Software Project Management Plan

U drop

Hanane Elkarni - Project Manager

Francys Cunanan - Assistant Project Manager

Ronnie Espinosa - Software Engineer

Daniel Firestone - Software Engineer

Ali Randell Kasim B Hamody - Software Engineer

Christopher Wise - Software Engineer

Kelsey Burgos - Software Engineer

Version 1.3

Last Updated: 6/10/20

Table of Contents

1. Overview **3**
   1.1. Project Summary 3
      1.1.1. Purpose, scope, and objectives 3
      1.1.2. Assumptions and Constraints 3
      1.1.3. Project Deliverables 3
      1.1.4. Schedule and Budget Summary 3
   1.2. Evolution of the Plan 3
2. References **4**
3. Definitions **4-5**
4. Project Organization **6-7**
   4.1. External Interfaces 6
   4.2. Internal Structure 7
   4.3. Roles and Responsibilities 7
5. Managerial Process Plans (pages 8-11) **8-11**
   5.1. Start-up Plan 8
      5.1.1. Estimation Plan 8
      5.1.2. Staffing Plan 8
      5.1.3. Resource Acquisition Plan 8
      5.1.4. Project Staff Training Plan 8
   5.2. Work Plan 8-9
      5.2.1. Work Activities
      9
      5.2.2. Schedule Allocation 9
      5.2.3. Resource Allocation 9
      5.2.4. Budget Allocation 9
   5.3. Control Plan 9
      5.3.1. Requirements Control Plan 9
      5.3.2. Schedule Control Plan 9
      5.3.3. Budget Control Plan 9
      5.3.4. Quality Control Plan 9
      5.3.5. Reporting Plan 9
      5.3.6. Metrics Collection Plan 10
   5.4. Risk Management Plan 10
   5.5. Closeout Plan 11
6. Technical Process Plans **12-13**
   6.1. Process Model 12
   6.2. Methods, Tools, and Techniques 13
   6.3. Infrastructure Plan 13
   6.4. Product Acceptance Plan 13

7. Supporting Process Plans **14**
7.1. Configuration Management Plan 14
7.2. Verification and Validation Plan 14
7.3. Documentation Plan 14
7.4. Quality Assurance Plan 14
7.5. Reviews and Audits 14
7.6. Problem Resolution Plan 14
7.7. Subcontractor Management Plan 14
7.8. Process Improvement Plan 14

1. **Overview**

1.1 Project summary

The Software Project Management Plan presents the description, estimate and assessment required in order to present a structured Udrop application. It will also include specific roles, predicted timelines and a history of testing protocols.

1.1.1 Purpose, scope and objectives

The Software Project Management Plan (SPMP) is to relay information brought from the development plan and the development cycle. It will assign roles to each member,as well as their duties to progress our development for our app. It will place necessary deadlines to meet completion requirements. Many of the specifications followed derived from previous versions of signed SRS's signed by the client.

The first prototype will contain UI design, featuring the navigation system of the application (from home page, to friend's list, and so on), the structure of the buttons so it may be easy to understand (user-friendly), and maybe see a global position of an address (slim chance but we'll try to see where it goes). The second prototype will be the tech savvy phase of the prototype. Once the team has settled down and has gained knowledge of developing the app, they will provide functionalities within the app such as being able to send messages to one another, create an account, and add a friend.

These prototypes are alpha testings to Udrop, a global positioning text-message application created by our client, Micah Langston, and he believes this will be widely used within colleges and the current youth. The team enlisted to complete such a task are as follows: Hanane Elkarni (Project Manager), Francys Cunanan (Assistant Project Manager), Ronnie Espinosa (Software Engineer), Daniel Firestone (Software Engineer), Ali Randell Kasim B Hamody (Software Engineer), Christopher Wise (Software Engineer), and Kelsey Burgos (Software Engineer).

1.1.2 Assumptions and Constraints

-Team members will make an effort to attend as many meetings as they can.
-Deliverables will be submitted by the deadline.
-The requirements in the SRS will be followed.
-Team members will devote sufficient time outside of class to complete the project.

1.1.3 Project Deliverables

- Prototype 1
- Prototype 2
- SRS,SPMP, Documented Source Code, Maintenance Manual

1.1.4 Schedule and Budget Summary

The team has no budget for this project, not even resources that's required for this project via a private server. However, the team will do the best they can with the current resources they have. Prototype #1 will be delivered in *Mid-Week of May* (week starting with May 18th, 2020) and Prototype #2 will be delivered in the *second week of June* (June 8th, 2020; final's week).

1.2 Evolution of the Plan

The PM and APM schedule a meeting with the client, Micah Langston, on April 22, 2020. By doing so, the team further understands the necessities and specifications on what the app should have or be. Next, we split the teams into two to give the app the potential to be released on both Android and iOS. Then we advance the client's former code with our own. We are expected to build the UI design and layout on the first prototype due to the restrictions of not having private servers for the project. Later, the app will have working features for the project where the teams can exchange messages to each other, add a friend's list, make an account (depending on servers), and be able to look and save locations.

2. References

Android Studio
https://developer.android.com/studio

Coyote Quest SPMP
https://mobileappdev.academic.csusb.edu/wp-content/uploads/2020/04/CoyoteQuest-SPMP-Revision-5-6-19.pdf

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

Udrop SRS
https://docs.google.com/document/d/1Y5OMZbyzwp5GYpqWRMY_DNOkqWMqWvqnFmqXP5cG-hk/edit?usp=sharing

Xcode
https://developer.apple.com/xcode/

3. Definitions, Acronyms, and Abbreviations

Android - mobile operating system designed primarily for touchscreen mobile devices such as smartphones and tablets.
APM - Assistant Project Manager

App - Short for Application.

Client - Micah Langston of *United Christian Academy*.

(AFQ) Automatic Force Quit - When a software system malfunctions and has no choice but to exit.

Comments - Strings of text that accompany source code for the purpose of making it easier to understand what the code does.

CSE - Computer Science and Engineering.

CSUSB - California State University of San Bernardino.

Documented Source Code - Source code that has been documented in such a way that the code is easily readable. Comments or diagrams may be used in such documentation

Drop(s) - messages from the Udrop app.

iOS - mobile operating system created and developed by Apple Inc. exclusively for its hardware.

Home page - The introductory page of the application.

Internet - Global network that provides information, communication, and storage of data

MB - Megabyte.

Memory - Stored information for immediate use of a device.

Mobile App - Mobile Application.

PM - Project Manager

QA - Quality Assurance. A team that tests a software for sufficient quality.

RAM - Random Access Memory.

Software Engineer - A person that designs, implements, tests, and maintains a computer software

SPMP - Software Project Management Plan.

SRS - Software Requirements Specification

TBD - To be determined

UI - User Interface.

User-friendly - Easy to use and understand.

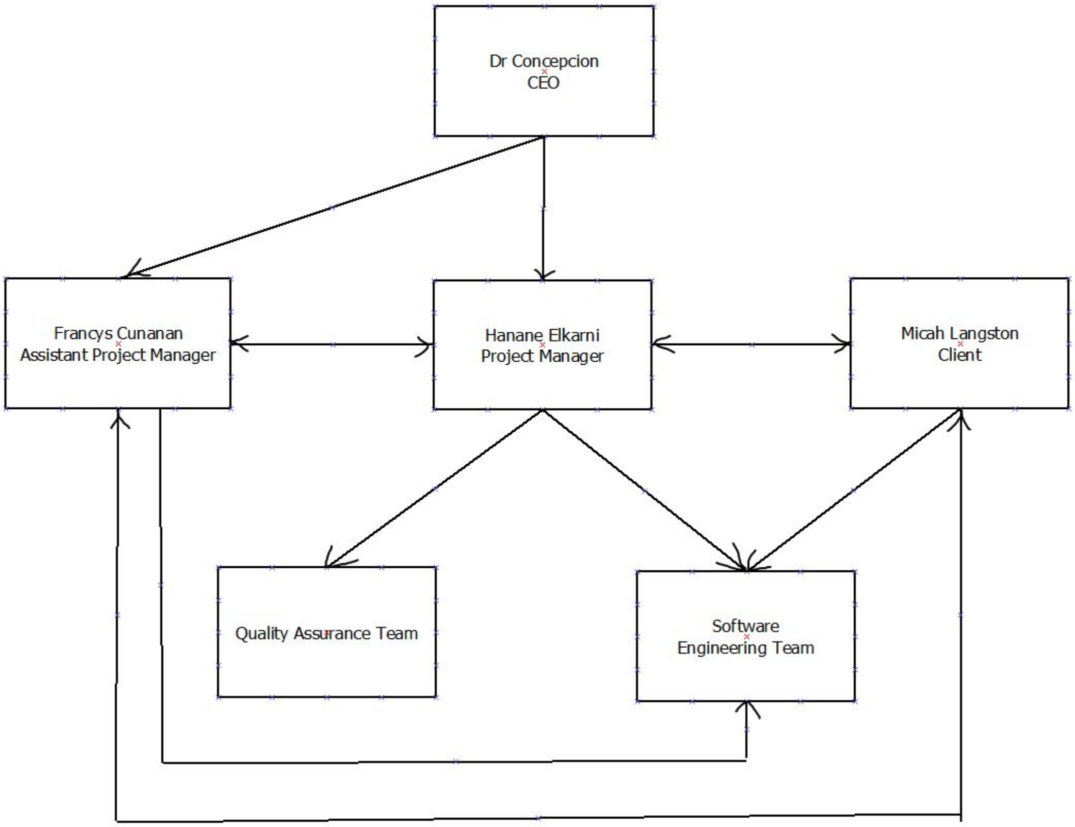
Wifi - A facility allowing computers, smartphones, or other devices to connect to the Internet or communicate with one another wirelessly within a particular area.

Xcode - Integrated development environment for macOS containing a suite of software development tools developed by Apple.

Ufavorites - Important contacts saved by the user

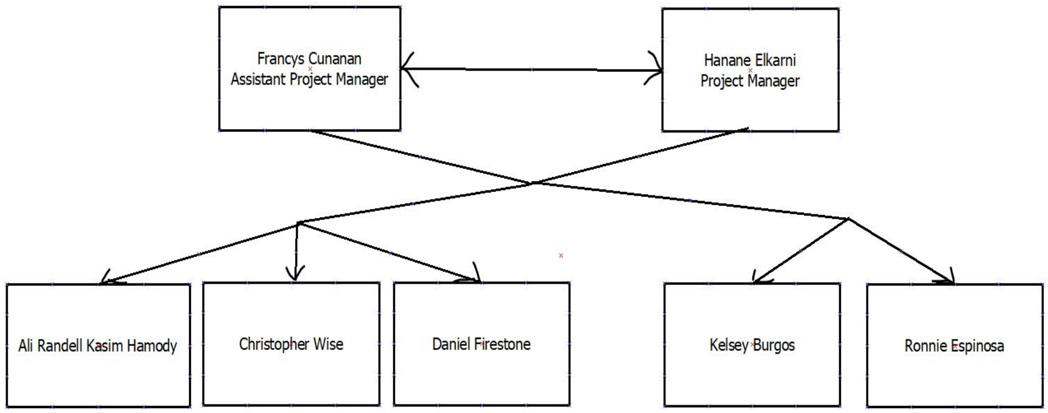
4. Project Organization

4.1. External Interfaces



Dr. Concepcion (CEO) - Guides and monitors the performance of each development team along with their product manager.
Hanane Elkarni (Project Manager) - Schedules meetings with the client and gathers this data and info, delivering it to the team. It will be done by Zoom Meetings, E-Mail, Slack, and even personal text message. Additionally, the PM will have full control of the app's programming.
Francys Cunanan (Assistant Project Manager) - Monitors team progress to stay on track of the app's personal goals. Fills in the PM's role if not present, even directing the application's development.
Micah Langston (Client) - Communicates with PM and APM to arrange a certain standard of what to expect or what needs to be done for his/her app. Additionally, the client will provide direct assistance for the team if they are confused or need something from him/her.
Software Engineering Team - Designs the code for the app, creating the layouts and features.
Quality Assurance Team - Verifies the prototype by attempting to find bugs or errors of the prototype.

4.2. **Internal Structure**



The project manager and assistant project manager cooperate to help complete the prototype's objective. They are also there to help the team get what they need to make sure the prototype is successfully finished before the deadline.

4.3. Roles and Responsibilities

Hanane Elkarni (Project Manager) - Communicates with the team's client to know what to expect or key desires he or she would want in the mobile app. Additionally, PM in charge of the iOS side of the team.

Francys Cunanan (Assistant Project Manager) - Assists with PM to help complete the prototype's development. Additionally, APM is in charge of the Android side of the team.

Ronnie Espinosa (Software Engineer) - Designs UI components and modifies design of the client's original code in iOS.

Daniel Firestone (Software Engineer) - Designs UI components and converts design of the client's original code into Android.

Ali Randell Kasim B Hamody (Software Engineer) - Creates app feature functionality in Android such as being able to create accounts.

Christopher Wise (Software Engineer) - Works on specific app features like via friend's list and text messaging.

Kelsey Burgos (Software Engineer) - Creates app feature functionality in iOS's side like managing locations.

5. Managerial Process Plans

5.1. Start-up Plan

5.1.1. Estimation Plan - PM and APM will do weekly meetings by the client to show what is done so far and steer the prototypes' development if the client requests a new task for the project's direction.
Observe the client's original code and modify or rewrite it.
Study and research for more tools to help improve the project's coding.
Keep on track of the team's progression.
Arrange key goals for both Prototype 1 and Prototype 2, and complete them before the deadline.

5.1.2. Staffing Plan - The team was chosen based on the skills survey each student took at the beginning of the course. We have since split into iOS and Android teams due to half of the team not owning a Mac. The iOS staff will specifically lead the prototype since the client specifically mentions that the app needs to finish before the Android version because it's a higher priority.

5.1.3. Resource Acquisition Plan - The software tools the team will use is all for free use. The team will use personal home computers or their macs to further continue the prototype's progression. The team will also be testing it with their smartphones if the app works as intended.

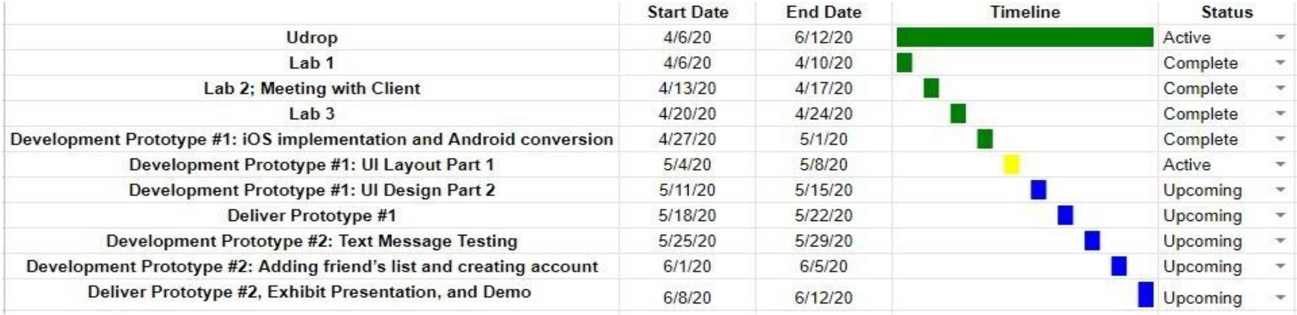
5.1.4. Project Staff Training Plan - Each team member has completed two lab assignments designed to introduce us to our respective development tools. The team of two platforms are expected to learn their side of coding language in the course's first three weeks. Afterwards, the team would need to research with online tutorials to come up with new methods to approach the needed features in the app.

5.2. Work Plan

5.2.1. Work Activities
Prototype 1 - The entire team will work on creating an existing UI layout of the overall app. What needs to be done is to create a navigation system with a design that looks inviting and user-friendly for average-day college students (our client's targeted demographics). A UML low-level design is only necessary for this prototype for the time being since we are prioritizing UI appearance, viewer efficiency, and making it easy to use (navigate for now) in the first half of the course.
Prototype 2 - The team will then implement features needed in the mobile app such as sending text messages, images, adding friends, and so on.
Document SRS and SPMP - The entire team will partake and review each section of the documents before submitting. To work with this prototype, the team would then need a UML high-level design to work with creating

an account and adding that account (or contacts) to their friend's list. This is because multiple functions are happening with different pages unlike the previous prototype the team has to deal with.

5.2.2. Schedule Allocation



	Start Date	End Date	Timeline	Status
Udrop	4/6/20	6/12/20	🟩	Active
Lab 1	4/6/20	4/10/20	🟩	Complete
Lab 2; Meeting with Client	4/13/20	4/17/20	🟩	Complete
Lab 3	4/20/20	4/24/20	🟩	Complete
Development Prototype #1: iOS implementation and Android conversion	4/27/20	5/1/20	🟩	Complete
Development Prototype #1: UI Layout Part 1	5/4/20	5/8/20	🟨	Active
Development Prototype #1: UI Design Part 2	5/11/20	5/15/20	🟦	Upcoming
Deliver Prototype #1	5/18/20	5/22/20	🟦	Upcoming
Development Prototype #2: Text Message Testing	5/25/20	5/29/20	🟦	Upcoming
Development Prototype #2: Adding friend's list and creating account	6/1/20	6/5/20	🟦	Upcoming
Deliver Prototype #2, Exhibit Presentation, and Demo	6/8/20	6/12/20	🟦	Upcoming

5.2.3. Resource Allocation

The members: Francys Cunanan, Christopher Wise, Daniel Firestone, and Ali Randell Kasim B Hamody all have access to the same resources pertaining to Android studios. Whereas, Hanane Elkarni, Ronnie Espinosa, and Kelsey Burgos all have access to the same resources pertaining to IOS. The resources we will definitely need to acquire is a private server for these beta testings. It should work for both iOS and Android.

5.2.4. Budget Allocation

There is no budget allocated that has been set for this project.

5.3. Control Plan

5.3.1. Requirements Control Plan

Each team member is required to attend the (lab) meetings in class. Additionally, each developer is required to document their code to showcase their contribution to the project. Furthermore, every member needs to follow their project's SRS so it will help them finish the project. Any inconveniences or difficulties addressed by the client are to be evaluated by the managing team so that they may weigh out decisions.

5.3.2. Schedule Control Plan

Besides the regular lecture and lab times, the team needs to perform a schedule to collaborate or meet outside of class time, in order to complete the product on time. Managers will make sure to arrange meetings so that completing it is possible.

5.3.3. Budget Control Plan

There is neither a budget for this project nor a private server currently to test the app out.

5.3.4. Quality Control Plan

Both the manager and development team are to check the quality of the software weekly to verify that it meets the client's standards; additionally, the client will be informed of the quality of the application.

5.3.5. Reporting Plan

The managing team will update bi-weekly of the project's progression. They will also keep track and report their team's attendance and the application's status.

5.3.6. Metrics Collection Plan

The PM and APM will collect two metrics to measure the performance of each team member. The first metric is Faults per 1000 Lines of Code, and the second is Total Lines of Code per Hour.

Productive and Quality Measure

Hanane Elkarni (Project Manager) - Will have regular communication with Micah Langston (client) to ensure that the team is meeting the expectations of his app. Will relay any project changes to the iOS team and assist where needed.

Francys Cunanan (Assistant Project Manager) - Will have regular communication with Micah Langston (client) to ensure that the team is meeting the expectations of his app. Will relay any project changes to the Android team and assist where needed.

Ronnie Espinosa (Software Engineer) - Design UI components and modify the client's original code in iOS . ~100 LOC

Daniel Firestone (Software Engineer) - Design UI components and convert design of the client's original code into Android. ~100 LOC

Ali Randell Kasim B Hamody (Software Engineer) - Create app feature functionality in Android. Specifically the ability to create accounts. ~100 LOC

Christopher Wise (Software Engineer) - Work on specific Android app features like friend's list and text messaging. ~100 LOC

Kelsey Burgos (Software Engineer) - Create app feature functionality iOS like managing locations. ~100 LOC

5.4. Risk Management Plan

Development:

The team needs to meet once or twice a week so that the project's progression rate doesn't stop.

There are scheduled deadlines so that the team doesn't fall behind on what is promised.

Each and every team member are well-informed and up-to-date. If there is ever a technical difficulty or inconvenience that's blocking the project's progression, it needs to be reported by the managing team so that they can work with a way to fix them.

Project Failure:

Not everyone is working on the same platform, so we each need to keep track of each other's progression so we stay in the same direction and one doesn't fall behind.

Any member shown not helping the team will be discredited from the project. Before that though, the managing team would need to watch that member more carefully.

Any member that is helping but still lacking in progression is still expected to receive help from other team members; if one person fails, then we all fail.

5.5. Closeout Plan

We will upload the following items to the project Bitbucket repository:
SRS, SPMP, SQAP, SAD, documented source code, maintenance manual, and any other items that we may deem necessary.

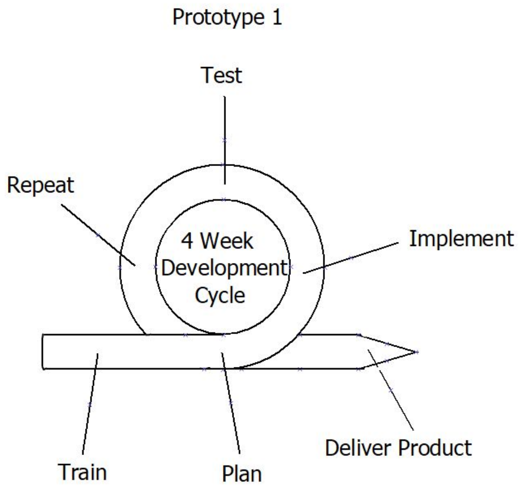
All these items will be made accessible to the next team that works on the project.

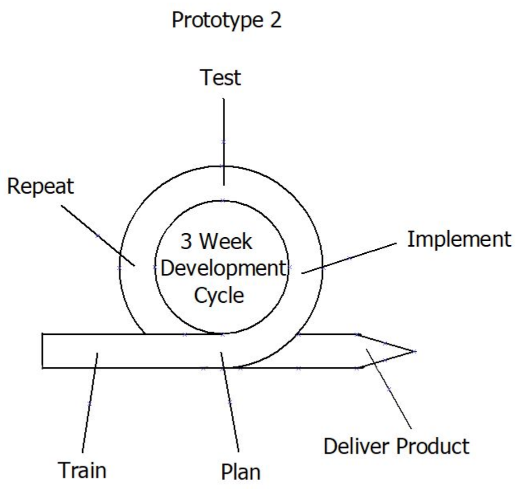
6. Technical Process Plans

6.1. Process Model

The Scrum model is divided into six parts: train, plan, implement, test, repeat, and deliver. By completing the assigned labs and doing independent research, each team member will train themselves in the knowledge needed to develop Udrop. By meeting with the client and communicating with team members we will make a plan for developing Udrop. The team will then implement what we have planned, test it through various methods, and repeat the process until the quality of the product is satisfactory. Once Udrop has reached the desired quality we will deliver the prototype.

Scrum Software Development Process





6.2 Methods, Tools, and Techniques:
Method: Incremental Development Model
Tools: Android Studio, Xcode, Bitbucket
Techniques: Regular communication with the client and team members

6.3 Infrastructure Plan:
Server services are TBD. We have spoken with the client about this, and he is looking into what server provider he wants to use.

6.4 Product Acceptance Plan:
All team members and the client will be kept up to date on the progress of development. Once a prototype has been made, the QA team will evaluate it.

7. Supporting Process Plans

7.1 Configuration Management Plan:
The team will use BitBucket for both versions of the app. It will contain all the changes made to the code to allow for smoother configuration

7.2 Verification and Validation Plan:
Verification and Validation is done through periodic testing of the app as the developers test their own code. Any bugs or errors that are found will be documented.

7.3 Documentation Plan:
The managing team will prepare the SRS and the SPMP. The developmental team will write the documentation for design and architecture.

7.4 Quality Assurance Plan:
Quality Assurance will be done routinely to check that the app is running up to the SRS standard. This will be done by bug testing extensively and fixing the app as we continue development.

7.5 Reviews and Audits
As we continue to develop and test the app, we will keep a developmental log to document bugs and possible design flaws for review and to fix at a later date.

7.6 Problem Resolution Plan:
Members of the developmental team will keep the members of the managing team up to date on any complications during development. From there, the management team will decide on how to proceed and fix the issue. They will also have the ability to make changes as they deem necessary to ensure that development runs as smooth as possible.

7.7 Subcontractor Management Plan:
We have no subcontractors.

7.8 Process Improvement Plan:
To improve the developmental process of the app, we will work to have an extensive documentation of the source code. If there are features that we cannot implement at the moment, we will as a group decide to push it to the next stage of development. Creating a solid foundation will allow development to go smoothly when we do not have to backtrack and continuously fix existing code. We will make sure that our client is kept up to date with all developmental progress and receive his input on our current progress.