Introduction

Software engineering is required for the computer science major as well as the ISM major at Bridgewater College, this course is taking place in the fall of 2014. The goal of this course is to develop an application for a client using standard software engineering methods.

Teams

The professor divided every individual in class into teams. The teams were: Quality control team, Planning team, Application development team, and website development team. There was also a project manager who managed all the managers of every group. Each team were given responsibilities; web development team made the website and a supporting my SQL database, application development tem designed an android application and made sub-teams to help with their app development. The planning team planned the whole process including making diagrams (use case, sequence diagrams) and the quality control team researched a debugging software. Each manager and their team had to fill out weekly reports; every manager compiled its member's week reports (time logs) and gave them to the project lead. The project lead then compiled every manager's and their team time log and game them to the professor. The managers of every group divided tasks among its group members.

Project Description

We often want to share with others not only pictures and sounds from the places we've been but the actual route that we have traveled and location cues attached to the pictures and sounds that we record. In ornithology lab, Dr. Puffenbarger (a professor at Bridgewater College) drives a route through Rockingham County with numerous stops to check out birds so students can learn field identification. Several students wanted to practice the rout on their own, but wanted an electronic map rather than written directions or the poor copy of physical map. The students stated that they would prefer a phone app that allows them to save the weekly routes that they ran in lab ideally with pictures, short movies typed text and links to birds seen at different locations along the route. Our software engineering took this information and formed a preliminary analysis to tackle this project, which consisted of functional and non-functional requirements. Some of the functional requirements were the website allows the users to create accounts, the app can display a map with the user's location, users can create excursions, and the app can display on a map the route he user traveled during an excursion. Some of the non-functional requirements were the system is restricted to registered users, users may download excursions prior to loss of cellular connectivity and use excursion data in the field. When taking on this tasks from our client they expect the functional and non-functional requirements to be meet by 11:00 a.m. on December 3, 2014. To help this process go along smoothly we have set deadlines for ourselves so we could stay on track with the whole process; an requirements Analysis Document (RAD) was due on September 17, 2014. An initial prototype of the system was expected six weeks after the functional requirements have been established on October 29.

Weekly Lectures

Another section that was very important and critical to this class was early in the year was when Dr. McGregor gave our weekly lectures which showed us the process of how to organize our thoughts and to teach us how to get started on this application. Dividing it up into several sections Dr. McGregor showed everyone the guide to developing this application and the use of different models that we would need in the future of the class. We had roughly about seven different lectures that we were taught and the information taught in those lectures would be necessary to get everything going for the application. The first lecture touched based on the beginning steps we needed to know before we even got started with the project, he taught us about the terms and definitions that were essential to know before even going into process. The second lecture was based on Software Lifecycle, Requirements elicitation challenges, Problem statement, Requirements specification (RAD), Types of requirements, Validating requirements, UML, and Use cases Models. In this lecture we learned more about key things we would need in the upcoming classes. All of these terms were words that were so important going down the road. The next lecture Dr. McGregor talked to us about was going into client meeting with good question about the application, and getting rid of the questions that we already knew the answer too. Following that lecture was an lecture on Finding Use Cases, Flow of Events, Use Case Associations, and Use Case Refinement, this would also be very important down the stretch. The fifth lecture was on the analysis, and object modeling which main goal was to find the most important abstractions of the model. Next we had a lecture on Dynamic Modeling or you could say the Sequence Diagram and how to go about doing sequence Diagrams and making them. This was very difficult at first but after you learned how to do it, it wasn't too bad. We used the Sequence Diagram to show how the application would be executed. The last lecture was about testing the application when it was completed; Dr. McGregor taught us that testing was the key to having a successful application. But not just with applications with anything you do, always go back and test your experiment, or application to see if it work and makes sure everything is good before you give it to your client.

Requirement elicitation

After working on the project for an extended period of time there was a client meeting that was held for the managers to attend. Prior to these client meetings each team went over the functional and non-functional requirements and brainstormed questions to ask the client. These questions were pertaining to something they would possibly want added to the application or website. Or questions pertaining to things that we felt were not necessary for the completion of the application and/or website. During these meetings the client also expressed what changes they wanted, while the managers expressed what could or could not be done by the deadline.

After the application and web development team had made significant process a demo was held for the client so they could get a visual on what progress was made.

Requirements Analysis

At the beginning of the semester Dr. McGregor gave us a Problem Statement, and in this Problem Statement was all of the information the Client had asked for to be in the application. We looked at the Problem Statement and the first thing that struck out to us was the functional and non-functional requirements. They were two of the most important things in this Problem Statement because what is an application without the functional and non-functional requirement. As a class we listed what we should focus on first and what we could try and get done in the short time period that we had. Once we got done discussing, we was to try and find a way to make this look as professional as possible so we decided that the first thing we should developed was a RAD document which stands for Requirement Analysis Document. In this document we put everything that was needed to explain the application and its abilities and non-abilities. We started off with the name of application and our logo. Our class as whole picked out the name of the application and the logo. Following the name and logo was our Table of Contents page showing the clients were everything was located and what pages they were to be found on. Next after the Table of Contents was our Introduction, which is where we explained the use for the RAD document. Preceding in the RAD document was the Proposed System which explained to our clients and viewers what the purpose of the application was, and what the application was for. The functional and non-functional requirements for the application were followed. We also put together a website so of course we thought we should tell about the functional and non-functional requirements for the web site too. The Sequence diagram, the Use Cases Models, and User interface which shows examples of what the application looked like, and a short glossary to define terms so people could understand what each word meant were followed. The RAD documents were given to our client so that they might would understand the application better and know were to find the functional and non-functional requirements for the application. The RAD documents wasn't only designed for our clients but anybody that wanted to look at our application and see what the application was for and if they liked what they read in the document they could be knowledgeable of what the application was about.

Software Models

We created models to help organize the various steps needed to create and organize the functionality of the app. These models include sequence diagrams, case diagrams, as well as the RAD Document. Each of these models are very important to the structure and function of the application that we have created. For example the case model shows how every part of the application is connected and what the user sees.

Web Site Development

The hosting platform used for this application was Microsoft Azure. Microsoft Azure is a free online website that allows developers to host a database and web source for a 6 months free trial. The program is compatible with IE, Chrome, Firefox, Safari. The software development tool that was essential for the development of the webpage was HTML 5 Markup. HTML 5 Markup is a free online template used for the design of the webpage. It can be found at http://www.html5up.net. The template chosen for the class' website was Arcana. HTML 5 Markup provides multiple templates to fit your taste of style and color scheme.

The Web Development Team needed to learn how to code a website using HTML and PHP. Web Matrix 3 provided a template of PHP and HTML code as a basis for creating web page and then which was edited by the developers to reflect the necessities of the webpage. This web development platform communicates with Microsoft Azure by allowing the code to be hosted on the temporary server.

The Programming languages used in the class were PHP and HTML. PHP (hypertext preprocessor) is scripting language that allows the user to enter user data into the website and the information to then be stored in a database and can communicate between PHP and HTML. HTML (hypertext markup language) is coding language that the computer understands that allows you to make banners, headings and allows you to order the web page however you wish. CSS (cascading style sheet) communicates more detail to the website and allows you to style the web page.

The website of our application contains multiple pages. They include:

Index (home page): allows the user to download the app, includes the name of the app, and has links to external websites that may be useful to anyone using the app

About: general summary of the developers which includes the developers (app, web, planning, quality control) image of the class and includes the description of the class goal, includes documents link

Documents [within the About page]: RAD document (describes the system in terms of functional and nonfunctional requirements and serves as a contractual basis between the client and the developer), Narrative (story of the process of achieving the goal), Android-Side SQLite EER Diagram, Android to Web to PHP Protocol, Application Server EER Diagram, Object Model (is the blue-print of the system and includes class diagrams, relationships between these classes, methods in the classes, properties. Etc.), Requirements Analysis (process of determining user expectation for a new product and is very detailed), Software Engineering Team Structure (gives details about each team, the managers in charge of those teams and the students who are part of those teams, in a visual diagram)

Help: contains the tutorial on navigation of the app (logging in, loading excursion, adding observations, editing excursions, registering for an account, start recording routes, viewing observations, saving excursions, main screen, stop recording routes, editing observations, logging out)

Downloads: includes all versions of the app. Each version is listed with the corrections made and explanation as to why the correction was made

The opportunities experienced during the website developed was the opportunity to learn how to HTML and PHP code. It allowed the Web Development Team to go in depth with their previous knowledge of HTML and build from a foundation that allowed them to create something more than the team thought possible. Some of the challenges experienced during the process was actually understanding the code and learning how specific coding can be. A lot of the team members had to overcome barriers. Research helped the web development team overcome those barriers.

Android App Development

The android team worked very hard to create the android application. There were various steps that the team had to take and those steps include: software development and processes, GUI development, implementing google maps, database design, database adapter development, application server development, and there are many opportunities and challenges that are associated with creating this application. The app. Team used a couple of tools to aid their work in software development. They used similar software to eclipse, which is called ADT. ADT allowed the app. Team to emulate the phone using the computer, so they did not have to work straight from the phone. The limitations with this software are that they cannot utilize the location features that the phone has. They also used law cat, DDMS and Vizio. The GUI development was a lot of fun according to the app. Team, but it required a lot of code. The gui system entitles the user interface and how the draw menu looks and operates. The system had many changes due to functionality but also to satisfy what the clients wants. Google maps was a hard part for the app. Team because none of the members had really worked with google maps prior to the creation of the app. It was a big learning experience for them and a lot of what they implemented had to be learned because google has their own code for google maps. The plus side of using google maps is that there are many features. The software is very versatile. Database design was also very hard for the app. Team. They had trouble with packaging their information and sending it to the database through the internet protocol. This aspect of the application is very important because everything is connected to the database. The server development was a learning experience for the team as well. The application server is the storage portion of the application, so it was very important that the team to complete this part of the app. One the biggest challenges that the team faced was managing their time. There was a lot of information for them to learn and their always was new content that they had to understand in order to move forward with the creation of the application. The application team feels that the app. has potential to start off at BC and will do well. But they hope that it will be able to be used publically and even be made open source for the public to use.

Quality Control

The Quality controlled team was put together by Dr. McGregor to test the application every time a new version was created. The Quality control team consisted of three people, their manager being Dylan McGraw and the workers under her being Paul Rachner and Brandon Larsen. The Quality Control team was made to get a bug tracking software able to read the code of the application and able to go in and show where we was having problems with the software. They did some pretty extensive research online to try and find a good but not so expensive bug tracking software and they found one. The first bug tracking software they found was BugZilla but the requirements of the application were a little bit too much for what we were looking for in the class. So they found another software available that was named Axosoft bug tracking they was pretty good and in our budget to get so they chose that bug tracking software. Quality Control also came up with a testing plan and the plan was to test the application on a weekly basis of the new versions that the Development team would put out. The major challenges that the Quality Control team faced was the creation of their test plan, there were a lot small things that they needed to capture in their plan, a lot of circumstances that they need to make sure they covered so that all of the bugs in the software were managed and found. The Quality team makes sure that all software is running smoothly and catching all of the bugs that are in our application.

Administration and Planning

Another part of the project was to find out who actually owned the application. We went through several theories which included getting Bridgewater College's IT center to host our website and application. We also considered using an online hosting website so that the ownership would go to all members of the project. We looked at GoDaddy and Ipage for our outside webhosting servers. Both websites were great candidates for what we are trying to accomplish, but we have yet to choose between the IT center and the outside sources. We soon found out that because the application is being made in a class setting Bridgewater would still have ownership of the application. Through research and meetings we came to the conclusion that Bridgewater College would have ownership almost indefinitely. As we continue to go through the project an important piece was missing and that is the project narrative. The project narrative consists of all the steps we took within the project. It will go up on the website and everyone who visits the website will be able to read it. It will give readers insight on how we conducted this project and will give them enough information to potentially spark an idea from our experiences.

Final Client Demo

The clients at our final client meeting was very grateful of our application. They expressed great remarks regarding how excited they were to start using the application. The application will be used for the first time in the spring semester 2015.

Conclusion

Taking on this very tedious task helped every individual in software engineering in many ways; it helped everyone learn how to work in a business environment with the various documents that we had to analyze and fill out (time logs). It helped the individuals meet deadlines in a business setting while also dealing with changes that the client wanted.

Overall the project stretched the abilities and talents of those who were involved. The application created experience, knowledge and perseverance that is needed for the real world.