Teaching Web Engineering using a Project Component

Daniel E. Krutz and Andrew Meneely
Rochester Institute of Technology
{dxkvse, axmvse}@rit.edu

***Abstract*—Web applications are an intricate part of the world today. Everything from banking to checking our Facebook status may now be done through the use of web applications. Todays students need to balance numerous concerns in order to create a web application that is robust, on time and on budget.**

At the Department of Software Engineering at the Rochester Institute of Technology, we created a course called *Web Engineering*. As part of this course, we developed an innovative project component which focused on students following software engineering principles such as elicitation, requirements generation, testing and deployment.

I. Introduction

Web applications represent a confluence of diverse technologies and numerous challenges. Some of which include networked environments, persistent storage, concurrency and usability. Web engineering is defined as the systematic, disciplined and quantifiable approach to development, operation and maintenance of web-based systems and applications [11] [7] [9]. While similar to software engineering, the concept of web engineering differs in several key areas [3]. The planning of continual growth and change has a higher significance in web applications [2].

Last year, the Software Engineering Department at the Rochester Institute of Technology (RIT) added a course entitled Web Engineering to their curriculum which is typically comprised of upper level 3rd through 5th year students. A signficant component to this course was a cross course collaborative effort with a focus on security. The cooperation is beneficial because it allows students to gain experience working with a an adjacent software team. Students will often collaborate in teams in industry, but are often unprepared to do so [6]. Additionally, focusing on security is valuable as web applications expose powerful technologies and assets to the Internet. Application security is an area which students and even workers in industry are typically deficient in [12] [4].

This project component is also distinct in the way it mimics a real world project as closely as possible. Students are not handed a firm list of requirements. They are expected to elicit, negotiate and comprehend changing requirements. This is an area that is extremely important for students have proficiency, but far too often lack [1] [10]. The project also utilized contemporary web technologies that allowed students to create a final product which they were actually interested in using and sharing with friends. This helped to foster student enthusiasm in the project.

In the following experience report, we describe the project as well as future improvements to be implemented in subsequent course offerings. Our goal is to allow other instructors to learn from our experiences and to be able to enact a similar project in their own web engineering courses at their own institutions.

II. Method

A significant aspect of our Web Engineering course was a project component. The main premise of the project was for each group to create a web application using both custom built and already existing components through web service and Application Programming Interface (API) calls while adhering to proper security standards for several vulnerability categories. Some of which included authentication, message encryption, authorization and session management.

The instructor took on two distinct roles for the project: teacher and customer. The way the customer reacts to student questions significantly differs depending on what role the instructor is currently playing. While representing the role of teacher, the instructor may give project advice and answer technical questions wherever possible. As the customer, they attempted to mimic a client in the real world and students were encouraged to clarify requirements with them. So students may understand which role the instructor is playing, students are encouraged to ask whenever they are unsure and begin their inquisitions with As the customer or As the teacher.

The goal of the project is to create a personalized web portal that would be customized for each user. The user initially logs in with their Facebook account. Once the user logged into the application, they are exposed to several pieces of personal, customizable information. One of the most significant is a section on the main page which is very similar to the wall in the traditional Facebook application. For this section, students were asked to again tie into the Facebook API to retrieve the necessary data. They were required to modify the appearance of these items and utilize aspects of usability covered in the course. Various other Facebook APIs such as photo albums, chatting with friends and status updates were used in similar ways. We selected these requirements not only because the Facebook API was readily available, but because we felt that incorporating Facebook here would help to encourage student interest in the project.

Several other aspects of the project required the students to write custom software to interact with extra data services or feeds. Students were asked to incorporate a stock viewing web page into their project. The user would initially enter in a stock that they mythically purchased along with the purchase price and number of shares. This information would be stored in a student created relational database. For all of these simulated purchased stocks, information would be retrieved from a third party web service and the page would be expected to display the current stock price, the day's high and low price, along with the amount of money the investment has thus far made or lost for the buyer. A chart is also displayed for the stock which is retrieved using an external feed of the groups choice. Other aspects of the application include a weather based component and a chat feature based upon HTML5. The reason for this chat component is to both familiarize the students with HTML5 and to acquaint them with how to properly place and utilize such an interactive element. In order to acclimate students to development environments like they would encounter in industry, several virtual machines were provided to each team. These were intended to act as development, staging and production environments.

During the ten week quarter, each team was expected to produce several deliverables. The first few weeks of the project aspect of the course focused on building up a base for understanding web engineering along with team formation. Teams of 4-6 students were created since this is often the size of groups in industry and has been found to be conducive to student learning in previous projects [5] [8]. Several roles exist on each team. These included team, development and testing coordinator. Since the course was comprised of upper level students, they were given the opportunity to self-appoint these roles. Students have indicated their satisfaction with this freedom. However, if the class was primarily made up of more novice students, the instructor may want to appoint team roles.

In the third week of the quarter, the students were asked to complete a requirements document and in the subsequent week, a design document. The expectation was laid out to each team that these were to be constantly evolving documents. The grading on these initial deliverables was not aimed at ensuring that the students had a completely accurate document on their first attempt. The main goal was for the students to have followed the proper guidelines for producing these deliverables and that an adequate effort was at least given to create them as accurately as possible. During the second half of each class session, teams were given the opportunity to meet with the instructor to ask requirement and general project questions. In these interactions, the students were also able to negotiate expectations with the customer. They were encouraged to show prototypes, screenshots and anything else they desired to the customer. The goal was not to limit customer interaction, or punish inquisitions as long as they were reasonable. The aim was to encourage customer interaction and elicitation.

The first software release was due in the sixth week of the quarter. Teams were asked to deliver a fully functional version of their application, but with only the functionality agreed upon with the customer for the first release. Less emphasis was placed upon appearance than functionality. Aspects such as mobile compatibility and cross browser support were not evaluated. Teams are also asked to provide updated requirements and design documentation, along with thorough test plans with the implemented tests. Some of which include unit and acceptance tests.

The Software Engineering Department at RIT places a large emphasis on public speaking, presentation and overall communication skills for their students. For the first release, each group was asked to give a 20 minute presentation about some of the major aspects and technologies used in their project. Other areas discussed were team roles and dynamics, a short demonstration of their application and their plan for the second release.

Immediately after the initial release, each group is asked to work on a team self-reflection document. Components of this paper include identifying areas of the project that went well, along with portions of the project which can be improved and how. Students are encouraged to deeply think and elaborate on these areas of went well and what can be improved.

Ensuring an adequate level of security is an important aspect of web based applications [4]. A week after the first release, each group is asked to release their application to a group in a concurrent software security course. The security course, entitled Engineering Secure Software, is a class designed to train students on the principles and practices incorporating security into the entire software development lifecycle. One of the class projects was the development of a web application fuzz testing tool (fuzzer), that automates the discovery of inputs and potential vulnerabilities in websites. Students would develop a set of scripts that would crawl a local website, discover the inputs, and then attempt to exploit those inputs using commonly-used attacks. Each fuzzer team was given a different web engineering product to fuzz, and was asked to report their fuzzing results to the web engineering team.

The second and final release occurs during the last week of the term and is conducted in a very similar fashion to the first release. The major difference is that the appearance and functionality are now both thoroughly evaluated. Additionally, applications are expected to be mobile device friendly. On the final day of class, each group again conducts a post mortem and investigates what went well and why along with what may be improved upon.

A goal of the project is to supply the groups with enough guidelines to provide them a solid direction, but allow them enough freedom in order to be creative. Additionally, the teams were encouraged to work with the customer to formulate extra features for the project which would be beneficial for the customer. This aspect was helpful in stimulating the students ingenuity for the project, working on their elicitation skills and in helping to add variability to each team's final product.

Future instructors are encouraged to deviate at moderate levels as they desire with the requirements for this project. These aberrations will not only keep the projects fresh and allow for freedom from both the instructor and the students, but will allow the instructor to explore and evaluate possible alternative paths for the project so it may be enhanced in future iterations.

III. Outcome

Before the beginning of the term, students expressed their excitement over the course and specifically for the project. They were interested in the real-world aspect of the project and how it interacted with contemporary technologies, tools and practices. Additionally, they were attracted to the freedom that the project structure would afford their teams.

At the conclusion of the course, an anonymous process was used to gather student feedback and was only made visible to the instructor after final grades had been submitted. Generally, the student feedback regarding both the course and project was positive. We feel that some of the reasons for dislike need to be addressed in upcoming course iterations. Other issues are ones which the students may not necessarily enjoy, but are essential for a proper student learning.

The students indicated that they felt the most beneficial learning aspect of the course was the project. Based upon this feedback, we believe that we are on the correct path with the project and feel it only needs tweaking in several areas. Student feedback also indicated several areas which they felt were beneficial. One of the most prevalent was the use of APIs and web services from groups such as Facebook and Google. They enjoyed using current and well known technologies for both their allure and practicality. Students also appreciated beginning the project with a reasonable list of requirements and not having to begin the elicitation phase from scratch. Other areas of positive feedback included the availability of the customer and the ability to self-appoint teams. The following are representative samples of written feedback we have received:

> I really like this project because it is giving us [software engineering students] experience with technologies that companies are truly looking for that without this class there was no formal way to learn. It was really interesting because it covered multiple aspects of web development from using certain frameworks, dealing with social aggregation, hosting our own chat service, and also learning about API's etc. Also it allowed us to see how rapid web development can be and how fast paced the field is
>
> After taking Web Engineering, I can confidently say: Why isn't there more of this class in our curriculum? As students living a web world, fast requirement shifts and one-click deployment are the norms for modern software vendors. Companies are now, more than ever before, looking for students with skills like JavaScript, Web Application Frameworks, and third party web API's. So far, this is the only class that has managed to capture the buzz that's ultimately here to stay.

IV. Conclusion

This paper presents some early findings regarding a project based component in a web engineering course. The primary areas of the project were discussed. These included the technical details, major deliverables and how the team was expected to interact with the customer. A main goal of this activity is to emulate a real world project situation as closely as possible. The role of the instructor acting as the customer was also conveyed. An interesting aspect of this project was the cross course collaboration which occurred with an adjacent security course. We will evolve and improve this collaboration in future course offerings.

While the course and project component generally went well, several areas can be improved upon. These include altering project requirements to aid in security testing and spending more time acclimating students with various technical concepts of web engineering. Future research will be done to discover how different technologies can be incorporated into the project in order make it both more appealing to the students, but more educational as well. This information will be gathered from instructor observations, official student feedback forms, and informal conversations with the students. We hope that our work in creating and refining a project based component can help others build a more educational and enjoyable web engineering Course for their students.

References

[1] Martin L. Barrett. Simulating requirements gathering. *SIGCSE Bull.*, 29(1):310–314, March 1997.

[2] Yogesh Deshpande, San Murugesan, Athula Ginige, Steve Hansen, Daniel Schwabe, Martin Gaedke, and Bebo White. Web engineering. *J. Web Eng.*, 1(1):3–17, October 2002.

[3] Athula Ginige. Web engineering: managing the complexity of web systems development. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, SEKE '02, pages 721–729, New York, NY, USA, 2002. ACM.

[4] William Bradley Glisson, Andrew McDonald, and Ray Welland. Web engineering security: a practitioner's perspective. In *Proceedings of the 6th international conference on Web engineering*, ICWE '06, pages 257–264, New York, NY, USA, 2006. ACM.

[5] Jiang Guo. Group projects in software engineering education. *J. Comput. Sci. Coll.*, 24(4):196–202, April 2009.

[6] Terhi Kilamo, Imed Hammouda, and Mohamed Amine Chatti. Teaching collaborative software development: a case study. In *Proceedings of the 2012 International Conference on Software Engineering*, ICSE 2012, pages 1165–1174, Piscataway, NJ, USA, 2012. IEEE Press.

[7] Emilia Mendes. Applying the cognitive flexibility theory to teaching web engineering. In *Proceedings of the fifth Australasian conference on Computing education - Volume 20*, ACE '03, pages 113–117, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[8] Dragutin Petkovic, Gary Thompson, and Rainer Todtenhoefer. Teaching practical software engineering and global software engineering: evaluation and comparison. *SIGCSE Bull.*, 38(3):294–298, June 2006.

[9] Gerald Reif and Harald Gall. Weesa - web engineering for semantic web applications, 2005.

[10] Asma Sajid, Ayesha Nayyar, and Athar Mohsin. Modern trends towards requirement elicitation. In *Proceedings of the 2010 National Software Engineering Conference*, NSEC '10, pages 9:1–9:10, New York, NY, USA, 2010. ACM.

[11] Till Schummer, Stephan Lukosch, and Joerg M. Haake. Teaching distributed software development with the project method. In *Proceedings of th 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!*, CSCL '05, pages 577–586. International Society of the Learning Sciences, 2005.

[12] James Walden. Integrating web application security into the it curriculum. In *Proceedings of the 9th ACM SIGITE conference on Information technology education*, SIGITE '08, pages 187–192, New York, NY, USA, 2008. ACM.