Virtual Classroom

Project Management Plan

COP 4331C, Fall, 2015

Modification History

Version	Date	Who	Comment
V0.0	09/8/2015	J. Bender	Initial Draft
V0.1	09/12/2015	J. Casserino	Updated Rough Draft Added; • Project Overview • Reference Documents • Applicable Standards • Project Team Organization • Deliverables • Technical Progress Metrics
V0.2	09/12/2015	J. Bender J. Casserino	Updated Rough Draft (Casserino) Added; • Tools and Computing Environment (Bender) • Configuration Management (Bender)
V0.3	09/13/2015	J. Casserino M. Friedman	Updated Rough Draft (Casserino) Added; • Plans for Tracking, Control, and Report of Progress (Friedman) • Table of Work Packages, Time Estimates, and Assignments (Friedman) • Quality Assurance (Friedman)
V0.4	09/16/2015	C. Armstrong J. Bender J. Casserino M. Friedman	Updated Rough Draft Added; • Risk Management • PERT
V1.0	09/18/2015	J. Casserino	Finished Checking Rough And Corrected Minor Grammar Errors

Team Name: Group 26

Team Members:

Joseph Bender	jbender94@knights.ucf.edu
Joshua Casserino	Joshua.casserino@knights.ucf.edu
Chad Armstrong	chad.armstro@knights.ucf.edu
Miles Friedman	milesfriedmanfl@gmail.com

Contents of this Document

Project Overview

The Virtual Classroom App is an iPhone app that will improve the current learning management system, Canvas. The system's main goal is to increase the availability of information the students have directly from the instructors, to improve the data integrity; and to increase the information the instructors have about the current level of understanding of the students. The secondary goal is to improve the current system's ease of use for all users. With a direct line of communication between the instructor and students in real time, the instructor will be able to cover sections more thoroughly. The real time lecture feed will also lessen the damages of out of school emergencies, which will plague everyone at least once in their academic career. With these improvements to an already successful system, the Virtual Classroom App will only improve the education of students and increase the proficiency of the instructors.

Reference Documents

- *Concept of Operations -* https://drive.google.com/open?id=0By95Fo5D2Zi8ZU9kcmQ3SmJtMU0
- *Project Management Plan template -* http://www.cs.ucf.edu/courses/eel5881/ProjectPlanTemplate.html#Overview

Applicable Standards

- Coding Standard – https://google.github.io/styleguide/javaguide.html
- Documentation Standard – https://owl.english.purdue.edu/owl/section/2/10/

- Artifact Size Metric Standard – Hours taken to complete the listed features in the https://drive.google.com/open?id=0By95Fo5D2Zi8ZU9kcmQ3SmJtMU0

Project Team Organization

Group 26 is comprised of four University of Central Florida (UCF) Undergraduate students in the COP4331 (Processes for Object-Oriented Software Development) class. The group members are Chad Armstrong, Joseph Bender, Joshua Casserino and Miles Friedman. Currently, all team members share equal responsibility for the design and documentation. All members will work on the code. Joseph Bender will be the lead project manager, Chad Armstrong will take the lead on the database design and management, Miles Friedman will take the lead on quantity assurance and testing, and Joshua Casserino will take the lead on documentation and record keeping. Each aspect of the project will be a product of group effort; no individual will solely create any feature or document.

Communication is key to the design and manufacture of a great product so the group decided to meet at least weekly on Tuesdays, and at any time required due to issues that may arise. Aside from face to face meetings the group is using a number of communication resources such as, but not limited to; GroupMe (phone app), mobile phones, GitHub, and Google Drive.

Deliverables

Artifact	**Due Dates**
Meeting Minutes	Within 48 hours after meeting
Individual Logs	Updated Weekly on Fridays through WebCourses
Group Project Management Reports	
Concept of Operation	09/17/2015
Project Plans	09/17/2015
SRS	10/08/2015
High-Level Design	10/29/2015

Test Plan	10/29/2015
User’s Manual	12/03/2015
Final Test Results	12/03/2015
Source, Executable, Build Instructions	12/03/2015
Project Legacy	

Tools and Computing Environment

The mobile application will be developed for the iOS platform. This will require the use of Xcode 6.4 on Mac OS to develop the app. Note: other third party cross-platform development tools such as Xamarin will be used as well (for developers without Xcode). The project will be programmed using the swift (objective-c based) programming language for iOS. Xcode’s standard iOS compiler will be utilized to produce builds for application testing. The project will utilize an open source version of iOS Canvas from Instructure called CanvasKit available on GitHub. This will provide us with necessary libraries and references to Canvas API endpoints. Libraries included are CanvasKit.h and Constants.h. These provide necessary functions useful to integrating with the Canvas web application functionalities.

Configuration Management

The group will utilize GitHub as a version control management system. This will allow the tracking of code updates. It will also help to provide metric data on our progress in terms of features completed (lines of code). Joe Bender is responsible for the master repository which can be accessed by all of the group members to support collaborative development. When the team is to do development, or needs to add files they will be pushed to Joe Bender’s repository under the directory VirtualClassroom on GitHub. The location will contain all code files, database files, libraries, readme documents, implementation documentation, and software documentation. Team members can follow the simple guidelines to push and pull code to and from the repository without the need for approval. The VirtualClassroom directory is shared to the team members' accounts for push and pull rights.

Quality Assurance

During our planning phase we will use UML diagrams to divide our mobile application in to a set of core functionalities. After any of these core functionalities is completed, we will be testing the feature to make sure it works as we expect it to. For each additionally added feature, we plan to both test that feature, and all of the already added features that interact with it. It is our hope that this way, we may catch bugs early and keep the development process as smooth and seamless as possible. Once the app is complete, we will go through a final phase where we will test the core functionalities of the application before submitting the final version.

The Project Manager: Joe Bender, will be responsible for making sure that we are all doing our share of testing. Since our phone will run on iOS and not android, testing will be done using the iPhones of Joe Bender and Miles Friedman, which means that Miles and Joe will most likely be responsible for most of the QA. We are currently trying to get our hands on an IPad as well which would be used as an additional test device.

When testing a feature we will write up a list of things that should happen based on a certain action, and then confirm that the desired behavior occurs. This will likely be done using third party issue tracking software which we will also be using for bug tracking. If there is an issue, we will report the bug and decide whether or not it needs to be fixed at the moment, or can be fixed at a later time.

Risk Management

Application Design Out Of Scope:

If our design is out scope for our allotted time then we would have to redesign our project that is more manageable for this class. This risk will be minimized by the time we allotted for testing, and also the time we set aside for implementation is long enough for a redesign if needed.

Application Feature Failure:

If we want to implement a feature of the original design of the application but are having trouble implementing that feature in the time we have, then we could just not implement that feature. This risk is caused by our general lack of experience with working at such an in depth level in app design, so as we learn more about it we can realize what is feasible in the time we have for this application. This risk is minimized by our long amount of time for implementing the applications code and features.

Collaboration of Application Feature Failure:

After we design and code the features, database, and server for this application, there is a risk that combining all these facets could cause a failure. This risk is applicable since this will be the first time our group implements an application like this. This risk is minimized by our long testing period.

Database Failure:

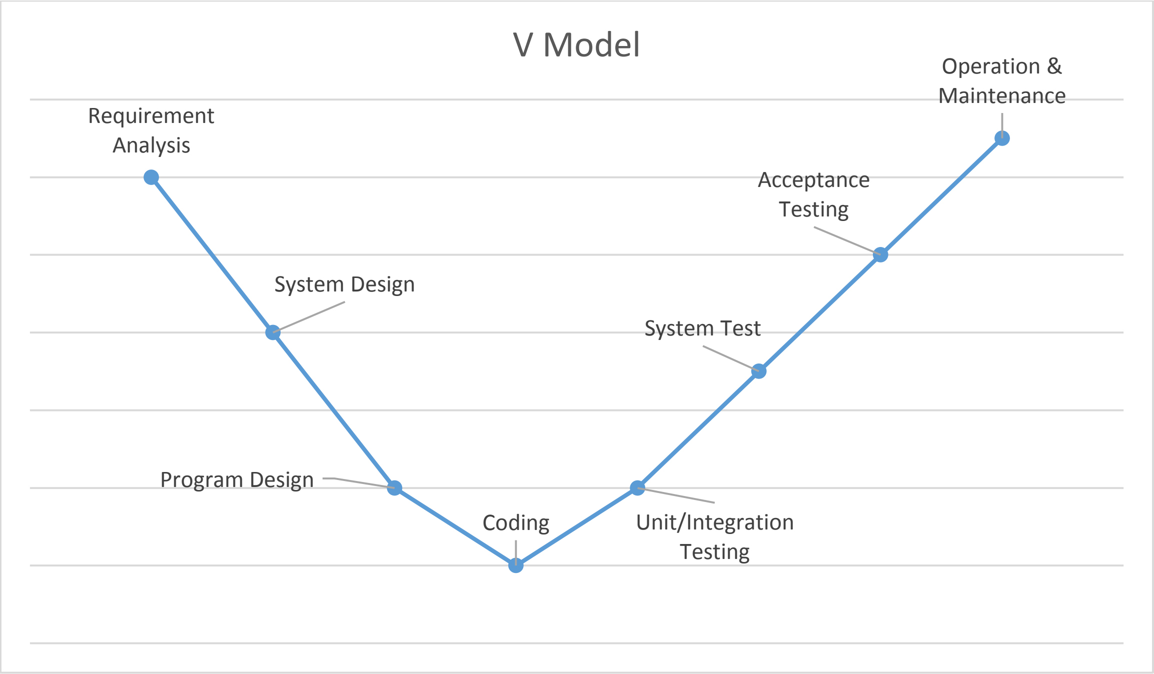
If there are problems with strictly our database implementation we could seek help from one of the many qualified professors to help us with this, or also we could ask our TA for this class. This risk is minimized by the long period of time allotted for database implementation and also the long testing period that is planned to be at the end of the implementation period.

Server Failure:

Similar to a database failure, a shortcoming with the server can be prevented with seeking additional help by either our TA or a professor who could be of assistance. This risk is minimized by the long period of time allotted to implement the server and also the long period of time allotted for testing.

Demo Failure:

The possibility of our demo going poorly can easily be avoided by pre-recording our demo. This can only be done if we plan to have time to record the video, which means finishing the project earlier and recording the presentation we would have to practice beforehand while testing.



V Model

1. Requirement Analysis

   a. Decide what the functionality of the project and what we want our product to do.

2. System Design
   a. Decide how to best to implement the features we want and draw up what our product will do and how it will do it.
3. Program Design
   a. Decide how to implement and code the project, including classes and how to implement each feature.
4. Code
   a. Implement the design that was previously drawn up.
5. Unit & Integration Testing
   a. During and after coding each feature test it concurrently with the rest of the system.
6. System Testing
   a. After testing each feature by itself and with the system, ensure the design follows the plan.
7. Acceptance Testing
   a. Test finished project with what our original product design would be, trying to bridge any gaps in the two designs.
8. Operation & Maintenance
   a. Collaborate and test product under stressful conditions and maintain operation through testing.

Table of Work Packages, Time Estimates, and Assignments

Requirement Analysis

In one week, all team members will have analyzed the requirements and will be aware of the required process we must follow throughout this Project. During this package we will address the following questions: How will we report our activity throughout this process? How will we handle version control, bug tracking etc.? What does our application need to accomplish from a user perspective?

System/Program Design

This package should take us two weeks. Joe Bender, as our project manager will be in charge of ensuring that the system design is completed. But all members of the group will participate in the design of the application in terms of layout and how it should be coded. Questions we will address during this package: Do we need to use a server? (And how?) Do we need to use a database? (And how?) What do we need to accomplish from a coding perspective to accomplish our desired goal?

Coding

All members will be responsible for the completion of the coding portion of our project. More specifically, areas of coding will be divvied up by our Project Manager during the System/Program Design phase and those assigned to a specific task will be responsible for the accompanying section of code. We estimate four weeks of coding.

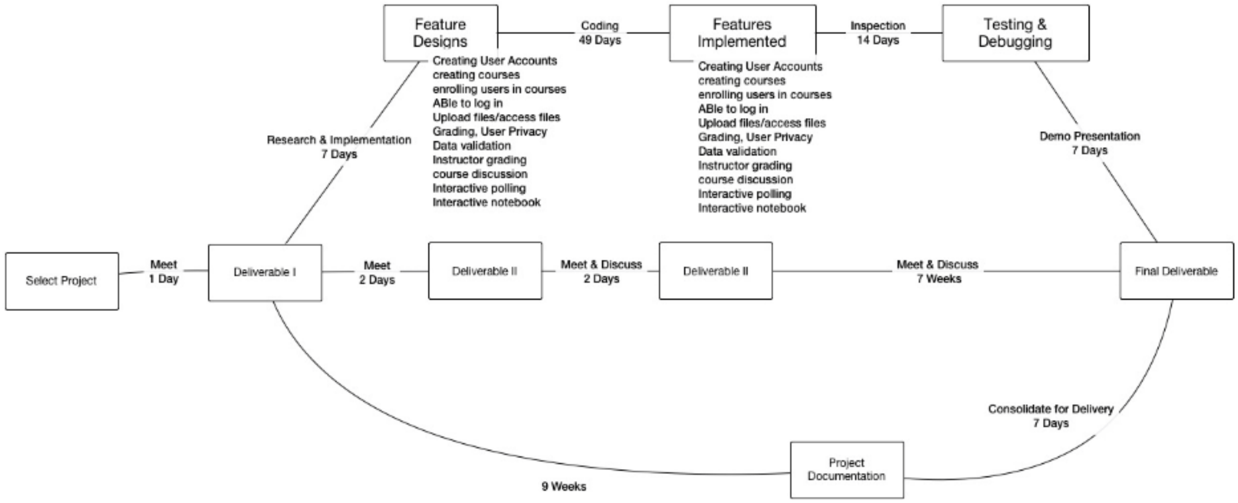
System Testing

This will be performed by each member of the group. By our estimations, about two weeks of testing will be needed to ensure that everything works appropriately.

Acceptance Testing

As of now, Joe Bender and Miles Friedman will be largely responsible for the acceptance testing as they are the only members of the group with IPhones. If we are able to gain access to an IPad or an additional phone, more members will be able to contribute here. This should take us two weeks and will be the final step.

PERT



Technical Progress Metrics

Work Package	Metrics	Estimate	Actuals
Meetings	Hrs.	16	2
Individual Logs	Hrs.	8	1
Deliverable I Documents	Days Required Sections	1 14	4 11
Deliverable II Documents	Days Required Sections	2 9	0 0
Deliverable III Documents	Days Required Sections	2 5	0 0
Creating user accounts and profiles	Days Percentage	70 100	0 0

Creating course(s)	Days Percentage	70 100	0 0
Enrolling users in course(s)	Days Percentage	70 100	0 0
Users being able to log in	Days Percentage	70 100	0 0
Users being able to upload/access course files (within profile limits)	Days Percentage	70 100	0 0
Privacy of grading, and personal data	Days Percentage	70 100	0 0
Data validation, stream	Days Percentage	70 100	0 0
Instructor being able to make/grade assignments	Days Percentage	70 100	0 0
Course discussion section	Days Percentage	70 100	0 0
Instructor being able to create an interactive poll	Days Percentage	70 100	0 0
Data validation, stream	Days Percentage	70 100	0 0
Students being able to create an interactive notebook (Private and Public)	Days Percentage	70 100	0 0
Final Deliverable Documents	Days Required Sections	3 5	0 0

Plans for Tracking, Control, and Report of Progress

At a minimum, each member of the team will post the following weekly: An individual activity log, Individual status information and weekly goal status. Using an as needed basis, bug/issue tracking and test plans (QA) will be recorded through a third party issue tracking software called LeanTesting. Version control and weekly reporting will be handled using GitHub.

Each week, all members will get together and set weekly goals. These may include coding goals, bug fixes, or anything else relevant to the project. During our weekly meeting we will also review each member’s weekly report and discuss progress towards goals from the prior week.