International Journal of Web-Based Learning and Teaching Technologies
Volume 16 • Issue 4 • July-August 2021

Learning From Doing:
Lessons Learned From Designing and Developing an Educational Software Within a Heterogeneous Group

Nicole Wang-Trexler, University of Pennsylvania, USA
https://orcid.org/0000-0003-3195-4205

Martin K-C. Yeh, Pennsylvania State University, USA
https://orcid.org/0000-0002-5630-1633

William C. Diehl, Pennsylvania State University, USA
https://orcid.org/0000-0002-2827-5327

Rebecca E. Heiser, Athabasca University, Canada

Andrea Gregg, Pennsylvania State University, USA
https://orcid.org/0000-0002-5588-8145

Ling Tran, ThoughtWorks, Inc, USA

Chenyang Zhu, University of Southampton, UK

ABSTRACT

Software applications in educational technology have been a strong driving force for the success of online learning at all levels. These applications are created for various purposes and are used by a range of experts. The development of a successful educational technology software takes a deliberate team effort and thoughtful project management. This interpretive case study details the processes, successes, and challenges determined throughout the development of an educational web application, the Social Performance Optimization Tool (SPOT). In describing the evolution of SPOT, and the processes the heterogeneous team followed in the development of the web application, this study provides analysis and guidance to educational researchers who are interested in developing educational web applications in the future. The study described how authors mindfully adopted software design models, team management techniques, and communication tools. Additionally, the paper highlights practical and unique implications developers must account for when working in higher education contexts.

KEYWORDS

Educational Applications, Web Application Design and Development

INTRODUCTION

Rethinking learning in the digital age requires educators and educational technology developers to respond to the fundamentally connected learning environment, which has rapidly evolved over the past

DOI: 10.4018/IJWLTT.20210701.oa3

two decades (Lamberton & Stephen, 2016). The so-called Web 2.0 social networking era has provided the education community a tremendous amount of opportunities to leverage networked technology in improving online learning experiences (Berger & Krousgrill, 2012). In an educational environment, students are typically grouped within a closed system such as a Learning Management System (LMS), and the design of the system limits their interactions. Although the LMS is well-adapted within online higher education as a core technology enabling online learning and asynchronous discussions (Legon & Garrett, 2018), they are not necessarily designed to address the social context where people learn.

There are very few web applications in higher education that support the need to increase learner engagement due to certain constraints. On one hand, for software companies that aim at supporting greater learner engagement, their backgrounds are often more established in computer science, and less in the educational field. As a result, educational web applications created by such companies are more visually pleasing and institutive to interact with, yet they fail to address the fundamental problem of supporting the learning process in a social and digital context. On the other hand, although educators are well-equipped with the knowledge and expertise in learning within a socio-cultural setting, they sometimes fail to have the time and resources to develop compelling and sophisticated web applications on their own; and even if they do, the web applications are typically less user friendly.

The purpose of this interpretive case study is to share the processes, successes, and challenges determined throughout the development of an educational web application. In describing the evolution of the Social Performance Optimization Tool (SPOT) web application, and the processes the team followed in the development of the web application, we aim to provide analysis and guidance to educational researchers who are interested in developing educational web applications in the future. In the following sections, we begin by reviewing relevant studies that have completed similar work, followed by an introduction of the Web application we developed. We transition to the methodology section which describes the research question and study design, followed by our findings. Lastly, we conclude with implications and potential future work.

LITERATURE REVIEW

In the 1990s, higher education began to gradually increase its adoption of technology equipment and infrastructure to support teaching and learning. In Kirkwood and Price's (2014) critical literature review, they found that the majority of technologies used to support teaching were to supplement existing teaching practices such as adding course materials in an online space. The majority of learning technologies were used to emphasize operational improvement, quantitative improvement in assessment or engagement, and qualitative changes for in-depth understanding.

In many ways, the educational landscape remains stagnant when it comes to adapting to the evolution of the digital age. Typically, an LMS includes tools for discussion, email, assignment submission, instructional content management, analysis, organization, delivery, and assessment (Cavus, 2007). Although existing tools are successful in supporting instructors in course management, there is a deficiency in supporting authentic social connections and providing the functionality for students to actively seek or provide peer assistance (Dron & Ostashewski, 2015). In order to compensate for these limitations, some instructors seek help from various commercial applications to transform a traditional classroom into an active learning environment (Gao, Zhang, & Franklin, 2013). This is especially problematic for online students who do not participate in discussions. As a result, interactions in an online course often become siloed to instructor-to-student exchanges rather than authentic peer-to-peer engagement. A study has shown that low peer interaction contributes to low student satisfaction (Lowenthal & Dennen, 2017).

Some researchers have built their educational applications to improve teaching and learning. For example, Liao et al. (2011) developed an application to encourage learners to practice math problems. The Web-based Inquiry Science Environment at the University of California at Berkeley has helped K-12 teachers with project-based learning. In higher education, researchers also utilized self-developed educational applications to facilitate teaching and learning (Huynh & Ghimire, 2015; Seiler, Kuhnel, Ifenthaler & Honal, 2019). While these applications have shown effectiveness in improving learning, a critique has been made that they have not paid sufficient attention to aesthetics and the user experience (Davis & Wong, 2007). This might be because most educators are not software developers, and they have limited time and resources to master application development (Huynh & Ghimire, 2015).

There is a paucity of literature that provides information on how to develop an educational web application from scratch. Huynh & Ghimire (2015) is the only literature we have discovered that provides a step-by-step procedure on how to develop a mobile-friendly web application to students who are more comfortable with using mobile devices to get access to learning materials. Although the authors provide detailed and valuable Web development information, there are specific components of developing an educational web application, such as legal regulations and working across industries, that the authors do not address. Our article, therefore, hopes to fill that literature gap and provide practical guidance to other heterogeneous.

THE SOCIAL PERFORMANCE OPTIMIZATION TOOL

To the best of our knowledge, a free or low cost, aesthetically pleasing informal learning platform did not exist before the development of our SPOT application. To respond to the need for something that goes beyond the standard LMS tools, we embarked on the design and development of SPOT application. SPOT is an environment in which students can initiate informal discussions with peers anonymously; and, in the meantime, view their learning performance as well as that of their peers, visualized through the health and emotional states of their animated pet dog avatar. Currently, SPOT is a standalone online application that provides performance visualization, a discussion forum, and direct one-to-one messaging (see Figure 1). Because of The Family Educational Rights and Privacy Act (FERPA) and the ethical need to protect students' privacy, the voluntary element of SPOT participation, and the anonymity of the student-selected pseudonyms are crucial. FERPA is a "federal privacy law that gives parents certain protections with regard to their children's education records" (Department of Education, 2007).

METHODOLOGY

Research Focus

This case study aims to elucidate the unique elements of a heterogeneous team of educators and software developers, working across time zones, languages, and professional backgrounds. Through the analysis of the processes, successes, and challenges, we identify best practices to inform both the literature and practice.

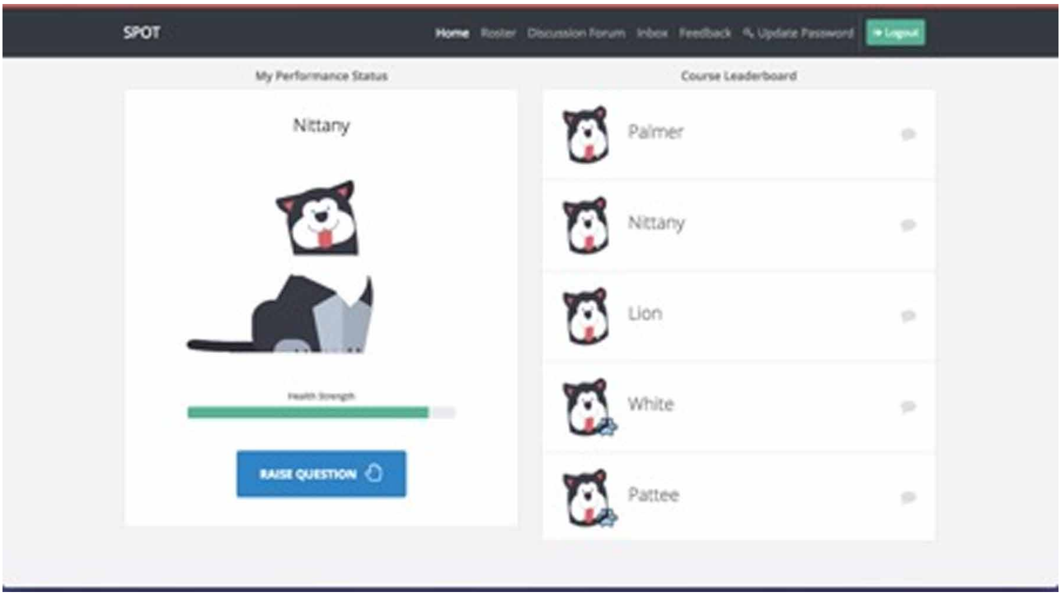
Study Design

This was an interpretive qualitative case study (Yin, 2014) relying primarily on observations and researcher reflections. It is an interpretive study in that we view that "our knowledge of reality is gained only through social constructions such as language, consciousness, shared meanings, documents, tools, and other artifacts" (Klein & Myers, 1999, p. 69). Because this study was intended to serve both research and practice, we established ourselves as reflective practitioners and, throughout the project, analyzed the process as well as our participation and perspectives.

The project started in the spring of 2017 after the authors and research team were awarded a university grant to fund the initial development and testing of the SPOT application. Throughout the two years that the project was funded, we designed, developed, deployed, and evaluated the web application as well as our processes, successes, and challenges.

International Journal of Web-Based Learning and Teaching Technologies
Volume 16 • Issue 4 • July-August 2021

Figure 1. SPOT Homepage



Data Collection

Consistent with the interpretive case study methodology, the research-practitioner team was immersed in the data collection throughout the project.

Data Analysis

The data were analyzed through a thematic framework intended to identify explicit processes, successes, challenges, and ultimately best practices for moving forward with similar projects.

FINDINGS

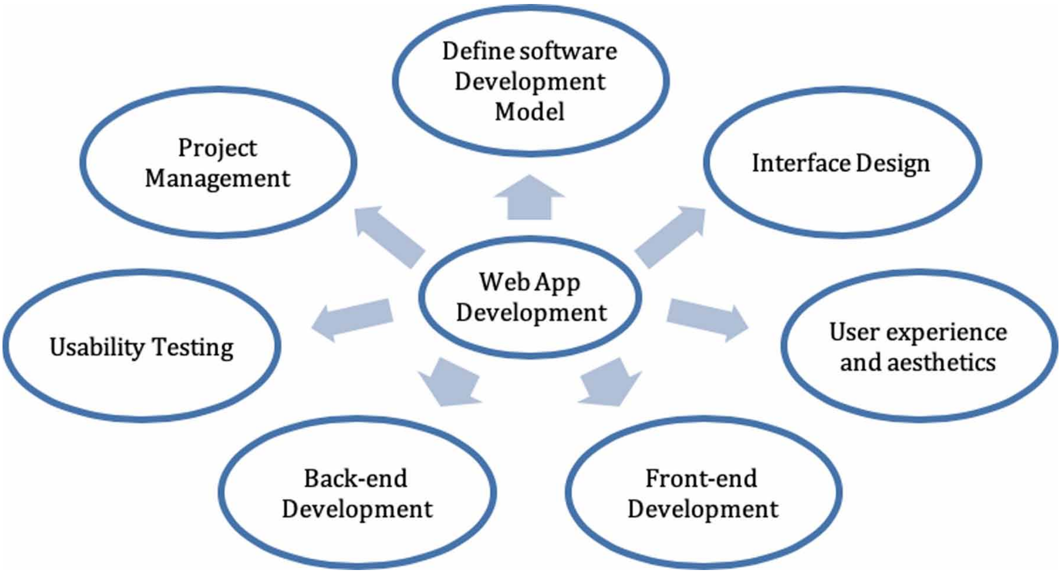
In this section, we describe the processes of how we adopted our software development model, followed by design and development processes, along with the best practices of managing the project. Figure 2. provides a visual overview of the findings.

Software Development Model

The very first step was to decide which software design model to adopt. Many software-development models, such as the incremental commitment model (ICM) (Pew & Mavor, 2007) that extends the spiral model (Boehm, 1988), increase demand for software applications and compete for shorter delivery time to the market. These models make it easier to create and maintain a complex information system successfully with frequent and rapid change. Among the factors of successful system development (Pew & Mavor, 2007), *Incremental growth of system definition and stakeholder commitment,* and *iterative system definition and development* are the two factors that our project particularly utilized. We did not know what features would emerge from our user experience survey results and expected several rounds of modifications. As a result, we decided to use principles from incremental and agile practices along with the tools that we believed would facilitate our collaborative development work.

Following ICM, we set each iteration to one semester, which equals three software releases (Spring, Summer, and Fall semesters). In each iteration, we added features that were suggested by

Figure 2. Web application development process overview



the stakeholders—students and instructors. We also evaluated risks and decided where we could mitigate or eliminate them. When we could not find a timely solution, the features were pushed to later releases. For example, we had a request to add the capability to enter mathematical formulas in LaTex and display them. After considering our limited resources, we decided to prorogue the feature.

Interface Design

We started by identifying the research problem and searching for possible existing solutions. After finding nothing on the market that we could easily adopt, we decided to build our application to fill the gap. In order to make sure the application was worth developing, we conducted a needs assessment by talking to people informally from different fields, different occupations, and different age groups of learners. After receiving positive feedback and suggestions, we submitted and received a seed grant to build the application.

User Experience and Aesthetics

We completed two phases of user interface (UI) design iterations, following the learning sciences principle that design-based research (DBR) is used in learning environments that are constantly modified to facilitate learning (Barb, 2014). In the first phase, we created a markup application using HTML5 and Cascading Style Sheets (CSS). We used an open-source pet dog sourced from GitHub, which saved us a tremendous amount of time and resources from creating our own animated avatar. After the initial design was complete, we distributed a UI design survey to online learners for feedback. Fifty-five learners responded to the survey and we used their feedback to develop SPOT in the second iteration.

Front-End Components Development

We separated the development into front-end and back-end development. The decision to separate the application to the front-end and back-end is twofold: (a) It allows two development teams to focus on their responsibility without having to wait for the other team, (b) It allows teams to reduce the wait time between the front-end and back-end members and enable both groups to work semi-autonomously was critical. We relied on the REST (representational state transfer) architecture (Fielding & Taylor, 2000) and carefully followed the Model-View-Controller (MVC) design principle that studies have suggested to be valuable in developing real-world systems (Pepper et al., 2002; Liu et al., 2011) during system development.

The front-end components of the application are created using the Angular2 framework. Angular2 is an open-source application framework developed by engineers at Google. Apart from its high performance in rendering UI and setting Document Object Model values dynamically through data binding, we chose to use Angular2 to develop the front-end components for its ability to create reusable web components (Google, 2016). Web components are a suite of different technologies allowing developers to create reusable custom UI components with functionality encapsulated in the component, which can be reused in web applications (Mozilla, n.d.). In our application, for example, we separated the display logic into four components: home component, dashboard component, forum component, and message component. We also created some reusable components (e.g. forum message components) to fit our specific needs.

We also integrated Bootstrap in the front-end component. Bootstrap is a toolkit that integrates HTML, CSS, and JavaScript into a grid system for fast UI development. To accommodate different types of devices (i.e., desktop, tablet, smartphone), the layout of the UI must adjust to the screen size, and Bootstrap enables shorter development time to accomplish a consistent look-and-feel on different devices.

Back-End Services Development

We adopted According Separation of Concerns (SoC) concept to develop the back-end service considering following advantages(Garcia et al., 2003): (1) front-end and back-end developers can concentrate on their tasks, (2) database management system and the back-end language can be changed without affecting the front-end, (3) user interface can be simple or complex depending on the front-end, and (4) programming issues or bugs can be isolated for easier troubleshooting.

To implement the concept of Separation of Concerns (SoC), the APIs must be able to serve the front-end components independently. Specifically, the back-end services were implemented using PHP, and the data was stored in a MySQL database. The back-end services were exposed through an endpoint URL with additional information appended to the endpoint URL or JSON objects. An illustration of using REST APIs between the front-end components and the back-end services is shown in Figure 3.

Usability Testing

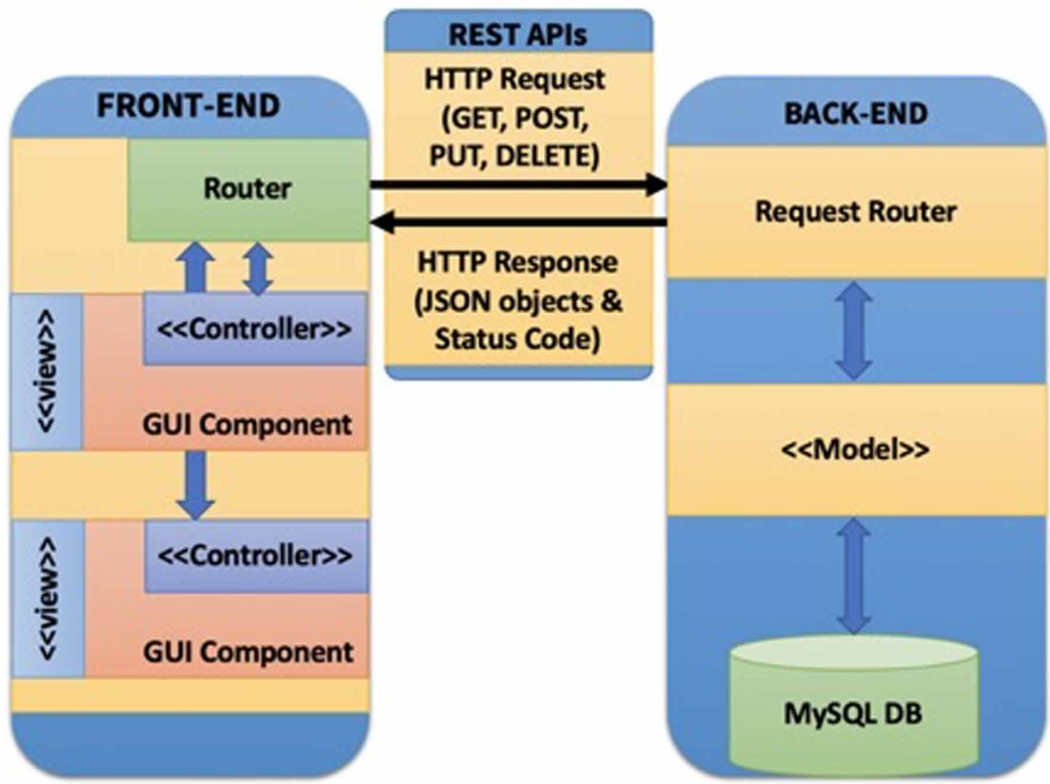
In the pilot phase, we used two usability-testing approaches: (a) Have undergraduate and graduate students take a survey with static screenshots of the application (low-fidelity prototype) to learn about their experiences; (b) Ask colleagues and friends to interact with the functional prototype of the application for feedback.

Twenty-two students and peers responded to the survey, and six students responded to the interview. Based on the feedback collected from the pilot phase, the functional prototype of the application was revised (Wang, Gregg, Yeh, Heiser & Diehl, 2019). We then implemented two additional phases: a soft launch and the actual launch. In the soft launch phase, we asked students of online courses to provide as much feedback as possible for revision. In the launch phase, we once again asked students to test the revised functional prototype for feedback for another round of revisions. Understanding that the application development has release dates but does not have a permanent close date helps us be mindful that the application should be continuously updated and improved.

Project Management

A crucial element of application development is project management as the team members varied not only in the area of expertise but also in geographic locations covering two continents. Project

Figure 3. An illustration of communication using RESTful APIs between front-end components and back-end services with annotation of MVC

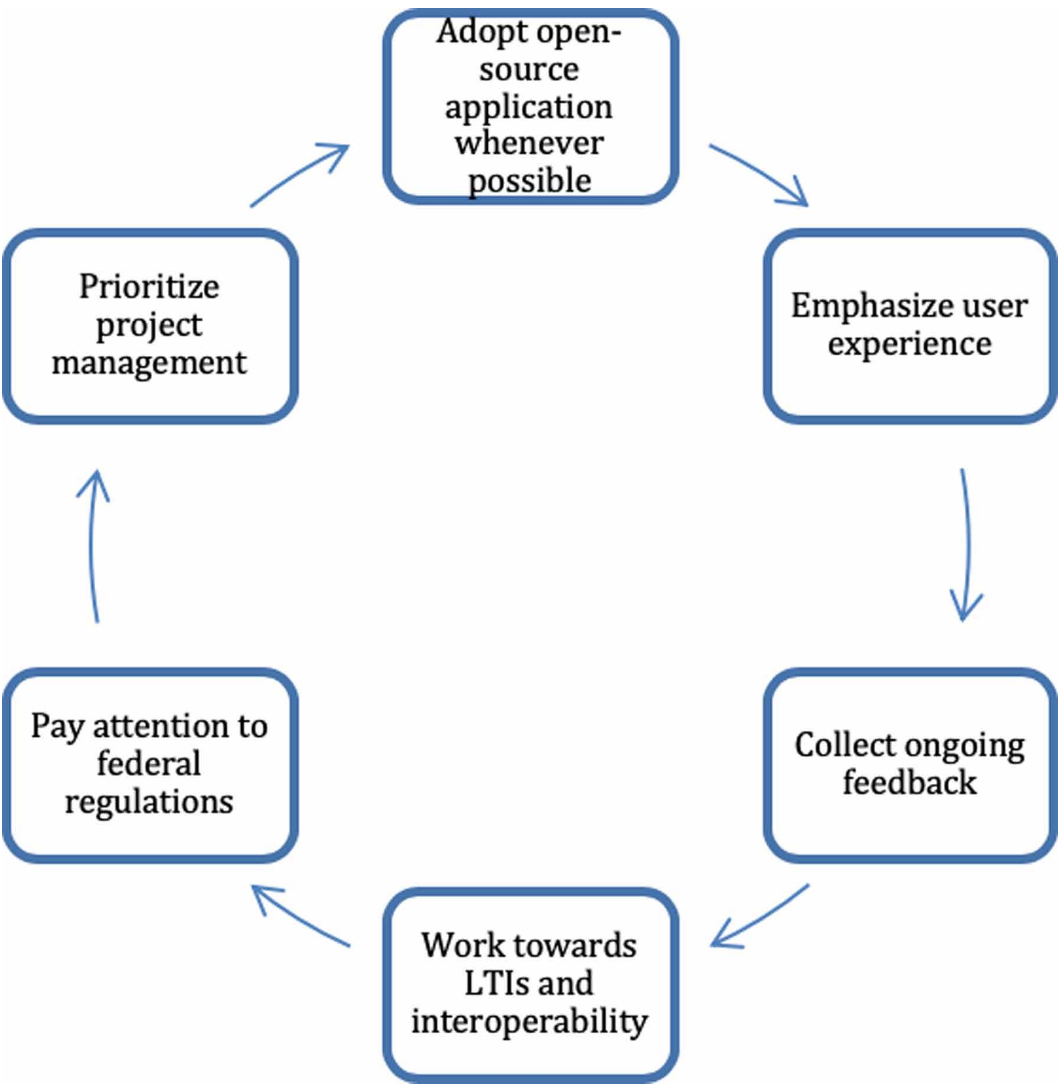


management serves as a gatekeeper to guarantee that the application development meets the deadline and to ensure every process goes smoothly (Kerzner & Kerzner, 2017). We formed a team of different expertise and skills, including front-end developers, a back-end developer, educational researchers, and online learning experts. Because everyone on the team had an equivalent of a full-time job, the time when team members worked also varied greatly. In reflection, the success of the project can be attributed to flexible software design models and effective communication tools that fit the characteristics of our team. We used two types of communication methods to ensure that everyone is apprised of the project progress and to resolve any existing development issues: Scrum and detailed documentation.

Scrum(™) process framework helped us to share team knowledge and expertise in decision making. Scrum is a method adopted by software development companies and projects that typically have a quick daily meeting to discuss the to-do list for that day (Schwaber & Sutherland, 2017). We iterated Daily Scrum to fit in our schedule. The research team and representatives from the design team held research team meetings each month. Other meetings were scheduled on demand. All the video meetings and screen sharing were conducted through the Zoom conference application. We found both features to be helpful and productive for our team building and meeting needs.

Documentation was utilized as another communication method. The design team used GitHub as a collaboration and issue tracking system, including the "Issues" feature to manage milestones/sprints in addition to software bugs. Between the development groups, we used Swagger Documentation[1] as an interface for API-related discussions. The unambiguous documentation effort not only enabled the development groups to work independently, but it also allowed non-technical members to see

Figure 4. Overview of implications



the system at a high level. As a result, they were able to suggest new features, point out issues, and submit bug reports, which indeed helped our team function effectively.

IMPLICATIONS

The following list of suggestions is based on our successes as well as lessons learned. Figure 4. provides an overview of the implications.

Adopt Open-Source Applications Whenever Possible

The amount of time and capital that these free resources have saved us in developing the application is substantial. It is increasingly common to create applications from using tools and services that are already available online. The open-source or free resources we have taken full advantage of are:

GitHub, Angular2, Bootstrap, MySQL, REST web service, and Google Drive. However, it is also often the case that several similar tools are available. It may be difficult to determine which one will best serve the project. In such situations, we suggest that teams: (a) ask colleagues or friends for recommendations or (b) stop trying to find the "best" option, instead, settle for the "practical and acceptable" option. To clarify (b), in today's technology-enhanced world, new tools are created so prolifically and quickly that to track all available or "best" options may hinder the development process. It is essential to set a clear deadline and use the most appropriate tools after a reasonable survey of tools is undertaken. Educators should make equitable decisions as they apply learning technologies to their course designs. The technologies applied should offer advantages including multimodal media enrichment, social interaction and asynchronous functionality to ensure perpetual access to course content and communication (Bates, 2008).

Emphasize User Experience

Aesthetic design can reinforce positive emotions to increase productivity and effectiveness (Aspinwall, 1998; Chawda et al., 2005; Crilly et al., 2008; Norman, 2004). In educational settings, designers and developers should be extra mindful of how the aesthetic design of an application can have extra impacts on learners' willingness to interact with the application. Similar to commercial products, we need buy-in from learners to make the educational tools sustainable and to have meaningful influence on the tools' intended purposes. If a tool with a solid theoretical foundation or effective pedagogy strategy has poor user experience, learners will not enjoy using it. As a result, the theory or strategy may be seen as ineffective when, in fact, the root cause is the poor user experience. Although user experience is typically not the main focus of educational researchers, it remains an important factor when building educational technology.

To be cognizant of user experiences, we conducted six interviews using the think-aloud approach through an online video conferencing tool. Specifically, interviewees were asked to perform a set of tasks while talking aloud about their thinking processes when performing the tasks; and then asked about their thoughts after their tasks were completed. All of the six interviewees did not have any problems in completing the tasks, and the majority of them kept informing the interviewer "the interface is simple to navigate through" (Wang, Gregg, Yeh, Heiser & Diehl, 2019).

Collect Ongoing Feedback

Constantly soliciting feedback and suggestions is beneficial to improving the user experience of the application. There have been many suggestions about involving users in the design process as early as possible, sometimes as early as at the beginning of the project, when feedback from low-fidelity design (wireframe or drawing) is still valuable (Abras, Maloney & Preece, 2004). Waiting until the first version of a functional prototype is ready to collect feedback can be costly if changes are needed. However, the research team should determine what feedback to adopt and what feedback not to. Everyone has a preference regarding aesthetics and functionality; therefore, different people will provide different feedback. It is necessary to determine whether feedback is necessary for the application to function or it is just nice to have the function.

Work Towards Learning Tools Interoperability (LTIs) and Interoperability

LTI has provided significant benefits, such as improving productivity and engagement. In our case, we intended to build SPOT so it could be embedded into Canvas, a popular LMS. Unfortunately, due to the institutional regulations, we could only develop the application as an external application. As a result, we needed to manually export grades from Canvas and then import them into SPOT. There are two apparent drawbacks to this approach. First, the process is time-consuming. We often had to remind the instructor to export grades whenever new ones were available and send them to us. We then had to import the grades manually to SPOT, which added a significant burden on both the research team and the instructors. Second, learners could not see their performance visualization immediately when Canvas notified them that new grades were available. The lag reduced learners' interest in using SPOT. For companies and research teams that have the same intention as our team, it would be wise to confirm with the target institution on LTI at the beginning of the development stage.

Pay Attention To Federal Regulations

Educational-related federal regulations need to be considered throughout the development process. Another issue researchers need to be cognizant of is that the functionalities of the tool do not violate federal or institutional regulations, such as FERPA and the Americans with Disabilities Act Compliance. Due to cost barriers and regulatory oversight, educational technologies are often not initially developed to meet the needs of a diverse student population that requires inclusive design interfaces and universal design principles (Schwartz, 2004). It is educators' or educational researchers' responsibility to ensure compliance.

Prioritize Project Management

Project management procedures need to be implemented early. In our case, everyone on the team had a full-time position and could only work on the project on a part-time basis. Furthermore, team member availability varied widely. Working in synchrony was a significant challenge. We would not be able to make any meaningful progress had we not employed strategic project management at the beginning of the project. The following suggestions are important elements that stem from lessons learned and improvements in our team's collaboration (see Figure 5 for visual demonstration):

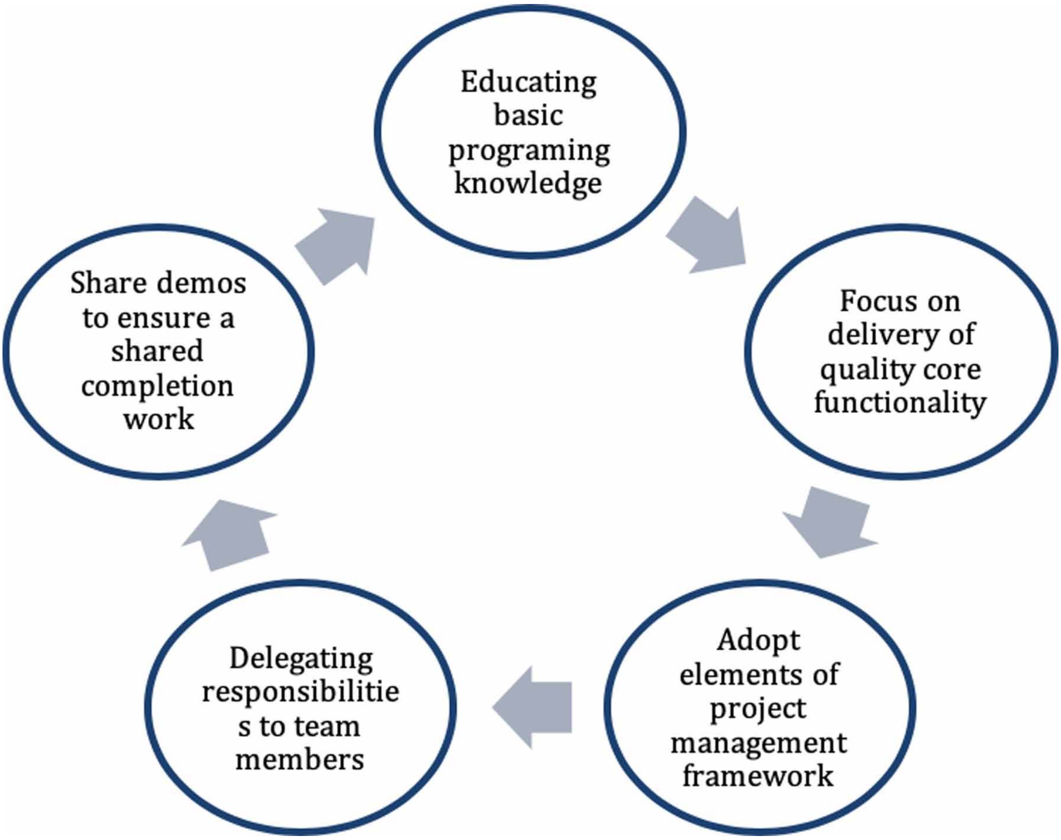
- Educating non-developer teammates about basic programming knowledge and terminologies can be very effective in terms of communications and productivity.
- Formulating business logic and managing scope creep to focus on the delivery of quality core functionality
- Borrowing elements of the project management framework (Schwaber, 2004). For example, use short stand-ups to elevate insight into how each team member is progressing on their stories without centralizing authority
- Delegating responsibilities allows every team member to participate in project implementation and ensures application features are delivered, and bugs are fixed.
- Sharing demos of completed work when possible and ensuring a shared definition of what completion means.

CONCLUSION

SPOT has two releases with minor modifications each semester and is still a work-in-progress application. In the meantime, we continue to run pilot studies to evaluate its efficiency and to improve its features and usability iteratively. In this study, we describe how we were able to use limited resources to create a software system for education with a heterogeneous team and what we believe are valuable to people who are facing similar positions. Although there are many "lessons learned" reports in commercial or enterprise settings (Maciaszek & Liong, 2005) and one best practice literature in educational space, we believe our article fills the gap in helping educational researchers and professional developers who plan on developing educational web applications. Additionally, this work fills the gap between academic members and software systems, and educational theory and real-world projects.

As mentioned before, the challenges we faced include different work schedules, working styles, lack of experience in the design and development of a system, and a diverse background without an identified project leader. We learned that technical skills and non-technical skills are equally important.

Figure 5. Overview of suggestions on team collaboration



To build the rapport between technical and non-technical members in a project, measures and practices need to be taken to cultivate shared language in order to propel the project toward the finish line. Future research and development should adopt a longitudinal approach to assess the functionalities of SPOT, its long-term impacts, and the types of courses for which SPOT might be best suited.

REFERENCES

Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). User-centered design. In Encyclopedia of Human-Computer Interaction. Thousand Oaks, CA: Sage Publications.

Aspinwall, L. G. (1998). Rethinking the role of positive affect in self-regulation. *Motiv. Emot., 22*(1), 1– 32.

Bates, T. (2008). Transforming distance education through new technologies. In D. Murphy, T. D. Evans, & M. Haughey (Eds.), *International handbook of distance education* (pp. 217–236). Emerald Group Publishing Limited.

Berger, E. J., & Krousgrill, C. M. (2012). *HigherEd 2.0: Web 2.0 in higher education*. Interactive Multimedia.

Bhattacharjee, S., Gopal, R. D., & Sanders, G. L. (2003). Digital music and online sharing: Software piracy 2.0? *Communications of the ACM*, *46*(7), 107–111. doi:10.1145/792704.792707

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61–72. doi:10.1109/2.59

Cavus, N., Uzunboylu, H., & Ibrahim, D. (2007). Assessing the success rate of students using a Learning Management System together with a collaborative tool in Web-based teaching of programming languages. *Journal of Educational Computing Research*, *36*(3), 301–321. doi:10.2190/T728-G676-4N18-6871

Chawda, B., Craft, B., Cairns, P., Heesch, D., & Rüger, S. (2005). Do “attractive things work better”? An exploration of search tool visualisations. *19th British HCI Group Annual Conference: The Bigger Picture*, 46–51.

Crilly, N., Good, D., Matravers, D., & Clarkson, P. J. (2008). Design as communication: Exploring the validity and utility of relating intention to interpretation. *Design Studies*, *29*(5), 425–457. doi:10.1016/j.destud.2008.05.002

Dron, J., & Ostashewski, N. (2015). Seeking connectivist freedom and instructivist safety in a MOOC. *Educación XX1, 18*(2), 51-76.

Department of Education. (2007). *Parents’ Guide to the Family Educational Rights and Privacy Act: Rights Regarding Children’s Education Records*. Retrieved from https://www2.ed.gov/policy/gen/guid/fpco/brochures/parents.html

Fielding, R. T., & Taylor, R. N. (2000). Principled design of the modern web architecture. In *ICSE’00: Proceedings of the 22nd International Conference on Software Engineering*. ACM. doi:10.1145/337180.337228

Gao, F., Zhang, T., & Franklin, T. (2013). Designing asynchronous online discussion environments: Recent progress and possible future directions. *British Journal of Educational Technology*, *44*(3), 469–483. doi:10.1111/j.1467-8535.2012.01330.x

Garcia, A., Sant’Anna, C., Chavez, C., da Silva, V. T., de Lucena, C. J., & von Staa, A. (2003). Separation of concerns in multi-agent systems: An empirical study. In *International Workshop on Software Engineering for Large-Scale Multi-agent Systems*. Springer.

Google. (2016). *Angular 2*. Retrieved from https://angular.io/

Hewitt, A., & Forte, A. (2006). *Crossing boundaries: Identity management and student/faculty relationships on the Facebook.* Poster presented at CSCW, Banff, Alberta, Canada.

Huynh, M., & Ghimire, P. (2015). Learning by doing: How to develop a cross-platform web app. *Journal of Information Technology Education: Innovations in Practice*, *14*, 145–169. doi:10.28945/2213

Kerzner, H., & Kerzner, H. R. (2017). *Project management: A systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

Kirkwood, A., & Price, L. (2014). Technology-enhanced learning and teaching in higher education: What is ‘enhanced’ and how do we know? A critical literature review. *Learning, Media and Technology*, *39*(1), 6–36. doi:10.1080/17439884.2013.770404

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *Management Information Systems Quarterly*, *23*(1), 67–88. doi:10.2307/249410

Lamberton, C., & Stephen, A. T. (2016). A thematic exploration of digital, social media, and mobile marketing: Research evolution from 2000 to 2015 and an agenda for future inquiry. *Journal of Marketing*, *80*(6), 146–172. doi:10.1509/jm.15.0415

Legon, R., & Garrett, R. (2018). The changing landscape of online education (CHLOE2): A deeper Dive. *Quality Matters & Adventures survey of chief online officers*. Retrieved from https://www.qualitymatters.org/sites/default/files/research-docs-pdfs/2018-QM-Eduventures-CHLOE-2-Report.pdf

Liao, C. C. Y., Chen, Z. H., Cheng, H. N. H., Chen, F. C., & Chan, T. W. (2011). My-Mini-Pet: A handheld pet-nurturing game to engage students in arithmetic practices. *Journal of Computer Assisted Learning*, *27*(1), 76–89. doi:10.1111/j.1365-2729.2010.00367.x

Liu, Y., Liang, X., Xu, L., Staples, M., & Zhu, L. (2011). Composing enterprise mashup components and services using architecture integration patterns. *Journal of Systems and Software*, *84*(9), 1436–1446. doi:10.1016/j.jss.2011.01.030

Lowenthal, P. R., & Dennen, V. P. (2017). Social presence, identity, and online learning: Research development and needs. *Distance Education*, *38*(2), 137–140. doi:10.1080/01587919.2017.1335172

Maciaszek, L. A., & Liong, B. L. (2005). *Practical software engineering: a case study approach*. Pearson Education.

Mozilla. (n.d.). *Web components*. Retrieved from https://developer.mozilla.org/en-US/docs/Web/Web_Components

Norman, D. A. (2004). *Emotional design*. Basic Books. doi:10.1145/985600.966013

Pepper, P., Cebulla, M., Didrich, K., & Grieskamp, W. (2002). From program languages to software languages. *Journal of Systems and Software*, *60*(2), 91–101. doi:10.1016/S0164-1212(01)00082-6

Pew, R. W., & Mavor, A. S. (Eds.). (2007). *Human-system integration in the development process: A new look*. National Academy Press.

Schwaber, K., & Sutherland, J. (2017). *The scrum guide*. https://www.scrumguides.org/

Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft press.

Schwartz, L. (2004). Advanced accessibility features for inclusive distance education. *International Review of Research in Open and Distributed Learning*, *5*(3). Advance online publication. doi:10.19173/irrodl.v5i3.199

Wang, N., Gregg, A., Yeh, K.-C., Heiser, R., & Diehl, W. C. (2019). Pet avatars, performance visualization, and social presence. *International Journal of Technology in Teaching and Learning*, *15*(1), 1–18.

Wenger, E. (2005). *Communities of practice: A brief introduction*. Retrieved from http://www.ewenger.com/theory/index.htm

ENDNOTE

[1] https://swagger.io/docs/

Nicole Wang-Trexler, PhD, is the Associate Director of Data Science and Research at Wharton Customer Analytics center and AI for Business center at the University of Pennsylvania. Her research focuses on online learning, evaluation, data analytics, and big data.has earned her Ph.D. in Learning, Design, and Technology with a dual-title in Comparative International Education at the College of Education, the Pennsylvania State University. Her research focuses on online learning, MOOCs, evaluation, and big data.

Martin K.-C. Yeh, PhD, is an Assistant Professor of Information Sciences and Technology at the Pennsylvania State University, Brandywine Campus. His research interests focus on understanding the impact of learning technologies and pedagogical strategies in human cognition and learning outcomes. He is also interested in human-computer interaction with regard to software engineering.

William C. Diehl, PhD, is an Assistant Professor of Lifelong Learning and Adult Education, College of Education at the Pennsylvania State University. His research interests include distance education, educational technology, adult education and intercultural communication.

Rebecca E. Heiser is a doctoral student at Athabasca University studying quality dimensions and assurance standards for transnational online distance education. She serves as the Interviews Editor for The American Journal of Distance Education, participates in the learning analytics research cohort with Oregon State University's Ecampus Research Unit, and acts as an OER Fellow with the Open Education Group. Also, Rebecca is the lead instructional designer for The Pennsylvania State University's Lifelong Learning and Adult Education online graduate program with World Campus.

Andrea Gregg, PhD is the Director of Online Pedagogy and an Assistant Teaching Professor in Penn State's Department of Mechanical Engineering. She facilitates faculty development and instructional design to maximize teaching and learning quality with a focus on online learning. Her research evaluates the intersection of educational technology supported pedagogical innovations (e.g. digital badges, adaptive learning) and stakeholder (e.g. students, instructors, instructional designers) experiences.

Ling Tran is a software developer consultant at ThoughtWorks, Inc. Her client engagements have involved application development in various language frameworks in Ruby, JavaScript, Golang; and cross-platform mobile frameworks like React Native. As a generalist consultant, she is also versed in Agile best practices and advocates for cross-functional collaborations.

Chenyang Zhu is a PhD student in the department of Electronics and Computer Sciences at the University of Southampton. His research interests include Formal methods, Time Modelling, Machine Learning, Data Visualization.