

**DropCop**

**Software Project Management Plan**

**Version 2.0**

**March 6, 2015**

Document Number: SPMP-001

Project Team Number: **B12**

Project Team Members:

**Name Poly ID**

**Pavandip Singh ps1929**

**Mohammed Alam mo1999**

**Justin Opraseuth jo858**

**Review and Approval**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Date** | **Signature** |
| Fred Strauss | Approval |  | On File |
| Pavandip Singh | Author and Reviewer | March 6th, 2015 | On File |
| Mohammed Alam | Author and Reviewer | March 6th, 2015 | On File |
| Justin Opraseuth | Author and Reviewer | March 6th, 2015 | On File |

**Revision Level**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
| November 8th, 2014 | Version 1.0 | Initial Release |
| March 6th, 2015 | Version 2.0 | Revised for Senior Project |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**1. OVERVIEW ............................................................................................................................................ 6**

**1.1 PROJECT SUMMARY ......................................................................................................................... 6**

**1.2 PURPOSE, SCOPE, AND OBJECTIVES ............................................................................................ 6**

**1.3 ASSUMPTIONS AND CONSTRAINTS ................................................................................................ 6**

**1.4 PROJECT DELIVERABLES ................................................................................................................ 6**

**1.5 SCHEDULE AND BUDGET SUMMARY .............................................................................................. 7**

**1.6 EVOLUTION OF THE PLAN ................................................................................................................ 7**

**2 REFERENCES ........................................................................................................................................ 7**

**3 DEFINITIONS .......................................................................................................................................... 7**

**4 PROJECT ORGANIZATION ................................................................................................................... 8**

**4.1 EXTERNAL INTERFACES ................................................................................................................... 8**

**4.2 INTERNAL STRUCTURE ..................................................................................................................... 8**

**4.3 ROLES AND RESPONSIBILITIES ...................................................................................................... 8**

**5 MANAGEMENT PROCESSES ............................................................................................................... 9**

**5.1 START-UP PLAN ................................................................................................................................. 9**

**5.1.1 Estimation Plan ................................................................................................................................ 9**

**5.1.2 Staffing Plan ..................................................................................................................................... 9**

**5.1.3 Resource Acquisition Plan ............................................................................................................. 9**

**5.1.4 Training Plan .................................................................................................................................... 9**

**5.2 WORK PLAN ...................................................................................................................................... 10**

**5.2.1 Work Activities ............................................................................................................................... 10**

**5.2.2 Schedule Allocation ....................................................................................................................... 11**

**5.2.3 Resource Allocation ...................................................................................................................... 11**

**5.2.4 Budget Allocation .......................................................................................................................... 11**

**5.3 CONTROL PLAN ............................................................................................................................... 11**

**5.3.1 Requirement Control and Traceability ......................................................................................... 11**

**5.3.2 Schedule Tracking and Adjustment ............................................................................................ 12**

**5.3.3 Budget Tracking and Adjustment ................................................................................................ 13**

**5.3.4 Quality Control ............................................................................................................................... 13**

**5.3.5 Reporting Mechanisms ................................................................................................................. 13**

**5.3.6 Metrics Collection Plan ................................................................................................................. 13**

**5.4 RISK MANAGEMENT PLAN ............................................................................................................. 13**

**5.5 POST IMPLEMENTATION PLAN ...................................................................................................... 14**

**6 TECHNICAL PROCESSES ................................................................................................................... 14**

**6.1 PROCESS MODEL ............................................................................................................................ 14**

**6.2 METHODS, TOOLS, AND TECHNIQUES ......................................................................................... 14**

**6.3 INFRASTRUCTURE PLAN ................................................................................................................ 14**

**6.4 PRODUCT ACCEPTANCE AND MIGRATION PLAN ....................................................................... 15**

**7 SUPPORTING PROCESSES PLANS .................................................................................................. 15**

**7.1 CONFIGURATION MANAGEMENT PLAN ........................................................................................ 15**

**7.2 QUALIFICATION (VERIFICATION AND VALIDATION) PLAN ........................................................ 15**

**7.3 DOCUMENTATION (LIBRARY) PLAN .............................................................................................. 15**

**7.4 QUALITY ASSURANCE PLAN .......................................................................................................... 16**

**7.5 REVIEWS AND AUDITS .................................................................................................................... 16**

**7.6 PROBLEM RESOLUTION PLANS .................................................................................................... 16**

**7.7 ENVIRONMENT MANAGEMENT PLANS ......................................................................................... 17**

**7.8 PROCESS IMPROVEMENT PLAN .................................................................................................... 17**

**8. ADDITIONAL PLANS ........................................................................................................................... 17**

**9 INDEX .................................................................................................................................................... 17**

**10 RATIONALE ........................................................................................................................................ 17**

**11 NOTES ................................................................................................................................................. 18**

**12 APPENDICES ...................................................................................................................................... 18**

**12.1 SCHEDULE TRACKING .................................................................................................................. 18**

**12.2 DEFECT TRACKING ........................................................................................................................ 20**

**12.3 GANTT CHART ……………………………………...……................................................................... 22**

**1. Overview**

**1.1 Project Summary**

The motive behind this project is to create an app that will help users in 3 ways. It will generate polls of specific items that are available in the world and get cheaper and more efficient deals on them. The more people that want an item, the lower its price should be. Not only that, companies and individuals can post about deals they have on certain products. This will help satisfy the three background goals we desire to get. The first being reducing time for searching for products. Second, companies can post deals themselves to get their sales up. Third, being minimizing users time spent buying/selling stuff. Since, so much technology is available in the world, people nowadays often shop online and get things shipped to their house in order to reduce time they spent shopping.The intended audiences of the SPMP are software will be the developers, managers, and the SQA group. The purpose of this document is to measure the record the progress and quality of the software project.

**1.2 Purpose, Scope, and Objectives**

The goal of DropCop is to simplify the average user shopping time and money spent on items they are willing to buy or need. This application will span the iOS and Android operating systems. DropCop will consist of a user friendly UI where companies can post deals that users can directly buy, or people can make polls about the items they will want, eventually leading to a group discount.

**1.3 Assumptions and Constraints**

The application will be able to run on all modern versions of the Android and iOS operating systems because there will be no hardware constraint. The interface for both devices should be should be able to provide a smooth and fluid experience while giving a sense of easiness. We will have to work out a schedule for all team members to come together and meet each other weekly. There is no budget involved in this project. We will need to learn how iOS and Android platforms work to implement in a C++ foundation.

**1.4 Project Deliverables**

|  |  |
| --- | --- |
| **Project Component** | **Due Date** |
| Project Proposal | February 13, 2015 |
| Requirements and Analysis Documentation (RAS) | February 27, 2015 |
| Software Project Management Plan (SPMP) | March 6, 2015 |
| Design Description (SDD) - Initial | March 20, 2015 |
| Design Description Final (w/Code) | April 17, 2015 |
| Implementation/Demonstration | April 24, 2015 |
| Formal Oral Presentations | April 24 - May 4, 2015 |
| Project Portfolio | Final Week |

**1.5 Schedule and Budget Summary**

We will begin to work on this project over the spring semester of 2014. Since there is no specific time involved on a schedule for the design project, the team will meet up 2-3 times a week to synchronize what has been completed thus far. Each individual will be responsible for about 4-8 hours of work in developing the code for the application. For the first milestone, we will focus on completing all the classes needed for the setup along with the UI layout. In the second milestone, we will check for errors and optimize all the classes and the UI setup. We will not have to worry about the budget of this project because it is not a concern.

**1.6 Evolution of the Plan**

Modifications to the project/application plans must be done through a mutual agreement by all the group members, developers, and the SQA group. All modifications made to the code implementation or system design, or requirement change will be recorded in the Software Requirements Specification documents, as well as the revision section of the document.

**2. References**

- “Project Proposal”, Team A12

- DropCop, Software Requirements Specification, Version 1.1, Revision 0

**3. Definitions**

DropCop: Name of the Application/project

SRS: Software Requirements Specification

SPMP: Software Project Management Plan

SAS: Software Analysis Specification

SQA: Software Quality Assurance

SDD: Software Design Document

WBS: Work Breakdown Structure

LOC: Lines of Code

**4. Project Organization**

**4.1 External Interface**

The project will be handed off to the software quality assurance (SQA) for different types of testing once its complete. Also, SQA will review and approve the SPMP and all other documentation to make sure that the requirements are met. The SPMP and all other documentation will also be reviewed by the client to make sure the clients approves of the specification laid out in this document. Not only that, there will be time spent weekly handling issues of the project with the client, and the SQA team. All the deliverables will be submitted to the client, the instructor, and SQA team on their specific due dates.

**4.2 Internal Structure**

This team will work in and form a democratic team with no egotistic programming or single leader. All work will divided between all members. They then will work together to detect faults in their own and other members’ faults. More than likely, they will be looking to find defects in code other than their own for best results. All the deliverables will be divided up also between each member so that each person has roughly an equivalent part to do. Afterwards, the SQA will receive the SPMP and will review each deliverable with inspections and walkthroughs to detect faults. If faults are found, they will be fixed by the development team.

**4.3 Roles and Responsibilities**

In this project, the instructor will serve the role as the Project Manager. He will be responsible for reviewing all deliverables. All the deliverables will be submitted to the instructor by their respective due dates. Also, since this is democratic team, each member has the same responsibility in the team and will be given equal amounts of work.

The chart below describes and provides the responsibilities for the people involved in this project.

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Email Address** |
| Fred Strauss | Instructor and Project Manager | fred.strauss@nyu.edu |
| Pavandip Singh | Author for Documentation,  Programmer, Reviewer, Tester | ps1929@nyu.edu |
| Mohammed Alam | Author for Documentation,  Programmer, Reviewer, Tester | ma1999@nyu.edu |
| Justin Opraseuth | Author for Documentation,  Programmer, Reviewer, Tester | js858@nyu.edu |

**5. Management Processes**

**5.1 Startup Plan**

**5.1.1 Estimation Plan**

|  |  |
| --- | --- |
| Pavandip Singh | $40,000 |
| Justin Opraseuth | $40,000 |
| Mohammed Alam | $40,000 |

**5.1.2 Staffing Plan**

The staffing plan for the product is listed in the table below:

|  |  |
| --- | --- |
| Android Developer | 1-2 years of experience |
| iOS Developer | 1-2 years of experience |
| Database Programmer: | 1-2 years of experience |
| Programming Secretary | 1 or more years of experience |

The staff listed will be needed throughout the whole Software development lifecycle. Since the source of the staff is existing team members, everyone will have more than one role. Assignments will be give and made in future revision of the SPMP.

**5.1.3 Resource Acquisition Plan**

We will need to train each member of our team the software requirements building the application. Learning the language, iOS and Android will be the most difficult time. We will set a plan to learn those requirements. This is included in the training plan below.

**5.1.4 Training Plan**

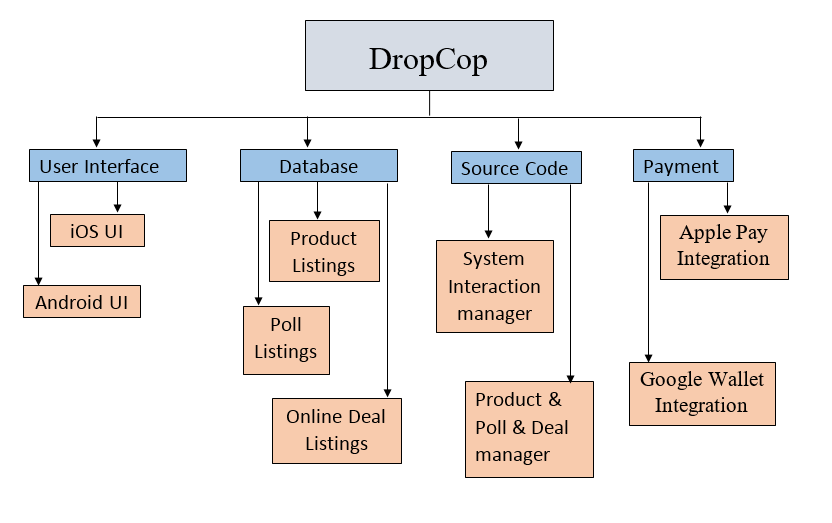
The following is a tentative schedule of training workshops for the appropriate staff members. Additional trainings will be added (or removed) based on staff needs as well as project requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Training** | **Number of Staff** | **Entry Criteria** | **Exit Criteria** | **Training Method** |
| Effective/Good Teamwork Skills | All members (3) | None | Each member will understand how to work with each other effectively & efficiently within the development team. | Presentation,  Scheduled Team meetings/ workshops |
| Application Development Basics (iOS & Android Platforms) | All members (3) | Basic Programming Skills (any language) | Each member will gain an understanding of the factors (such as security, market rules and standards etc) that go into making mobile applications. Also, will learn the process and necessary implementation to make a mobile application for different platforms. | Workshop / Hands on practice/ Lecture |
| Management Skills | All members  (3) | Basic Documentation skills | Each member will learn effective methodologies of managing team members while keeping track of the most up to date documentation and product information. Not only that, members will come together to synchronize everything they have completed or in the progress of and have it evaluated by the others. | Lecture / Workshop |

**5.2 Work Plan**

**5.2.1 Work Activities**

Work Breakdown Structure:



**5.2.2 Schedule Allocation**

A detailed schedule outline can be found on the Gantt Chart in section 12.3. The provided chart details the allocation of work activities and the major milestones. The work activities are broken down into sections so they can be completed independently. This will allow tasks to be completed concurrently without overlap of resources. Extra time was added to account for delays and setbacks. All tasks and resources will be prioritized based on criticality and time with more intensive/important tasks receiving more resources in order to ensure the tasks are completed within the scheduled time.

**5.2.3 Resource Allocation**

The planned resource allocation is listed on the Gantt chart in section 12.3. Tasks will be assigned to personnel familiar with the required tools needed to complete the task. Skilled personnel will be assigned more complex tasks. Because of the way work activities are divided into independent sections, some resources will only be used by the personnel working on those tasks. As requirements change, resources may increase or decrease and as such the Gantt chart may change. All changes will be noted in the revisions of the SPMP.

**5.2.4 Budget Allocation**

N/A

**5.3 Control Plan**

**5.3.1 Requirement and Control Traceability**

All changes to the requirements will be documented with the information regarding the changes. This includes who requested the change, who made the change, what was changed, when it was changed, why the change was made, and team members approval or rejection of change.

The procedure for any modifications are as follows:

1. The client or development personnel requests a change or improvement in the requirements.

2. A team member will record the change information on the document specified. The team member in charge of the task that is being modified will also be in charge of documenting the request.

3. The team will analyze the change impact using the assessment methodology specified below and will verify the changes accordingly. A decision will be made to keep or reject changes.

4. The team member in charge of the task that required modification, will make the required changes.

5. All information regarding the changes such an implementation or rejection will be documented.

Requirement change assessment methodology:

The product before and after changes will be compared in order to assess which overall product is closer to the intended finished product. The team will account for overall impact of the change such as regression faults and future changes caused by the request.

Dependent of the severity of the requested modification, the team’s assessment may include: cost-benefit analysis, schedule change analysis, requirement review, and risk analysis.

If the change is found to have minimal impact on other parts of the product and can be implemented without a major impact on time, the change will be accepted.

Otherwise, it may be rejected if it is a nonessential change or if it impacts the overall planning and requires major reallocation of time and resources. It may also be rejected if the change is requested late in the implementation process, which could threaten the entire product completion schedule.

Finally, after the team decides to either make the changes or reject the request, a team member will document the decision and act accordingly. If the request is rejected, team members have the ability to offer alternatives that would have less of an impact on the project goals.

**5.3.2 Schedule Tracking and Adjustment**

The scheduling will be based on tasks outlines by the WBS in section 5.2.1. A document will detail the estimated completion dates for the milestones.

The team will have scheduled weekly sessions, whether in person or online, to evaluate the progress of the team. Each member will explain the progress they have made, what they are planning to do next, any problems they may have experienced, and any help they may require. Adjustments will have to made if any member has a problem meeting the deadline for a task. All issues regarding resources will also need to be accommodated accordingly. In addition, members are encouraged to have daily contact with fellow members should a problem arise, in order to fix the issue immediately. Documents should be reviewed by all members at this meeting, whether digitally or on paper. Team members may have to jump from one of their current tasks in order to help a fellow member complete their task and meet the milestone deadline. These meetings are meant to aid the team in satisfying deadlines by keeping track of progress and documentation, as well as minimizing problems.

**5.3.3 Budget Tracking and Adjustment**

N/A

**5.3.4 Quality Control**

To ensure quality control, team members will be encouraged to peer review. The democratic team structure will require members to review other members code. Because each member is in charge of writing code, documenting, and testing an individual task, it is crucial that another member is able to understand documentation and verify proper operation. The team members reviewing will work to find faults and help a programmer solve issues. The team will have peer review sessions after the completion of each members tasks, to verify all aspects of each others work are completed and meet the requirements.

**5.3.5 Reporting Mechanisms**

N/A

**5.3.6 Metrics Collection Plan**

The performance metrics are listed:

* Lines of Code (LOC)
* Performance time (time need to complete given task)
* Number of faults

The metrics will be collected after each task is completed and peer reviewed. This is to ensure that the all faults are found and the tasks have actually been completed. The relationship between the LOC and the number of faults will be compared time in order to assess the efficiency of each member and the team overall. Members with a stronger ability to complete tasks with less time and less faults, may be asked to help out struggling members. The time to complete tasks will also help the team adjust the schedule.

**5.4 Risk Management Plan**

There will be a bi-weekly risk evaluation, which will include any changes that team members suggest, functionality, progress, faults detected, and technical issues. The purpose of this risk evaluation is to find risks and to predict any impending risks that may occur due to changes. Risks will have to be prioritized based on criticality. Any highly critical risk will need to be dealt with immediately. Noncritical risks should also be dealt with because they may turn critical later in the development. Technical issues will automatically be marked highly critical because if someone does not work correctly, the risk of it delaying the project is almost certain. Team members are encouraged to evaluate risks on their own and introduce them during meetings. The team should work together to mitigate risks by taking the appropriate action.

The major risks that are probable are shown below with how they are detected and ways to mitigate the issue.

|  |  |  |  |
| --- | --- | --- | --- |
| Risk Type | Risk Description | Detection | Mitigation |
| Technology/  Software | Incompatible software tool that makes interfacing with the appropriate device a problem or performance needs are not met. | Member finds that one of the required parts of software is incompatible. The time it takes to run a module or program is longer than expect. | Team should find a workaround or alternative. If they cannot, the requirements must change. A new piece of software may be needed. |
| People | Programmer cannot meet a deadline | During the schedule tracking meetings, team should evaluate which member are behind schedule | Team should evaluate if the programmer needs assistance or may need to decrease workload. Schedule changes and task allocation changes are required. |
| Technology | Piece of equipment or software may fail. For example, one of the servers may go offline or an application fails and terminates. | Users cannot access data or application fails to load. | Must have a backup server that automatically goes online should the primary server fail. Software should use autodetection to make sure all parts are working, should one part fail, it should reset. |

**5.5 Post Implementation Plan**

N/A

**6. Technical Processes**

**6.1 Process Model**

The agile process will be utilized of the project development. Once the first version of the project is completed, additional specifications will be added one by one in order of importance. Because of this, the first version of the application will be a rough model. Code inspections will happen weekly to ensure that the code is up to standards. The initial project will be divided in terms of components. First, the object-oriented backend will be coded, then a MySQL server will be implemented, and finally the mobile applications will be developed. After each iteration of the product all artifacts will be reviewed by the SQA team.

The project documents will be revised as business needs change and in the event that any mistakes are encountered. The project deliverables can be seen by glancing at section 1.4 of this document.

The project will be terminated if and only if CopDrop is no longer considered profitable.

**6.2 Methods, Tools, and Techniques**

The object-oriented paradigm will be used to implement this project. Code written by a programmer will be reviewed with another team member after every 500 LOC. Microsoft Visual Studio will be utilized as the IDE for the project. C++ will be utilized for the backend. For the database, a MySQL server will be put into place. Also, the development tools for both Android and iOS applications will be utilized. Microsoft Word will be used to generate the project planning documents. An online site called Tom’s Planner will be used to generate the Gantt chart.

**6.3 Infrastructure Plan**

The software will be coded on each team member’s personal laptop. Code will be managed via Git. Group members will communicate via email, Facebook group chat, Skype, and SMS messaging. Most work will be completed individual but with inspections and meetings likely held in the Dibner Library.

**6.4 Product Acceptance and Migration Plan**

N/A

**7. Supporting Processes Plans**

**7.1 Configuration Management Plan**

All source code for the project will be maintained via Git. The master copy of the source code will be hosted on the team leader’s Git server. Local copies of the source code will be on each member’s laptop.

**7.2 Qualification (Verification and Validation) Plan**

The code that is written by the team will be verified constantly. CppUnit will be the automated test framework used. Test cases will be maintained and regression testing will take place to ensure that no new bugs are introduced. Each unit of code will need to be traced back to the SRS to ensure that no man-hours are wasted on extraneous features. Every Git push made by a programmer will include a note identifying what changes were made and where in the SRS does the feature stem from. Each Git push should be accompanied with the completion of the unit tests created by the SQA team.

As stated in section 6.2, there will be code inspection between team members every time 500 new lines of code are committed to the repository. Walkthroughs will be initiated whenever a team member runs encounters a difficult situation.

**7.3 Documentation (Library) Plan**

Each of the project documents will be stored via Google Drive. This includes the SRS, SPMP, SAS, SDD, and meeting minutes. Deliverables will be well-documented and stored on the Git main repository. These deliverables also include the unit tests written by the SQA team. Each team-member is expected to contribute equally to the project documentation. All errors will be recorded and steps will be taken to try to reduce future errors. The notes associated with each Git commit will also have to be cross-referenced with the project documentation periodically. All members will be expected to contribute to the verification of the final versions of each project document.

**7.4 Quality Assurance Plan**

As part of the agile process, testing is required in all phases of project development. Walkthroughs will take place whenever there is a problem in hand that a team member cannot complete by his/her self. Inspections will occur at 500 LOC increments. The SQA team will be responsible for delivering quick solutions to any errors found by the development team. As stated previously, the sections of code will be traced back to design specifications by the Git commit notes.

**7.5 Reviews and Audits**

First and foremost, all developers should be take care to ensure that their code is well-reviewed before each commit to Git. In the event that the unit testing catches an error, SQA team members are expected to convene within 24 hours to patch the software. Most coding errors are expected to be found at the inspection phase at the conclusion of every 500 LOC. It is important that the same team member does not review the same person’s code too often. The same team member should not review another member’s code (during inspection) more than twice in a row. As part of the object-oriented paradigm, each module of code should be as contained as possible. This will simplify the unit tests to be run and create higher efficiency as a whole. Prior to each software release, team members should convene to determine where most efforts have been placed in, and also if there were any issues that arose.

**7.6 Problem Resolution Plan**

Problems are to be documented within a folder of the Google Drive after patching. These documents will be organized by date, and the progression of faults will be analyzed periodically. As part of the agile process, different tasks are to be prioritized and it is expected for team members to be able to jump to other tasks as needed.

The scope of the unit tests also will be stored in the Google Drive. The SQA team will be responsible to maintain this document and its revisions.

**7.7 Environment Management Plan**

N/A

**7.8 Process Improvement Plan**

N/A

**8. Additional Plans**

We will need to make sure the system is stable after it’s delivered to the users. Thus, we will need to regularly check the maintain the server. We also have to make sure people posting the available or taken parking spots aren’t a hoax, so another algorithm will need to be implemented to ensure the detection and rejection of spam.

**9. Index**

N/A

**10. Rationale**

N/A

**11. Notes**

N/A

**12. Appendices**

**12.1 Schedule Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact or**  **Deliverable** | **Who (individual**  **or Team)** | **Estimated** | **Actual** | **Difference** |
| Initial SRS 1-6 | Pavandip Singh | 3 hours | 3 hours | 0 |
| Initial SRS 8-13 | Justin Opraseuth | 4 hours | 3 hours | -1 |
| Initial SRS 7,14 | Mohammed Alam | 3 hours | 3 hours | 0 |
| Initial SRS | Total | 10 hours | 9 hours | -1 hour |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact or**  **Deliverable** | **Who (individual**  **or Team)** | **Estimated** | **Actual** | **Difference** |
| Final SRS 1-6 | Pavandip Singh | 1 hours | 1 hours | 0 |
| Final SRS 8-13 | Justin Opraseuth | 1 hours | 1 ½ hours | ½ |
| Final SRS 7,14 | Mohammed Alam | 1 hours | ½ hours | -½ |
| Final SRS | Total | 3 hours | 3 hours | 0 hours |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact or**  **Deliverable** | **Who (individual**  **or Team)** | **Estimated** | **Actual** | **Difference** |
| SPMP 1-5.1 | Pavandip Singh | 3 hours | 2 ½ hours | -½ hour |
| SPMP 6-11, 12.3 | Justin Opraseuth | 3 hours | 3 hours | 0 hour |
| SPMP 5.2-5.5, 12.1-12.2 | Mohammed Alam | 2 hours | 2 ½ hours | +½ hour |
| SPMP | Team | 8 hours | 8 hours | 0 hour |

**CUMULATIVE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Who (individual**  **or Team)** | **Estimated** | **Actual** | **Difference** |
| Pavandip Singh | 7 hours | 6 ½ hours | -½ hour |
| Justin Opraseuth | 8 hours | 7 ½ hours | - ½ hour |
| Mohammed Alam | 6 hours | 6 hours | 0 hour |
| Total | 21 hours | 20 hours | -1 hour |

**12.2 Defect Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact or**  **Deliverable** | **Who (individual**  **or Team)** | **Estimated** | **Actual** | **Difference** |
| Initial SRS 1-6 | Pavandip Singh | 10 | 5 | -5 |
| Initial SRS 8-13 | Justin Opraseuth | 5 | 4 | -1 |
| Initial SRS 7,14 | Mohammed Alam | 10 | 7 | -3 |
| Initial SRS | Total | 25 | 16 | -9 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact or**  **Deliverable** | **Who (individual**  **or Team)** | **Estimated** | **Actual** | **Difference** |
| Final SRS 1-6 | Pavandip Singh | 5 | 3 | -2 |
| Final SRS 8-13 | Justin Opraseuth | 5 | 6 | 1 |
| Final SRS 7,14 | Mohammed Alam | 5 | 4 | -1 |
| Final SRS | Total | 15 | 13 | -2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact or**  **Deliverable** | **Who (individual**  **or Team)** | **Estimated** | **Actual** | **Difference** |
| SPMP 1-5.1 | Pavandip Singh | 8 | 6 | -2 |
| SPMP 6-11, 12.3 | Justin Opraseuth | 10 | 8 | -2 |
| SPMP 5.2-5.5,12.1-12.2 | Mohammed Alam | 10 | 7 | -3 |
| SPMP | Team | 28 | 21 | -7 |

**CUMULATIVE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Who (individual**  **or Team)** | **Estimated** | **Actual** | **Difference** |
| Pavandip Singh | 23 | 14 | -9 |
| Justin Opraseuth | 20 | 18 | -2 |
| Mohammed Alam | 25 | 18 | -7 |
| Total | 68 | 50 | -18 |

**12.3 Gantt Chart**

