COYOTE
QUEST

Software Project Management Plan

Project Manage/Software Engineer: Eugene Kim
Assistant Project Manager/Software Engineer: Nicholas Perez
Software Engineers: Gan Liu, Jasmine Pena, & Daniel Martinez

1. Overview

1.1 Project summary

The Software Project Management Plan layouts the details of the management plan that will be followed in order to develop the CoyoteQuest application. It includes the development cycle, organization, specific roles, projected timelines and testing protocols.

1.1.1 Purpose, scope, and objectives

The Software Project Management Plan (SPMP) will relay all the details regarding the development plan and the development cycle. It will assign each member roles and duties regarding the development, and how and what methods they will use in order to finish their tasks. It will also assign deadlines so that project will developed on time. It will follow the specifications declared in the most current version of the SRS that was signed by the client. The first prototype will contain the UI design. It will be able to navigate through different pages, contain the logo and the remaining texts according to the client. The reason for developing the CoyoteQuest app was the need for a dynamic map to help locate and route students, faculty, or staff to their destinations. Using the app would be more efficient than reading from a static map that cannot locate one's location with respect to another point. Finding points of interests such as school campus amenities can appear on a map at the touch of a button. Therefore, CoyoteQuest would be the solution to guiding people's destination routes without guessing one's initial location.

1.1.2 Assumptions and constraints

The list of all the assumptions and constraints

- Team members will attend all meetings
- Team members will meet all the deadlines
- Team members will follow the requirements specified in SRS
- Development of the application requires Android OS with an API of 19 or above ● Team members will work on the project outside the class to finish it on time

1.1.3 Project deliverables

- Publishable Application on Google Play Store
- SRS, SPMP, Documented Source Code, Software Architecture, Maintenance Manual

1.1.4 Schedule and budget summary

No budget scheduled for this project but the team will be efficient as much as possible as if we had a budget.

1.2 Evolution of the plan

The whole team met with the client on April 15, 2019, and discussed the expectations of the applications and its functionality. From there, we made plans to design and implement new features on top of the existing code. Because this project is a continuation of an older project, the team must update the existing code in order for it to run on an upto-date Android Studio developer tool.

2. References

Android Studio
https://developer.android.com/index.html

IEEE SRS Template 1058-1998
http://www.cse.msu.edu

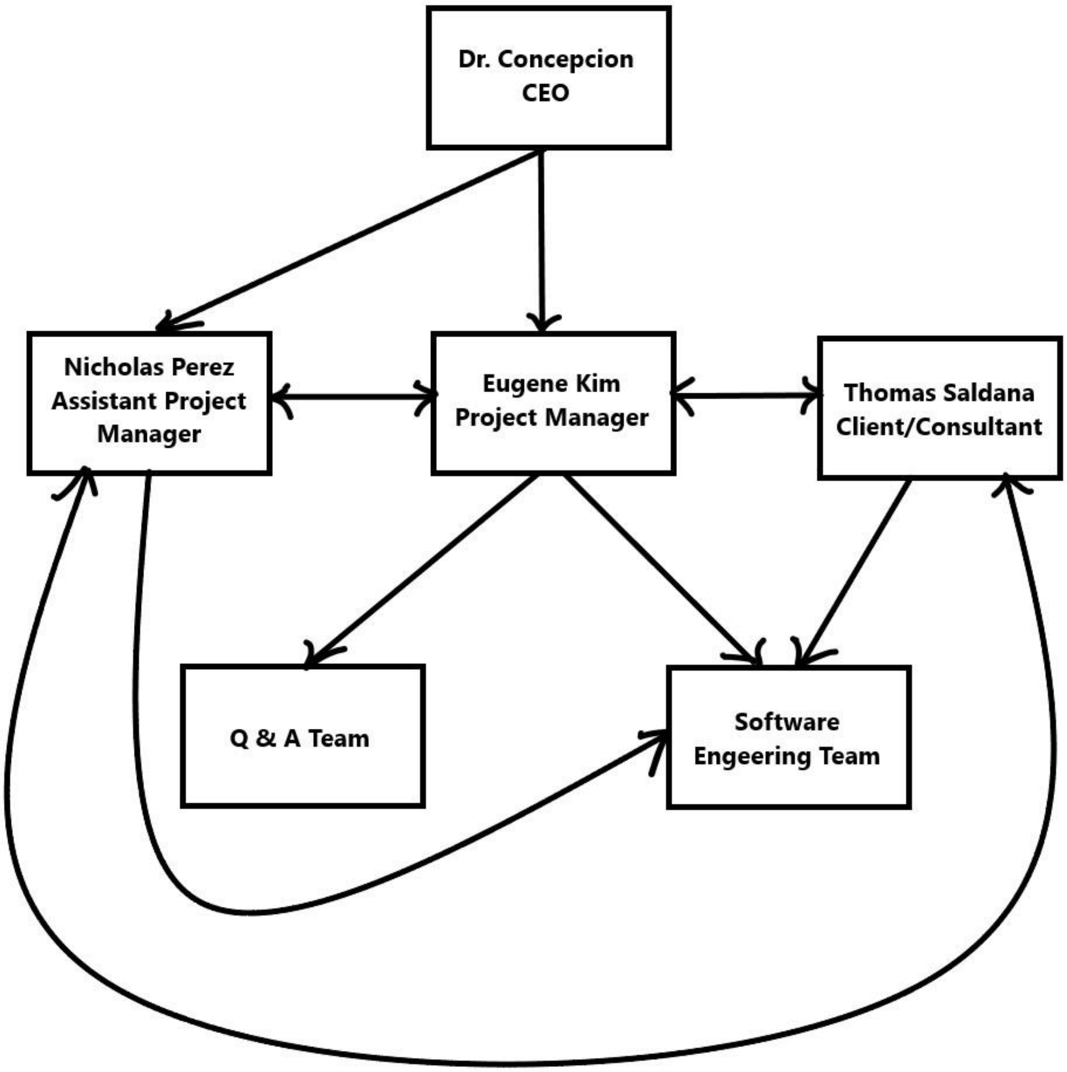
UMVELT Software Project Management Plan
https://mobileappdev.academic.csusb.edu/wp-content/uploads/2019/04/Software-ProjectManagement-Plan.pdf

3. Definitions

CSUSB: California State University of San Bernardino
Mobile App: Mobile Application - Application operating on small device
UI: User Interface - the means by which a user and system interact
GUI: Graphical User Interface - method used to mediate relations between user and device through visual representations and text.
Android: an open-source operating system used for smartphones and tablet computers.
GIS: Geographic Information System – a system designed to capture, store, manipulate, analyze, manage, and present spatial or geographic data.
ArcGIS: a geographic information system for working with maps and geographic information developed by ESRI.
API: Application Programming Interface – refers to a set of routines, protocols, and tools that is used in the building of software applications.
Geocoding: the process of transforming a description of a location—such as a pair of coordinates, an address, or a name of a place—to a location on the earth's surface.
Navigation Drawer: Destination routes for the user to choose and follow a suggested path.
Grid Tools: A list of amenities that the university campus provides mapped on the interface.

4. Project organization

4.1 External interfaces



Dr. Concepcion (CEO) - Monitors performance of the development team and provides guidance to the project managers.

Eugene Kim (Project Manager) - Fosters communication between the client and the team. Uses communication resources such as email and basecamp to communicate between the different groups associated to the development team. He also has direct involvement in the programming of the application.

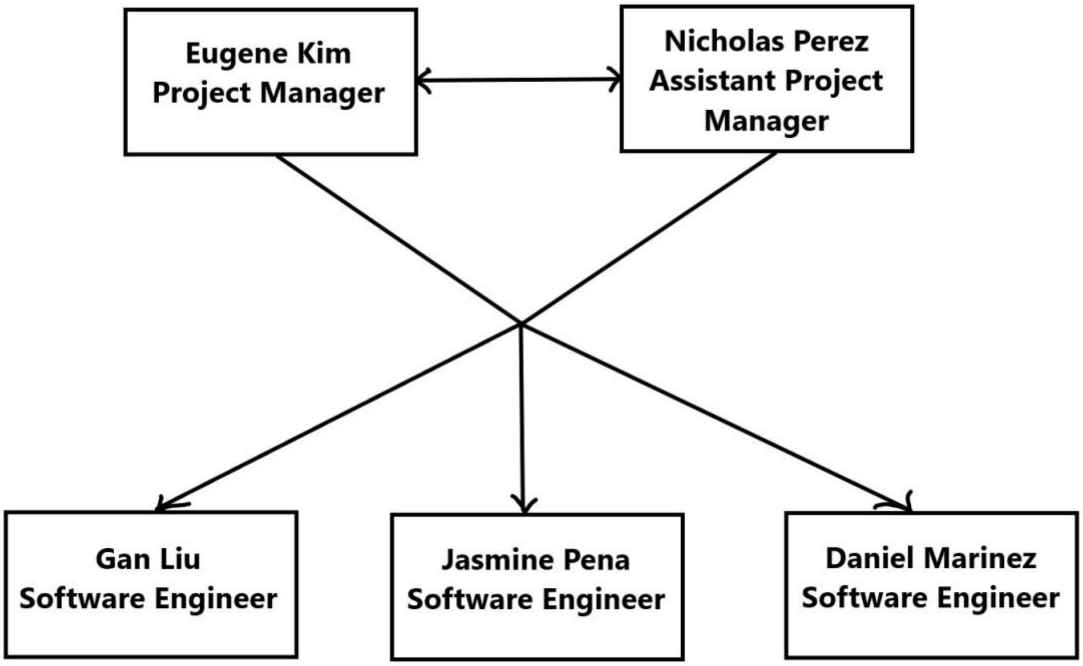
Nicholas Perez (Assistant Project Manager) - Monitors team progress in accomplishing development goals and manages documentation resources. In the event of the project manager's absence, APM can take over duties and responsibilities associated with the Project Manager. He also has direct involvement in the programming of the application.

Thomas Saldana (Client/Consultant) - Communicates to Project manager and the Assistant Project Manager the requirements of the mobile app and gives feedback on presented prototypes. Additionally, he provides any direct assistance to the software engineering team.

Software Engineering Team- Codes the required features of the app

QA Team - Evaluates prototypes as they are made available and tests for worst case scenarios.

4.2 Internal structure



The project manager and assistant project manager work together to facilitate progress in completing the prototypes.

4.3 Roles and responsibilities

Eugene Kim (Project Manager) - Communicates to the team what the client desires in the mobile app.

Nicholas Perez (Assistant Project Manager) - Works with PM and the client to complete prototype development. Responsible for geocoding locations of the application.

Gan Liu (Software Engineer) – Responsible for geocoding locations of the application.

Daniel Martinez (Software Engineer) – In charge of ArcGIS map image layer URL

Jasmine Pena (Software Engineer) – codes for the Navigation Drawer

5. Managerial process plans

5.1 Start-up plan

5.1.1 Estimation plan

- Discuss with the client all the specifications for the application.
- Study and research the existing code to integrate new features on the application
- Keep tabs on the progress of the whole team
- Create guidelines for working prototypes 1 and 2 including deadlines

5.1.2 Staffing plan

Project staffing based was based upon survey responses that every member of the class partook. The professor and teacher assistants helped to evenly organize the teams according to the survey responses.

5.1.3 Resource acquisition plan

All the software tools involved in the development of the application are free of charge with the exception of a publication fee to upload the app onto Google Play Store. Additionally, there are computer workstations on campus grounds to assist members in developing the application.

5.1.4 Project staff training plan

All the group members will complete the tutorials given the first three weeks of the class. Specific to this team, each member is required to learn the existing project code as well as methods to implement the planned new feature.

5.2 Work plan

5.2.1 Work activities

Work Units:

- Navigation Drawer Development Prototype 1: The first prototype will contain a working Navigation Drawer which would be developed by the whole team.
- Prototype 2: In this version of the application, the first Prototype will be integrated with Google Maps, and receive data from that application. The application will also be able to use the tutorial screen where pressing an arrow button will relay a vibration pattern for that instruction. The entire team will work on it.
- Documentation: All members will make sure that all the code that they write is properly documented.

5.2.2 Schedule allocation

4/15/19	4/19/19	Prototype #1 Development (Navigation Drawer Feature)
4/22/19	4/26/19	Prototype #1 Development (Floor Buttons Feature)
4/29/19	5/3/19	Prototype #1 Development (Grid Buttons Feature)
5/6/19	5/10/19	Prototype #1 Unit, Integration, & Systems Testing
5/13/19	5/17/19	Delivery of Prototype #1 & Prototype #2 Development (Search Feature)
5/20/19	2/24/28	Prototype #2 Development (Layers Feature)
5/27/19	5/31/19	Prototype #2 Development (Navigation & Routing Feature)

6/3/19	6/7/19	Prototype #2 Unit, Integration, & Systems Testing
6/10/19	6/14/19	Delivery of Prototype #2, Exhibit Presentation, & Demo

5.2.3 Resource allocation

Each member has access to the same resources as all the other members. The resources include Android Studio, and the workstations available for all the students in Jack Brown.

5.2.4 Budget allocation

No budget has been allocated for this project.

5.3 Control plan

5.3.1 Requirements control plan

Each member of the group is required to attend the meetings in the class. Additionally, each member is required to document their code, follow the guidelines decided in the SRS and meet each deadline as well. Any unexpected issues, technical difficulties or requests by the clients will be assessed by the Managing team and decided upon.

5.3.2 Schedule control plan

Aside from the regular class meeting time, the team might be required to meet outside the class as well in order to finish and deliver the product on time. Managers will make sure that each member is completing their tasks on time.

5.3.3 Budget control plan

There is no budget allocated to the CoyoteQuest team. However, we will make sure that the project is efficient enough so that it doesn’t overtax the budget.

5.3.4 Quality control plan

The managing and the developing team will constantly perform a quality check on the software at least once a week to make sure that the project meets all the expectations. Also the client will be present during the development of the application and therefore will be informed of the quality of the application.

5.3.5 Reporting plan

The managing team will notify Dr. Concepcion of the bi weekly meetings. The project manager will generally send him the attendance of the group members and make sure that the application is up to date.

5.3.6 Metrics collection plan

The managing team will make sure that each week every developer completes their tasks so that the project is on track and progressing as expected. The managing team will make sure that the code is efficient and meeting all the standards.

Productivity & Quality Measure:

Productivity Metric: 30.6 lines of code per hour
Quality Metric: 200 faults per 1000 lines of code

Jasmine: Detailed video conferencing with consultant regarding Navigation Drawer feature and step-by-step instruction on implementing ~150 lines of code.

Daniel: Spent ~2 hours per week, in addition to meeting and lab hours, on development, especially ArcGIS SDK implementation and API, and implemented ~100 lines of code.

Nicholas: Spent extensive time on researching & debugging of existing code, and implemented ~100 lines of code.

Gan: Spent extensive time researching the ArcGIS SDK regarding geocoding an address for the search feature being implemented and coded ~100 lines of code.

Eugene: Spent extensive time on researching the ArcGIS SDK regarding navigation and routing feature, tutorials on implementing the grid buttons feature, and coded ~100 lines of code.

5.4 Risk management plan

Development

- The team will meet regularly to make sure that the production is not stopped at one particular point.
- There will be scheduled deadlines that everyone will be following to make sure the project is not delayed
- Each team member will be kept up to date, and inform managing team of any changes or difficulties that might affect their ability to complete the task they are given on time.

Project Failure

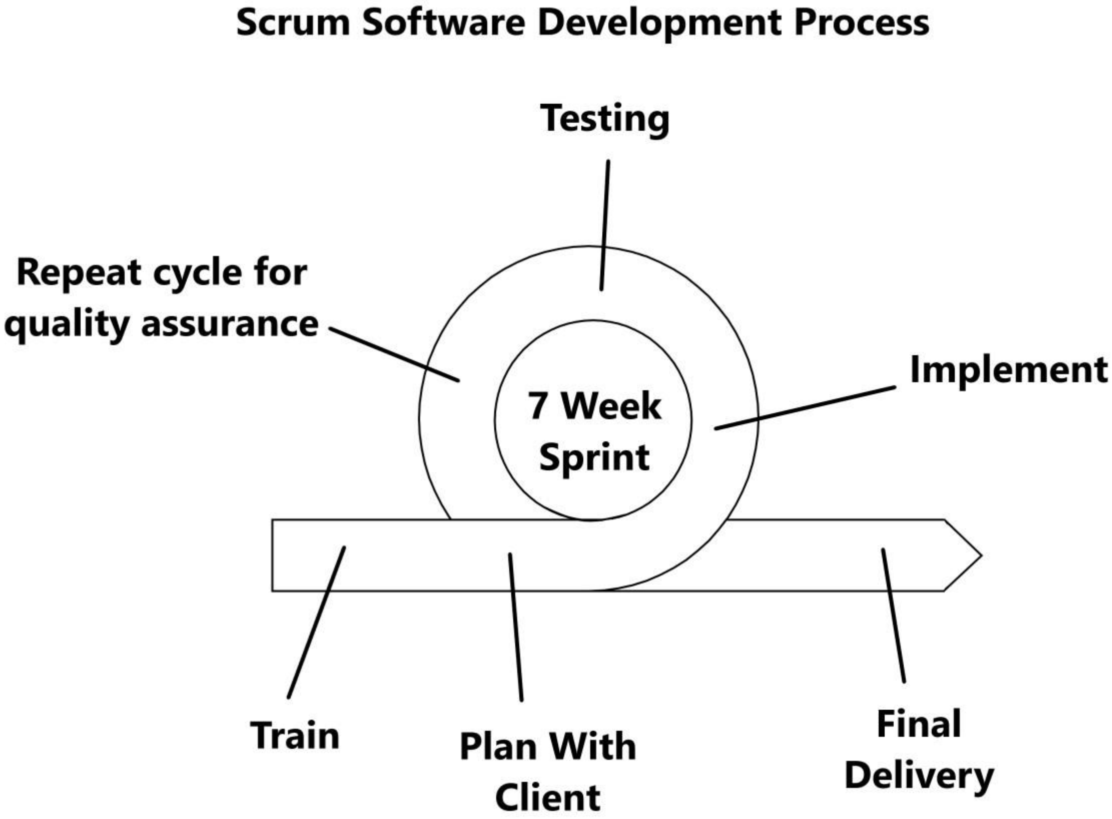
- The source code towards creating the app on Android Studio may become outdated and would need constant maintenance to make sure it is up to date with the latest Android Studio support files.
- Any member not showing enough progress for the team will be dealt with immediately. This might possibly mean banning this member from the team.
- Every member is expected to help carry the weight of any struggling members. The team must be able to adapt to changes to work environment including absent or banned team members.

5.5. Closeout plan

The team will submit all the deliverables on to the BitBucket repository, along with a maintenance manual. The team will also present their application on the finals day.

6. Technical process plans

6.1 Process model


Scrum Software Development Process
Testing
Repeat cycle for quality assurance
Implement
7 Week Sprint
Train
Plan With Client
Final Delivery

The Scrum model is essentially divided into six components: train, plan, implement, test, repeat x times, and deliver. For our purposes, the CoyoteQuest team will first train. Each member is responsible for familiarizing him or herself with the necessary tools to develop the app. Next is planning with the client. Before development can begin, the team must consult with the client about the expectations and procedures of the app. After planning, the team must implement the designs according to the client's specifications. After implementation, each member involved in the coding must test his or her components that were added to the software. If more changes or concerns need to be discussed, the cycle is repeated for quality assurance. The cycle may be repeated for any amount of times if more quality software code is needed starting from planning with the client. The last step is to then deliver the final product given that no repeat of the Scrum cycle is needed. All of these procedures must be done within a seven week span.

6.2 Methods, tools, and techniques

Method: Incremental Development Model Tools: Android Studio, Bitbucket Techniques: Regular meetings with the client.

6.3 Infrastructure plan

There is no need for an infrastructure plan at the moment because no server has been created for this app.

6.4 Product acceptance plan

The client will be kept in loop throughout the development. After which the Q&A team will test the application and make sure that the prototypes are working and acceptable.

7. Supporting process plans

7.1 Configuration management plan

The team is using Bitbucket for the configuration. It will contain all the changes made to the code and, will allow for a smoother configuration.

7.2 Verification and validation plan

Verification and validation is done through periodic testing of the mobile app: unit, integration, and system. Any bugs or errors that are found are documented and reported to the managers. For unit testing, each member involved in the programming must test his or her own code that is pertaining to their assigned software component. This kind of testing would be most often used because it is often checking the fluidity of the program as it is being designed. Once each member’s assigned components are completed, the next phase in testing is integration. Every software component is combined and tested for errors. Each member on the team must also be involved in integration testing so as to catch his or her own and each other’s errors in the code. The last phase is system testing in which the prototype software is evaluated between the two project managers, the client and the quality assurance (QA) team. This phase of testing is necessary before final delivery of the product. Any changes needed or errors found at this stage of testing, software development will need to repeat the Scrum cycle starting from “planning with client.” Testing will proceed once more from unit, to integration, and then to system.

7.3 Documentation plan

The managing team will prepare the SRS and SPMP. The development team will write the documentation for design and architecture.

7.4 Quality assurance plan

The Q&A team will make sure every aspect of the application is running according to the SRS.

7.5 Reviews and audits

During development and testing phases, every member of the development team will test and report any deficiencies in the mobile app. Design flaws or bugs will be reported and documented for immediate fix or future review.

7.6 Problem resolution plan

Each member of the development team will keep the managing team members up to date on any issues that they might encounter. Afterwards the managing team will decide on how to handle those issues and make sure that the project is completed. They will also make any changes necessary to make sure that the project runs as smoothly and as efficiently as possible.

7.7 Subcontractor management plan

We have no subcontractors.

7.8 Process improvement plan

In order to improve the process and development of the software, it is crucial to write a good documentation of the source code that they will be provided. There are features that cannot be implemented in the given time at the moment, and therefore have been pushed back to the next stage of development. Writing quality code now, documenting it and creating a good maintenance manual will help the future development team to make all the changes that they think are necessary. We also made sure that the client was kept up to date on these features, and so he will also be able to help the future development team figure out what to do next. To help improve code production or code quality, every member must report to the managers on his or her progress. Any member showing severe lack of improvement will be micromanaged by the one the project managers. This is so that a member can be closely evaluated and the project manager can therefore propose solutions regarding development issues.