only in the areas of expertise, but also the specific knowledge gaps they bring. Working with these additional groups of students has made me realize that I have unwittingly made assumptions about their prior knowledge when selecting and creating resources for the course. For example, in the second year of my course, no LDT students enrolled. I rather naively assumed that Curriculum & Instruction students would be "close enough" and able to play a similar role to LDT students within their teams. However, I have found that these students were not necessarily prepared to conduct learner and needs analysis, or plan formative and summative evaluations of the learning aspects of the software. Therefore, I have created additional scaffolding and activities around these areas.

CONCLUSIONS

Students were generally positive about the course and indicated that they had met their own learning goals, as well as having learned things did not expect to when they enrolled(discussed in our co-authored publication about goals and expectations for this course (Exter et al., 2018)). The students who expressed the most enthusiasm also had the most recommendations for future improvement. In my mind, the biggest area of success of the pilot semester of was that it provided a lot of data for me, allowing me to continue to improve the course and find new ways to help prepare students to work in multi-disciplinary teams and meet the needs of real clients and users.

ACKNOWLEDGMENTS

Thanks to Dr. Micah Modell and Dr. Colin Gray for their feedback on the course design, Dr. Colin Gray and Iryna Ashby for their substantive feedback on this paper, and the students willing to be guinee pigs in the first year of this course offering.

REFERENCES

Cox, S., & Osguthorpe, R. (2003). How do Instructional Design professionals spend their time? *Techtrends*, *47*(3), 29,45-47. https://doi.org/10.1007/BF02763476

Dieste, O., Juristo, N., & Shull, F. (2008). Understanding the customer: What do we know about requirements elicitation? *IEEE Software*, *25*(2), 11–13. https://doi.org/10.1109/MS.2008.53

Exter, M. (2012). *The educational experiences of software designers working in education/instructional technology related fields . Instructional Systems Technology*. Unpublished doctoral dissertation, Indiana University, Bloomington, Indiana.

Exter, M. (2014). Comparing educational experiences and on-the-job needs of educational software designers. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education - SIGCSE '14*, 355–360. https://doi.org/10.1145/2538862.2538970

Exter, M., Alshammari, A., Fernandez, T., Anthony Randolph Katherine Chartier, Kuo, Y.-T., Lancette, S., & Nemelka, B. (2018). Empowered guinea pigs: Stories of cross-disciplinary projects in an experimental educational software design course. In B. Hokanson, G. Clinton, & K. Kaminski (Eds.), *Educational Technology and Narrative* (pp. 165–175). Springer. https://doi.org/https://doi.org/10.1007/978-3-319-69914-1_14

Exter, M., Caskurlu, S., & Fernandez, T. (in press). Comparing computing professionals' perceptions of importance of skills and knowledge on the job and coverage in undergraduate experiences. *Transactions on Computing Education*.

Exter, M., & Turnage, N. H. (2012). Exploring experieinced professionals' reflections on computing education. *Transactions on Computing Education*, *12*(3). https://doi.org/10.1145/2275597.2275601

Gray, C. M. (2014). Evolution of Design Competence in UX Practice. In *CHI'14: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems* (pp. 1645–1654). New York, NY, USA: ACM. https://doi.org/10.1145/2556288.2557264

Gray, C. M., Dagli, C., Demiral-Uzan, M., Ergulec, F., Tan, V., Altuwaijri, A. A., … Boling, E. (2015). Judgment and Instructional Design: How ID Practitioners Work In Practice. *Performance Improvement Quarterly*, *28*(3), 25–49. https://doi.org/10.1002/piq.21198

Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software*, *18*(4), 58–66. https://doi.org/10.1109/MS.2001.936219

Hooper, S., Rook, M. M., & Choi, K. (2015). Reconsidering the design of a learning design studio. In B. Hokanson, G. Clinton, & M. Tracey (Eds.), *The design of learning experience: Creating the future of educational technology* (pp. 63–76). New York, NY: Springer. https://doi.org/10.1007/978-3-319-16504-2_5

Koszalka, T. A., Russ-Eft, D. F., Reiser, R., Canela, F. A. S., Grabowski, B. L., & Wallington, C. J. (2013). *Instructional Designer Competencies: The Standards* (Fourth Edi). Charlotte, NC: Information Age Publishing.

Modell, M. G. (2013). Iterating Alone over a Method and Tool to Facilitate Equitable Assessment of Group Work. *International Journal of Designs for Learning*, *4*(1), 39-53. https://doi.org/10.14434/ijdl.v4i1.3283

Niknafs, A., & Berry, D. (2012). The impact of domain knowledge on the effectiveness of requirements engineering activities. In *Requirements Engineering Conference (RE), 20th IEEE International* (pp. 181–190). IEEE Press.

Niknafs, A., & Berry, D. (2017). The impact of domain knowledge on the effectiveness of requirements engineering activities. *Empirical Software Engineering*, *22*(1), 80–133. https://doi.org/10.1007/s10664-015-9416-2

Ritzhaupt, A., & Martin, F. (2014). Development and validation of the educational technologist multimedia competency survey. *Educational Technology Research and Development*, *62*, 13–33. https://doi.org/10.1007/s11423-013-9325-2

Ritzhaupt, A., Martin, F., & Daniels, K. (2010). Multimedia competencies for an Educational Technologist : A survey of professionals and job announcement analysis. *Journal of Educational Multimedia and Hypermedia*, *19*(4), 421–449.

Smith, K. M. (2008). *Meanings of "design" in instrucitonal technology: A conceptual analysis based on the field's foundational literature. Instructional Systems Technology*. Unpublished doctoral dissertation, Indiana University, Bloomington, Indiana.

APPENDIX A

Key Sections of Syllabus

PURDUE UNIVERSITY | EDCI 62700 Syllabus
Educational Software Design

Course description

In this hands-on course, you will work with students from across disciplines to design educational software to be used in K-12 or higher education. As part of the course, we will explore the processes and languages used by specialists in different disciplines to design and develop software. Then, we will go through the design process ourselves as we design a new piece of educational software or a significant new feature set for an existing software package currently being used in educational settings. Since the course is cross-disciplinary, each student will bring a different skill-set.

Course objectives

Instructor's Goals

After completing this course, learners will be able to:

1. Demonstrate understanding of language, design processes, and techniques used by other fields, including instructional design, software engineering, graphics design, and quality assurance.
2. Collaborate with peers as part of a diverse design team.
3. Work through a systematic design process while designing software for a real client and users. This will include:
    - Communicate effectively with clients and end-users (k-12 or higher education students).
    - Gather requirements from a client.
    - Create a design document that can be used to communicate with other team members and stake-holders.
    - Create a prototype of an educational software package or feature set.
    - Conduct a usability test.
    - Utilize usability test results to plan for future modifications to the software design.

Learner's Goals

In addition to meeting the instructor's goals provided above, each student will set his or her individual goals.

1.

2.

3.

Course Text

No textbook will be used for this course. Readings will be drawn from a variety of sources. Many of the readings will be selected by students.

Assignments

You will have a number of individual activities to complete throughout the semester, as well as participating in a large group project. You will also be serving to evaluate and

PURDUE UNIVERSITY | EDCI 62700 Syllabus
Educational Software Design

provide feedback on group members' work throughout the semester. In-class time will be provided for many of these activities. However, both individual and group work will be done out-of-class as well. In some cases, we may begin an activity in-class and complete it outside of class.

Individual Activities:

- *Plagiarism Certificate:* During the first week of the course, complete the Plagiarism tutorial and submit a copy of the report. No assignment will be graded without submission of this item. Review the resources on http://www.education.purdue.edu/discovery/research_integrity.html and https://www.indiana.edu/~istd/ . Then, take the certification test at https://www.indiana.edu/~istd/test.html . Submit your certificate in the **Assignments** area in Blackboard.

- *Presentations:* You will lead part of a class session relating to one of the special topics covered in this course. You may wish to incorporate interactive activities into your presentation. Prior to the presentation, you will provide one or more readings for your peers to complete. Presentation topics will include the following – but if you have an idea for a topic not on the list, discuss with the instructor!
    - Instructional design models (ADDIE and alternative models)
    - Software Engineering design process models (You might include traditional processes such as waterfall and spiral model, as well as alternative processes, such as RAD, Agile, SCRUM, extreme programing, etc.)
    - Learner & Needs Assessment
    - Gathering/Eliciting requirements
    - Visual design languages: UML, Use Cases
    - Rapid Prototyping & Design Iteration
    - User Interface Design
    - Graphic Design
    - Usability Considerations
    - Usability testing (overall purpose and specific types of testing)
    - Types of Software quality testing (overall purpose and specific types of testing)
    - Formative & Summative Evaluation – Does the instructional product meet its goals?
- *In-Class Activities:* Individual or pair activities will allow you to practice skills related to the topic of the week. Specific topics are listed in the class schedule.

Group Project

You will work as part of a group of 3-5 peers to design a piece of educational software or a significant new feature set for an existing educational software package. By the end of the semester, you will have created a testable prototype and conducted a usability test on it. Along the way, your group will create the following artifacts/activities:

- *Design document:* This PDF document will include:

PURDUE UNIVERSITY | EDCI 62700 Syllabus
Educational Software Design

    - Results of your learner and needs analysis: description of gap or need you are filling, users, and context. This should include some references to research on the users, context, and/or learning or other objectives for the software (for example, this could include academic research on adult learning theory; motivation theory; gamification. It could also include statistics on the break-down of students at Purdue, data provided to you by the client on previous requests or typical usage patterns, etc.)
    - Requirements Specification document and Use Cases: use the template provided
    - Initial interface design: Design for all screens you believe will be included for the software/feature you are designing. The screen designs may be hand-drawn or electronic (you may use wire-frame software, visual design software, PowerPoint – whatever allows you to generate images that convey your current design ideas).
- *Client feedback report:* You will get feedback from your client at least three times. This report will contain the feedback you receive, and your plans for modification based on that feedback.
- *Instructor and peer feedback report:* You will get feedback from your instructor and peers twice during the semester. This report will contain the feedback you receive, your reflection on that feedback, and your plans for modification based on the feedback.
- *Prototype:* This will be the prototype you will use for your pilot user testing. Depending on the needs of the project and the skill-set in your team, your prototype may be in many forms, including (but not limited to): a paper prototype; to an interactive PowerPoint document; an HTML mockup; or a partially functional application.
- *Usability Test Plan:* This will include a script which will describe in detail what users will do as part of the usability test, and supporting materials.
- *Final Project Report:* This documents your work from the entire semester, and will include:
    - Executive summary, including original purpose, summary of key steps in your process, the current state of your prototype, and future plans
    - Design document (as an appendix)
    - Documentation of your process – show photos, screenshots, and other images or text that documents the progress of your development over time.
    - List of changes made to design since design document was created (and rationale for changes)
    - Key screenshots of prototype, and link to the prototype or include all pages/slides in an appendix
    - Usability test result report (including the test plan, the test results, and your interpretation of the results)
    - Final feedback from client
    - Plans for future modifications based on usability test results and client feedback

PURDUE UNIVERSITY | EDCI 62700 Syllabus
Educational Software Design

    - Implementation and evaluation plan (can use what was created as part of in-class activity)
- *Project Presentation:* Your group will present on the context and users, the software you designed, usability test results, plans for future development, and lessons learned. All team members must participate in presenting the project.
- *Individual Reflection:* Each team member will separately reflect on the design process and group experiences.

During the semester, group members will be asked to participate in **Self and Peer Evaluations** each week once group work has begun**.** This is designed to make students more aware of the group process, and their role and performance within the group.

ASSIGNMENT	POINTS POSSIBLE
Plagiarism Certificate	N/A
Individual Activities	
Presentation	20
Participation (active participation in class; weekly self-and-peer feedback; active participation in group activities)	20
Group Project	
Design Document	60
Client feedback report 1	5
Client feedback report 2	5
Instructor & peer feedback report 1	5
Instructor & peer feedback report 2	5
Prototype	20
Usability test report	20
Final Report	55
Project Presentation	10
Individual Contribution (based on level of participation, attitude towards the team, and creative input in group project)	20
Individual Reflection	5
TOTAL POINTS	**250 points**

APPENDIX B

Initial Course Schedule

PURDUE UNIVERSITY | EDCI 62700 Syllabus
Educational Software Design

The schedule below was the timeline as initially envisioned. It was adjusted throughout the semester based on student progress and access to clients and users.

Dates	Activities	Due
Week 1 (Aug 27)	Introductions Interests & Personal Goals **Guest Presentation:** Intellectual Property concerns at Purdue **Client Presentations:** Introduction of potential topics **Discussion/Q&A:** Discuss potential projects and IP considerations	• Complete project preferences survey
Week 2 (Sep 3)	**Discussion:** What are design models? **Student Presentation:** Instructional design model (ADDIE and alternative models) **Student Presentation:** Learner & Needs Analysis **Activity:** Rapid prototyping (design a wallet) **Project groups assigned** **Group Project Work:** • Begin learner and needs assessment. *You may think you know them, but doing research will give you another perspective!* • Begin considering strategies for getting input from client and potential users.	• Plagiarism certificate
Week 3 (Sep 10)	**Activity:** Interview a partner & determine needs **Student Presentation:** Eliciting requirements **Group Project Work:** • Team charter • Refine questions, prepare to meet with clients • Schedule meeting with clients. • Continue learner and needs assessment	• Post draft of initial interview questions for feedback.
Week 4 (Sep 17)	**Student Presentation:** Software Engineering Design Processes **Activity:** Generate a persona and a scenario (group) **Group Project Work:** • 2 Personas and scenarios • Debrief client responses (if possible) • Determine what other types of information you need to understand requirements and make plans to collect additional data (e.g. through user interviews or observations, use of existing tools)	• *Initial Client Meeting* • Collect additional information (reading, investigating competitor tools, user interviews/observations, etc.)
Week 5 (Sep 24)	**Student Presentation:** Visual design languages: UML, Use cases **Activity:** Create a Use-case diagram **Group Project Work:** • Work on UML **Discussion:** Requirements Documents: what to include?	• Collect additional information (reading, investigating competitor tools, user interviews/observations, etc.)

PURDUE UNIVERSITY | EDCI 62700 Syllabus
Educational Software Design

Week 6 (Oct 1)	**Presentation:** Rapid Prototyping & Iteration **Group Project Work:** • Continue work on requirements & UML • Begin sketching out ideas for the interface, and how users will move between different Use Cases	
Week 7 (Oct 8)	**Student Presentation:** User Interface Design **Activity:** Interface design – Page Layout. **Group Project Work:** • Begin to work on interface design	• Design document (Draft)
Week 8 (Oct 15)	**Student Presentation:** Graphic Design concepts/principles **Guest Presentation:** Colin Gray - Wireframes **Group Project Work:** • Critique by guest presenter • Each group will meet with instructor to discuss design document feedback • Update design document based on peer and instructor feedback • Continue working on interface design • Decide what you will show the client and what to ask the client	• Peer feedback on draft design documents (each student reviews design document of one other team (assigned))
Week 9 (Oct 22)	**Discussion:** Client check-in: how did it go? What did we learn? What didn't we learn? **Activity:** Peer review **Group Project Work:** • Discuss modifications based on client meeting • Make modifications to requirements and interface design based on client meeting • Begin to consider what form the prototype will take/what software you may use to create it	• *Check in with clients*
Week 10 (Oct 29)	**Presentation:** Usability testing **Group Project Work:** • Begin to consider how you will usability test prototype, and what level of fidelity you will require to test it • Begin design of prototype	• Peer & instructor feedback report
Week 11 (Nov 5)	**Activity:** Write a usability test script and test with a peer **Group Project Work:** • Continue work on prototype • Brainstorm how to gain access to participants for usability test	• Client feedback report • Reach out to potential usability test participants
Week 12 (Nov 12)	**Activity:** Write a usability test script for one of your group's features and test with a peer **Activity:** Peer review of prototypes **Group Project Work:** • Continue work on prototype • Begin to write usability test document; Plan for usability test	• Reach out to potential usability test participants • Design document (Complete)

PURDUE UNIVERSITY | EDCI 62700 Syllabus
Educational Software Design

Week 13 (Nov 19)	**Presentation:** Types of software quality testing (including overall models and specific types of testing) **Activity:** Bug testing & Write a bug report **Activity:** In-class usability testing & peer review **Group Project Work:** • Plan for usability test • Peer reviews – visit other tables	• *Client Feedback* • *Conduct Usability test* • Peer & instructor feedback report
Week 14 (Nov 26)	**NO CLASS – THANKSGIVING BREAK Oct 25-28**	• *Conduct Usability Test*
Week 15 (Dec 3)	**Student Presentation:** Formative & Summative Evaluation **Activity:** Implementation & Formative Evaluation Plan (in project groups) **Group Work:** • Debrief from usability testing • Write a plan for further modifications • Write plan (outline) for larger-scale pilot and formative evaluation • Create final project write-up	• Usability test report • Client feedback report
Week 16 (Dec 10)	**Final presentations** **Individual Course Review Meetings**	• Final project report • Individual reflection
Week 17 (Dec 17)	**Individual Course Review Meetings**	

APPENDIX C: SAMPLE DELIVERABLE TEMPLATES

Section of the Analysis Document Template

LEARNER ANALYSIS

Complete the table, and then fill in details in the sections below. Include at least the headings given. Depending on the nature of your project, you may include additional characteristics. For each area, use data from data source(s) you collected. Indicate how the data was collected and, if appropriate, next steps to collect additional data.

Component	**Data Source** (literature, publicly available information, client interviews, user interviews/surveys/observations, etc)	**Information Learned** (What do you now know about this aspect of learners?)	**Potential Implications for Design** (What does this tell you about what your design needs to do/should not do or assumptions you can or cannot make within it?)
Type(s) of learners			
Prior knowledge/ability			
Motivations			
Learning preferences			
Special Needs			

TYPE(S) OF LEARNERS

Are there one or more learner types? What are their characteristics?

For example “new Freshmen” might be different and have different needs (relevant to this project) than “transfer students” or “graduate students”. Students who have already been using this (or a similar) system may also have different needs from new users.

Other considerations may include, if potentially applicable for your project:

- ESL learners
- Learners’ reading/writing level
- Current computer skills
- Cultural diversity of group
- Level of knowledge about college
- Comfort level with different types of teaching or learning
- Anxiety about the topic/college in general/particular aspects of college/technology
- Self-efficacy
- Perceptions of own technology skills
- Lack of access to technology for some students (Do all students own their own laptop? Data Phone? iPhone (if software is restricted to that)?)

FUNCTIONAL REQUIREMENTS

This section will provide details about all of the functionality you plan to deliver as part of your software tool or feature set.

A ***very simplified*** example is provided in Appendix A. Some notes in ***orange*** point will give you more ideas about what might be done in a real requirements document for a much larger team.

USE CASE MODEL

Use UML to create a use case model, consisting of all relevant **actors** and the **software product** *(rectangle or rectangles)* including key **use cases** *(ovals).* You can use whatever tool you wish to create the UML diagram – but this may be a good option: http://www.gliffy.com/#uml-diagrams .

DETAILED FUNCTIONAL REQUIREMENTS

In this section, you will provide details for each business activity. Please continue to use hierarchical numbering as it will be easier for everyone to keep track of later.

A sub-section will be provided for each ***use case*** (each oval in the use case model)**.** This will include:

1. A brief description of this use case, including what it is for and how it is triggered.
2. A list of detailed requirements which gives sufficient details for someone *other than yourself* to design the interface, program the system, or plan for testing before the software is completed. *NOTE: this part of the document can be shown to clients or users, but many clients and most users will not be able to follow this technical document. They will respond much better to a quick-and-dirty prototype.*

NON-FUNCTIONAL REQUIREMENTS

USABILITY

Describe important usability aspects of the interface for your target users, such as:

- Ease of learning: The system must be easy to learn to use for both users experienced with similar systems, and new users.
- Task efficiency: The system must be efficient for the frequent user.
- Ease of remembering: The system must be easy to remember for the casual user.
- Understandability: The user must be able to understand what the system does.
- Satisfaction: The user must feel satisfied with their experience with the system.

Although these may all seem important, some may be more important than others for your design.

PERFORMANCE (IF APPLICABLE)

List specific requirements related to performance of the system, including response time, number of users that can be handled simultaneously, etc. For purposes of this project, you only need to include this if it is directly important to your design and you believe it may be a concern.

SYSTEM REQUIREMENTS

Indicate requirements for the system that your software will run on. For example, if it is a phone app, will it run only on iPhone or also on Android and Windows Phone? If this is a web-based tool, you may need to have requirements for both the *server* that the system runs on, and the *client* (computer, tablet, phone, etc), including the *web browser* (IE, Firefox, Chrome, Safari).

Client Feedback Report

Overview of Client Meeting

- When and where did you meet?
- What topics were discussed (at a high level)?
- What new things did you learn? How will they impact your design process or your software design going forward?

Plan to Address Client Feedback

After each meeting with your client, you will fill out the following table. To consider as you record the client feedback:

- Each distinct item of feedback from the client should have its own row.
- "Next steps" are what you may do to follow up on the feedback. This could include doing additional research, following up with more questions, considering alternative designs, trying something out with users, etc.
- "Impact on your design" indicates changes you will make (or not make) based on client feedback. This may include:
    - Literally implementing the feedback
    - Making a change that addresses the underlying concern/problem in a different way (describe how and why you choose to do this)
    - Not implementing any changes because of lack of time/resources but adding to a list of potential future work (explain why you decided to do this)
    - Not implementing any changes because you do not feel they are appropriate/relevant to the design (explain why and justify how you are still meeting client's overall needs.)
- "Justification" of your next steps/impact on design. You will need to plan to be able to explain your decisions to clients at next meeting or upon questioning after a final presentation. Therefore, the justification should be clearly articulated.

Client feedback	**Next steps**	**Impact on your design**	**Justification for next steps/impact on design**