Enriching Technical Communication Education: Collaborating Across Disciplines and Cultures to Develop the piClinic Console

Experience Report

Robert B. Watson
Department of Technical Communication,
Mercer University School of Engineering,
Macon, GA, USA 31220
watson_rb@mercer.edu

ABSTRACT

I have observed that technical communication lessons with a real-world impact provide students with a more engaging learning experience. While this observation is consistent with the theory of adult learning, it is an ongoing challenge to keep the examples fresh from term to term. This report describes my experience developing the piClinic Console from the beginning of project in the spring of 2017 to the present. The piClinic Console is a low-cost, patient information system designed for use in limited-resource clinics in lower-middle-income countries. In addition to helping these clinics begin to automate their patient records, the project has provided opportunities to demonstrate and apply technical communication skills in a real-world context and has initiated inter-departmental and international collaborations. This experience report describes a brief history of how the project began, the design and educational lessons it has provided to our students, the collaborations it has encouraged, some lessons learned, and the plans for future development.

CCS CONCEPTS

• Applied computing~Health care information systems • Applied computing~Collaborative learning

KEYWORDS

Electronic Health Record (EHR) systems, Information Communication Technology for Development (ICT4D), International teams, Technical Communication education.

ACM Reference format:

R. Watson 2018. Enriching Technical Communication Education: Collaborating Across Disciplines and Cultures to Develop the piClinic Console: Experience Report. In *Proceedings of The 36th ACM International Conference on the Design of Communication (SIGDOC '18). ACM, New York, NY, USA, 8 pages.* https://doi.org/10.1145/3233756.3233929

SIGDOC '18, August 3--5 2018, Milwaukee, WI, USA

978-1-4503-5935-1/18/08...$15.00
https://doi.org/10.1145/3233756.3233929

1 INTRODUCTION

Course and assignment feedback have made clear the value of technical communication lessons that have a real-world and practical connection. My students consistently find such lessons to be interesting, valuable, and engaging. The theory of adult learning [1] says that adult learners require a need to know the lesson, see the need for the lesson, and be motivated to learn the lesson. The lessons with a real-world connection seem to provide my students with the necessity, relevance, and motivation that create an engaging and productive learning experience. However, it is an ongoing challenge to find fresh examples from term to term. To further complicate this challenge, technical communication at Mercer University includes a wide range of topics for which to find real-world examples—topics that include web design, usability testing, project management, social-media management, instructional design, and, of course technical writing and editing. This experience report describes my initial experience with a project I am developing to solve a real-world problem and that has provided a steady source of technical communication experiences across the breadth of our courses.

In this report, I describe my experience developing the piClinic Console from the beginning of the project in the spring of 2017 to the present as it is readied for field-testing by clinical personnel. This experience report describes a brief history of how the project began, the design and educational lessons it has provided to our students, the collaborations it has encouraged, some lessons learned, and the plans for its future development and application in the classroom.

2 BACKGROUND

I observed the end-user need for the piClinic Console when I provided communication system support for medical missions at an eastern Honduran clinic in 2014 and 2015. During these missions, I observed the clinic's record keeping and inquired about the information flows as the staff attended to more than 1,000 patients each week we were at the clinic. The clinic kept paper records of each patient's visit; however, the clinic staff rarely referenced these records after they completed and filed them. The records contained detailed information that could inform future patient visits and provide statistical reports for the clinic and the mission teams, if only the data could be accessed. A brief review of the literature [2–4] and follow-up interviews with medical personnel who had experience on similar medical missions agreed that these observations were not outliers.

When I first became aware of this information-management challenge in 2014, the options to collect and automate patient information were few, complex, and expensive to install and maintain. Late in 2016, however, the Raspberry Pi, a low-cost single-board computer originally released in 2012, was updated to the Model 3 [5, 6]. There was already some precedent for using Raspberry Pi solutions in similar applications [7, 8] and the specifications of the Model 3 suggested that it might be able to provide computing resources sufficient for basic patient-information functions.

In the spring 2017 semester, the Research that Reaches Out program at Mercer University funded a small research project that enabled the Technical Communication Department to develop and test a system based on the Raspberry Pi Model 3. Student researchers tested a simulation of the basic patient-information tasks I observed on the missions to determine if a system based on the Raspberry Pi Model 3 could support the basic patient-information tasks that would benefit a small clinic, such as the one I visited in 2014 and 2015. We found that a system based on the single-board computer could support basic patient-information management and reporting tasks, and it could be built for about $300 USD each.

Past experiences of earlier electronic health record (EHR) introductions into limited-resource clinics identified some recurring challenges. The most common approach to automating patient information in a small clinic was to install a complete (or the subset of a complete) EHR system—often at considerable expense. This approach required the clinics to change many aspects of their processes and information workflows to adapt to the new system [2, 9, 10]. The success rate of these projects in limited-resource clinics was disappointing and most successes that had been observed were temporary—often lasting only as long as the clinic received external technical and financial support [2].

Based on the past research [2, 9, 10] and our findings from the spring 2017 prototype testing, we identified an opportunity for a low-cost, low-maintenance, system that did not disrupt existing clinic systems and processes. To develop this system, we launched the program to research and develop what would become the piClinic Console project.

3 DESIGN EXPERIENCE

This section describes the development and testing of the piClinic Console and the technical-communication aspects of the development. The Principles for Digital Development [11], a set of user-centered design principles that support sustainable project design, were adopted as the development philosophy for the piClinic Console. These principles provide a convenient way to communicate the project's design philosophy to current and future students, researchers, and other stakeholders. Development of the piClinic Console progressed by designing with these principles and applying the following aspects of technical communication—each of which provided student research experiences, examples to incorporate into future classes, or both.

3.1 User Research

The Principles of Digital Development include "Design with the User" and "Understand the Existing Ecosystem" [11], and they reflect the contextual inquiry and design methods that Holtzblatt et al. describe [12]. While the results of the spring 2017 feasibility test were encouraging, they were derived from limited, informal observations made several years earlier. Understanding the end user's requirements sufficiently to provide a successful solution and being data-driven (another principle) would need a formal contextual inquiry of the patient-record management systems currently in use at a larger set of clinics. At the beginning of the summer 2017 semester, I studied the operations at five Honduran clinics and interviewed their staffs to update the piClinic Console's design requirements.

Findings from the clinic visits in 2017 confirmed the initial observation that automating some of the patient information could improve patient information management and reporting and that the clinic staff interviewed were generally receptive to such automation, similar what Hochwarter, et al. [4] found. The clinic visits also simplified the physical requirements of the system. Initially, it was believed that an environmentally protected system would be required. While some field teams would find such systems useful, clinics situated in an existing building could use standard, office-style computing equipment (such as the computer, monitor, and keyboard), which simplified the initial system requirements.

The biggest change to the earlier assumptions that the clinic visits identified had to do with the need to digitize the patient records. Our earlier prototype testing determined that a Raspberry Pi-based system could support digitizing and storing many patient visit records; however, the clinic visits and interviews indicated that image capture and digitization of patient records was not a high-value feature to the end users. The clinics visited in 2017, as well as those visited later in 2018, have used paper records for many years and have established satisfactory storage and identification systems.

The more detailed study of the clinics revealed that they required the system to record only the subset of patient-visit data that enabled summary reports. With this data in the system, clinics could automate the reporting of patient activities to the national health system and reduce the time required for a clinic to produce their reports from days to seconds.

Collecting the data to develop the requirements for the piClinic Console provided several opportunities to conduct on-site research. The user research in this context provided a first-person case study of how user research is applied to product development and how it clarifies and identifies latent requirements.

3.2 Agile Project Management

The changes to the end user's requirements of the system that resulted from the findings of the clinic visits were significant, but not uncommon. Changes to system requirements and the development plan have been the norm throughout the project, suggesting that an Agile project management style would best suit the project [13, 14].

Agile project management seeks to provide increasing value to the customer or end-user by iteratively testing the value of new features and reprioritizing planned features. The evaluation of planned features includes determining which features will provide the most value to the customer in a dynamic context [14]. The project's grant-based funding model and reliance on student resources—both of which varied from week to week and term to term—created the very definition of a dynamic context.

It is not uncommon to discover latent requirements when developing a new system—some requirements can only be identified as the system is built and used by the customer. Agile project management is designed to produce the best possible system by using the latest information as it becomes available. Relying on student resources and independent study projects to develop project deliverables introduces considerable variability into a project. To succeed in this environment, the plan for this project consists of a backlog of tasks that are addressed as their priorities require and the opportunities to complete them present themselves. Such an approach differs from a plan-driven project, which consists of a list of sequential tasks that are defined in advance and that must be completed in a specific order.

The Agile project management processes are iterative. Each iteration's goal is to add end-user value and inform the next iteration [14]. To iterate towards a field-testable prototype by the summer of 2018, the project was broken into semester-sized *epics* (in Agile terminology) that would provide necessary end-user value and answer critical product development questions. Each epic drove the individual task (*story*, in Agile terminology) priorities, and the stories were completed and reviewed in two-week cycles (*sprints*). Table 1 lists the research questions that informed the goals for each semester to date.

A critical finding from the summer 2017 trip was that continued development of a successful system would require collaboration and cooperation with Honduran expertise to prevent the design from wandering away from the known and latent end-user requirements of the system. With support from the United States Embassy in Honduras, collaboration was established over the summer between the Technical Communication Department of Mercer University and faculty from the Departamento de Salud Pública in the Facultad de Ciencias Médicas (the Department of Public Health in the School of Medical Science) at the Universidad Nacional Autónoma de Honduras (UNAH). The plan for the autumn 2017 semester was established by the end of the summer.

Table 1: Project Research Questions by Semester

Semester	Research Question
Spring 2017	Can a Raspberry Pi-based system support the functionality that a small clinic requires to manage their patient information and reporting?
Summer 2017	Can the end-user requirements be converted into tasks and features that can be prepared for field-testing over the coming academic year?
Autumn 2017	Will the Raspberry Pi support the detailed application infrastructure (back-end database functions) necessary to provide the required end-user functionality?
Spring 2018	What features will help clinics the most?
Summer 2018	Can a Raspberry Pi-based system improve the patient information and reporting functions of a small clinic?

The research question for the autumn 2017 semester was, "Will the Raspberry Pi support the detailed application infrastructure (back-end database functions) necessary to provide the required end-user functionality?" Another Digital Development Principle "using open standards, open data, and open innovation," [11] encouraged selecting the Debian Linux operating system and other open-source system software. These selections also supported the "Design for Scale" principle in that the software being developed for the Raspberry Pi platform will run on many other, more powerful platforms.

The Debian Linux operating system supports Firefox and Chromium web browsers so we assumed that we could develop the user interface with little risk. There was, however, some uncertainty that the small, single-board computer could run the patient-information data collection and reporting functions the end users required. For the autumn 2017 semester, development focused on the information flows and the "back-end" database functions required to support them. Iterative user testing by student researchers and participants demonstrated that the system could support at least 20,000 patients on file and at least 100 visits per day—more than enough for the target clinics.

Although the user interface design was not the autumn 2017 semester's priority, a design student expressed interest in working on the user interface as a class project. Agile project management enabled us to accommodate this change and finish autumn 2017 with a tested back-end design and a set of design sketches that inspired the UI redesign.

The goal for spring 2018 was to produce a system that could be field-tested in small, Honduran clinics to answer the summer 2018's research question, "Can a Raspberry Pi-based system improve the patient information and reporting functions of a small clinic?" During the spring 2018 semester, the UNAH faculty and students collaborated with Mercer University to answer the research question, "What features will help clinics the most?"

As with the user research examples, managing the piClinic Console development provided many first-hand examples of how design and development interact in an Agile project. This gave the student researchers first-hand experience in this environment and provided a case study example to incorporate into future technical communication coursework.

3.3 Usability Testing

Usability testing played a critical role in the development of the system since the project started. The Agile approach to system design and implementation revolves around a cycle of research questions, solution development, and testing to answer those questions [14]. Student researchers and participants contributed much to this effort through hands-on participation in designing and conducting the iterative user tests as Barnum [15] describes. These tests provided the empirical data used to answer the research questions posed at each stage of development described in the previous section.

During the spring 2017 prototype testing, student researchers tested the feasibility of digitizing patient record images and, in the process, tested the storage and processing abilities of the Raspberry Pi-based prototype. During the 2017-18 academic year, student researchers tested basic data query and entry tasks to identify user-interface requirements while testing the robustness of the underlying system infrastructure. After incorporating the participant feedback and updating the user interface, student researchers recruited undergraduates with healthcare experience and Spanish fluency to test the workflows and user interfaces in English and Spanish. Student collaborators from the UNAH contributed subject-matter-expert feedback to the design as well.

Student researchers ran many usability test sessions in a variety of settings during the academic year. This provided students with experience in running usability tests in a product-development environment and gave many students from other departments around the university a chance to see some of the technical communication students' work.

3.4 Internationalization

The piClinic Console must support both Spanish and English user interfaces in its first release, so the language support of the system prototype was designed with the Open-Closed Principle described by Shalloway and Trott [16] in mind. Applying this design pattern, the piClinic Console can support additional languages (*Open* to extension of capability) by adding the corresponding translations to enable its deployment in clinics around the world without the need to re-engineer the software (*Closed* to the need for modification of the system). While a requirement of the final project, the internationalized design also supports simultaneous development and testing by American and Honduran collaborators.

3.5 Cross-Department Collaboration

Almost every aspect of the piClinic Console's development has embraced the Digital Development Principle to "Be Collaborative" [11]. As the piClinic Console evolves towards its field test, the diverse nature of the project has invited support from other disciplines around Mercer University and, internationally, from the UNAH. Aspects of the system design provided many tasks for Computer Engineering students to incorporate into their coursework. The international aspect of the project has encouraged collaboration with the Spanish department of Mercer University to help with localization and language-specific testing. As the system matures, it will be available to support public-health research projects in the future.

3.6 Cross-Cultural Collaboration

Development of the piClinic Console could not succeed without local subject-matter expertise in Honduras. Students and faculty from the Facultad de Salud Pública (Public Health Department) in the Facultad de Ciencias Médicas (School of Medical Science) at the UNAH has provided local subject-matter expertise, access to public clinics, and access to the national health system to support the development and field-testing of the piClinic Console prototype. Their faculty has been able to incorporate the project support into their graduate-level curriculum, making the project mutually-beneficial.

Developing the system as a web-based application provides considerable system-design flexibility. Hosting test versions of the system software on the web made it easy to support international collaboration. Having the system available on the web as an unlisted web site (a site with no domain name) has enabled the collaborators in Honduras to provide their subject-matter expertise and conduct local design and usability reviews. This online access has made test-and-review iterations very efficient.

4 EDUCATIONAL EXPERIENCE

In addition to the humanitarian goals of improving patient information management in limited-resource clinics, the project had two key educational goals:

- Provide a range of technical communication projects to involve students and incorporate into our regular technical communication curriculum
- Demonstrate the range of impact that technical communication skills have on a development project.

4.1 Technical Communication Projects

The technical communication projects that the piClinic Console development created include the design and research projects described in the preceding section. As piClinic Console development began in earnest in autumn 2017, most of the projects were completed as special projects and independent study assignments. Courses designed around project work were the easiest-to-adopt piClinic Console development tasks on short notice. Table 2 lists some of the courses and the project development tasks performed by students during the 2017-2018 academic year.

Table 2: Technical Communication Projects by Course

Course	Development Tasks
Web Design	User interface mockup design
Multimedia	Promotional video for website
Usability Testing	Usability tests of project features

The piClinic Console project has provided projects for courses in other departments such as Spanish, whose students participated as usability-testing participants. It also encouraged competitive product analysis and technical feature design projects for computer engineering students. Internationally, the project has provided public-health data, process workflow, and terminology review projects for the UNAH students.

4.2 Technical Communication Impact

The piClinic Console project serves as a tangible example of many aspects of technical communication. We have used the project as one of our displays in recruiting and in information events around the university. These displays have helped promote the department to future students and motivate our current students [17]. Involving students from other departments around the university to participate in design projects and usability testing has helped make other students and faculty around the university aware of the technical communication department and its field of study.

5 DISCUSSION

This section reviews the project's successes, the factors that contributed to those successes, and some of the challenges the project has faced to help future projects.

5.1 Successes

Aligning the project with the university's service-learning and undergraduate research goals helped the project receive support at many levels in the university. The faculty of the Technical Communication Department and the School of Engineering, the Research that Reaches Out program of Mercer University's Quality Enhancement Plan, and the other schools and departments mentioned earlier all provided support for the piClinic Console.

Having general project goals expressed as a dynamic list of tasks that provide end user value, as Agile project management encourages, provided the flexibility that facilitated collaboration within and between departments. Dynamic task management enabled collaboration and made the best possible progress with the resources available to accomplish each semester's tasks. Modular tasks also made it easier to find attractive intersections between the project's backlog of tasks and the courses that could adopt them into their curricula to invite their collaboration.

5.2 Challenges

While the project's development tasks were easy to integrate into project-based courses, incorporating them in the more traditionally-structured courses has been difficult. Development tasks changed too quickly for courses with a more defined curriculum to adopt. However, it should become easier to incorporate project tasks into traditionally-structured courses as the nature of the project stabilizes and tasks are more clearly defined.

Creating the internationalized version of the software added additional complexity early in the project at a time when many features were still provisional. However, addressing this complexity early in development forced the design to accommodate this requirement from the beginning—avoiding the need to refactor the design to support internationalization later. Having bilingual talent contributing to the project mitigated some of the impact of supporting two languages.

Collaboration with the UNAH took time to establish due to the challenges Brewer [18] describes, such as working across languages, time zones, cultures, and disciplines. Collaboration began in the form of email and Skype conversations and improved after meeting in person.

The incremental and iterative approach that resulted from semester-oriented, grant-based funding and the use of student projects to complete tasks made it difficult to collaborate with stakeholders more than a semester in advance. Fortunately, Agile project management made it easy to keep track of the tasks and priorities and the project has received consistent support, so it has been able to deliver a series of successful outcomes. Nevertheless, constantly redefining and reprioritizing tasks is labor-intensive.

5.3 Lessons Learned

This section summarizes some of the key lessons learned from this project that might also apply to other large, technical communication projects.

- **Embrace change.**
  Nothing is now as it appeared it would be earlier. Fortunately, it usually turned out better. In the worst case, it was as good as it could be at the time.
- **Deliver increasing end-user value.**
  Because of uncertainties in just about every aspect of the project, it has been better to promise and deliver an improving project over time rather than a specific deliverable or feature on specific date. Stakeholders might not accept this approach on every project.
- **Define tasks that are easy to share.**
  Tasks that are not dependent on other tasks are easier to share and distribute than those with multiple dependencies. Tasks with dependencies that cannot be avoided should be assigned only to reliable resources.
- **Negotiate course collaborations in advance.**
  As much as possible, let people know of the opportunities the project provides and work with them

to understand how project tasks can meet the needs and interests of their courses. Recognize that not every course can change assignments on short notice.

- **Expect to do more work than planned.**
  To date, the project has taken an incredible amount of effort to launch. This should come as no surprise when mixing technology with students, faculty, and international users, but it is something to keep in mind. Nevertheless, the successes have made the effort worthwhile.

6 CONCLUSIONS AND FUTURE WORK

Although the project is a little over a year old, developing a platform that provides real-world technical communication learning experiences has been valuable to our technical communication students and to those in other departments of Mercer University and the UNAH. A Fulbright Scholars Award will support upcoming research and field-testing of the piClinic Console during the summers of 2018 and 2019, which will provide new technical communication experiences and additional data to inform new features during the intervening and subsequent academic years.

6.1 Future Projects

Data from the summer 2018 research will inform improvements to develop a system that is robust enough to deploy to several small clinics throughout Honduras in 2019 and, eventually, around the world. The open-source development towards this goal supports the “Reuse and improve” Digital Development Principle [11] and will create many technical communication projects in the fields of user documentation, training materials, and even more usability testing. Development and deployment of the project will also create research opportunities in technical communication and other fields of study.

6.2 Future Collaborations

As the system becomes more robust and its features become known by other disciplines such as public health, collaborations with other schools will be pursued, as the Digital Development Principles [11] encourage. For example, public health research could benefit from the data collected and reported by these systems while they provide value to local clinics in other countries.

The international nature of the software encourages developing versions for other countries to use in their clinics. Initial field-testing is being planned with the cooperation of the project’s Honduran collaborators and government health officials. However, the system is also designed to support non-governmental organizations (NGOs) and the information needs of their clinics and medical missions.

The open-source nature of the system supports adoption and adaptation to develop applications that have not yet been imagined. This design also encourages developing local expertise by providing projects for students in the countries that use it.

ACKNOWLEDGEMENTS

The Mercer University Institutional Review Board has approved the piClinic Console research. Mercer University’s School of Engineering, the Research that Reaches Out program of Mercer University’s Quality Enhancement Plan, and the Fulbright Scholars program have provided funding and Clínica Esperanza, Missionary Air Group, and the Universidad Nacional Autónoma de Honduras have also provided support for the piClinic Console.

REFERENCES

[1] Malcolm S. Knowles, Richard A. Swanson and Elwood F. III Holton. 2011. *The Adult Learner, Seventh Edition: The definitive classic in adult education and human resource development*. Taylor & Francis.

[2] Pan American Health Organization. 2016. *Electronic Medical Records in Latin America and the Caribbean: An Analysis of the current situation and recommendations for the Region*.

[3] Eugenio G. Lopez and Karina M. Diaz. 2017. Electronic Health Record in Bolivia and ICT: A Perspective for Latin America. *International Journal of Interactive Multimedia and Artificial Intelligence*. 4, Regular Issue (2017), 96–101. DOI: https://doi.org/10.9781/ijimai.2017.4412.

[4] Stefan Hochwarter, Do Duy Cuong, Nguyen Thi Kim Chuc and Mattias Larsson. 2014. Towards an Electronic Health Record System in Vietnam: A Core Readiness Assessment. *Journal of Health Informatics in Developing Countries*. 8, 2 (Dec. 2014).

[5] Raspberry Pi 3 Model B.: 2016. *https://www.raspberrypi.org/products/raspberry-pi-3-model-b/*. Accessed: 2017-06-02.

[6] A History Of The Raspberry Pi.: 2015. *http://novadigitalmedia.com/history-raspberry-pi/*. Accessed: 2017-06-02.

[7] 5 Ways the Raspberry Pi Can Be Used in Schools.: 2015. *http://www.makeuseof.com/tag/5-ways-raspberry-pi-can-used-schools/*. Accessed: 2017-06-22.

[8] E. P. M. LaRochelle and J. A. Dobbins-Bucklad. 2015. Where there is no Internet: Experiences from rural Honduras 2013 - 2015: Phase I implementation. *2015 IEEE Global Humanitarian Technology Conference (GHTC)* (Oct. 2015), 87–91.

[9] Lemai Nguyen, Emilia Bellucci and Linh Thuy Nguyen. 2014. Electronic health records implementation: An evaluation of information system impact and contingency factors. *International Journal of Medical Informatics*. 83, 11 (Nov. 2014), 779–796. DOI: https://doi.org/10.1016/j.ijmedinf.2014.06.011.

[10] Electronic Health Records: Manual for Developing Countries.: 2006. *http://apps.who.int/iris/bitstream/10665/207504/1/9290612177_eng.pdf*. Accessed: 2017-05-13.

[11] Principles | Principles for Digital Development.: *https://digitalprinciples.org/principles/*. Accessed: 2018-05-30.

[12] Karen Holtzblatt, Jessamyn Burns Wendell and Shelley Wood. 2004. *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design*. Morgan Kaufmann.

[13] Mary Poppendieck and Tom Poppendieck. 2003. *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional.

[14] Ken Schwaber. 2004. *Agile project management with Scrum*. O’Reilly Media, Inc.

[15] Carol M. Barnum. 2010. *Usability Testing Essentials: Ready, Set...Test!*. Morgan Kaufmann.

[16] Alan Shalloway and James R. Trott. 2004. *Design Patterns Explained: A New Perspective on Object-Oriented Design*. Pearson Education.

[17] Dylan DeBoer. 2018. *Technical Communication Promotion Video*.

[18] Pam Estes Brewer. 2015. *International Virtual Teams: Engineering Global Success*. Wiley-IEEE Press.