**NYU Polytechnic School of Engineering**

**Computer Science and Engineering**

**eHarbinger**

**Software Project Management Plan**

**Version 1.0**

**Document Number: SPMP-001**

Project Team B3

Project Team Members:

Meghan Clark mc4677

Brian Marks bm1549

Priyam Nidhi pn613

Benson Tsai bt773

**Table Of Contents**

1 OVERVIEW.............................................................................................................

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

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

1.3 ASSUMPTIONS AND CONSTRAINTS...........................................................................................7

1.4 PROJECT DELIVERABLES.............................................................................................................7

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

2 EVOLUTION OF THE PLAN.............................................................................................................

3 REFERENCES.............................................................................................................

4 DEFINITIONS.............................................................................................................

5 PROJECT ORGANIZATION.............................................................................................................

5.1 EXTERNAL INTERFACES........................................................................................................

5.2 INTERNAL STRUCTURE.........................................................................................................

5.3 ROLES AND RESPONSIBILITIES..............................................................................................

6 MANAGEMENT PROCESSES.................................................................................................................

6.1 START-UP PLAN...................................................................................................................

6.1.1 Estimation Plan ....................................................................................................

6.1.2 Staffing Plan ........................................................................................................

6.1.3 Resource Acquisition Plan....................................................................................

6.1.4 Training Plan........................................................................................................

6.2 WORK PLAN.........................................................................................................................

6.2.1 Work Activities.....................................................................................................

6.2.2 Schedule Allocation .............................................................................................

6.2.3 Resource Allocation.............................................................................................

6.2.4 Budget Allocation ................................................................................................

6.3 CONTROL PLAN ...................................................................................................................

6.3.1 Requirement Control and Traceability.................................................................

6.3.2 Schedule Tracking and Adjustment.....................................................................

6.3.3 Budget Tracking and Adjustment.........................................................................

6.3.4 Quality Control.....................................................................................................

6.3.5 Reporting Mechanisms .......................................................................................

6.3.6 Metrics Collection Plan........................................................................................

6.4 RISK MANAGEMENT PLAN...................................................................................................

6.5 POST IMPLEMENTATION PLAN ...........................................................................................

7 TECHNICAL PROCESSES.......................................................................................................................

7.1 PROCESS MODEL ................................................................................................................

7.2 METHODS, TOOLS, AND TECHNIQUES.................................................................................

7.3 INFRASTRUCTURE PLAN......................................................................................................

7.4 PRODUCT ACCEPTANCE AND MIGRATION PLAN.................................................................

8 SUPPORTING PROCESSES PLANS..............................................................................................

8.1 CONFIGURATION MANAGEMENT PLAN..............................................................................

8.2 QUALIFICATION (VERIFICATION AND VALIDATION) PLAN ..................................................

8.3 DOCUMENTATION (LIBRARY) PLAN.....................................................................................

8.4 QUALITY ASSURANCE PLAN.................................................................................................

8.5 REVIEWS AND AUDITS ........................................................................................................

8.6 PROBLEM RESOLUTION PLANS ...........................................................................................

8.7 ENVIRONMENT MANAGEMENT PLANS...............................................................................

8.8 PROCESS IMPROVEMENT PLAN ..........................................................................................

9 ADDITIONAL PLANS ......................................................................................................................

10 INDEX..................................................................................................................................................

11 RATIONALE.................................................................................................................................

12 NOTES.............................................................................................................................................

13 APPENDICES.................................................................................................................................

13.1 SCHEDULE TRACKING........................................................................................................

13.2 DEFECT TRACKING ............................................................................................................

13.3 GANTT CHART/MICROSOFT PROJECT SCHEDULE .............................................................

**Section 1 - Overview**

This section is a general overview at the Project Management Plan for eHarbinger. It provides summary of the project, as well as purpose and objectives, assumptions and constraints, deliverables, schedule and evolution of the project.

**1.1 Project Summary**

As class time, work, and other responsibilities take a large amount out of the day for an avid gamer, they can find themselves not being able to have time to play along with their friends. Then once they have time to play alongside their friends, they might not be at the same level anymore and cannot truly enjoy playing together. Our team had decided on this idea for a project after seeing multiple other students in the School of Engineering come across this issue. A lot of students within the school like to play a popular multiplayer online game such as League of Legends. However sometimes it isn’t possible to get enough people on the same level to progress within the game. Our team believes that we will be able to solve this issue through our own matchmaking website.

This project is commenced to fulfill the requirement of CS-UY 4523, Design Project at NYU Polytechnic School of Engineering for fall 2015.

**1.2 Purpose, Scope and Objectives**

The purpose of this document is to outline milestones for the project based on analysis conducted on previous documents, and deliverables to be presented at each milestone.

The scope of this document is restricted to the core software development of the project, including front-end user interface, back-end database, server, as well as integration with third party websites. Other issues such as hardware availability is not within the scope of this document.

The objective of the document is to lay out deliverables requirements to be completed at each milestone; to ensure these deliverables are delivered on time and within budget.

**1.3 Assumption and Constraints**

The project is made possible based on the following assumptions:

* There is adequate interface or method with which the software can obtain product data and prices to integrate with our database.
* The team will work together to complete the project.

The project is planned and executed based on the following constraints:

* Commitment from each team member is limited to 10 hours per week due to other course commitments.
* Additional team member is not possible for the project.

**1.4 Project Deliverables**

The team will deliver a complete working system, meeting the expectations as specified in document SRS-001, system requirements specification. Pending on the progress of the team in the course, the final software product, as well as documentation associated will be delivered at the end of fourth quarter of 2015.

Because this is a web-based software platform, the final product will be delivered through the internet. Customer will be able to access the platform through a website, which will be ready on the date of delivery. Documentation will be delivered electronically; however, physical copies will also be provided.

**1.5 Schedule and Budget Summary**

The project milestones will follow the schedule below:

|  |  |
| --- | --- |
| **Artifact** | **Date** |
| Software Project Management Plan (SPMP) | 9/23/2015 |
| Requirements and Analysis Specification (RAS) | 9/30/2015 |
| Initial Software Design Document (SDD) | 10/21/2015 |
| Final Software Design Document (SDD) with code | 11/18/2015 |
| Implementation and Demonstration | 11/30/2015 |

**Section 2 - Evolution of the Plan**

As this is the first release of the plan, no information is available at this moment. However, should there be any amendment to the plan, a new version will be designated and released, and summaries of changes outlined in this section.

**Section 3 - References**

<http://www.giantbomb.com/>

This is a comprehensive website offering game information, reviews, news, as well as a forum for user discussions on strategies, gameplay and compatibilities of various games available in the market. It also has wiki feature that allows users to collaboratively edit gaming tips and tricks. This is a good resource for the team to design a user-friendly forum interface, and to structure it so users can find the subforum they need as quickly as possible.

<http://store.steampowered.com/search/>

This website lists all games available on steam, the largest game distribution platform. This can help the team develop better match-making method tailored to different styles of gaming.

**Section 4 - Definitions**

**C++:** An object oriented coding program that will be used to create the server for this project.

**COTS:** Commercial off-the shelf software. This software is available to the team and will cut down on time and cost of the product’s software.

**Database:** A set of data put together and created for easier access

**Democratic Team:** A team created by itself with egoless programming. The team can be extremely productive, especially if the problem is difficult and they are in a research environment.

**Evolution Tree Model:** A life-cycle model for software process, in which there are 4 phases of requirements, analysis, design, and implementation. At the end of each episode of phases there is a baseline of all artifacts, artifacts being work done in each phase.

**Hardware components:** Any physical piece needed for part of a computer. This can include anything from a motherboard to a mouse.

**PostgreSQL:** An open sourced database system that is available to anyone and created by a small group of volunteers.

**Server:** a piece of hardware that is used to access a resource in a network and keep a product access to the internet

**Section 5 - Project Organization**

**5.1 External Interfaces**

As a project initiated by the team itself, there is no client to report to at this moment. We have decided, however, to invite John Warder, our mutual friends, to provide input from a user perspective on any improvements that can be done to the product. Everyone in the team will be in communication with the customer consultant in periodic meetings. All the work will be done by the four team members who will work together during each workflow.

**5.2 Internal Structure**

The team will be a democratic team with a project manager and 4 programmers. The four programmers would be responsible for the database and server end of the product. The team would be a “linear” one with no hierarchies as the skill set possessed by the team members is comparable in terms of the technical knowledge. Each member, however, will be assigned to concentrate on specific components of the product in order to provide consistency in programming concepts and code documentation within certain stages.

**5.3 Roles and Responsibilities**

|  |  |
| --- | --- |
| **Name** | **Role** |
| Meghan Clark | Documentation Handler and User Interface Designer |
| Brian Marks | Project Manager and Architectural Designer |
| Priyam Nidhi | Server Designer and Lead QA/Tester |
| Benson Tsai | User Interface Designer and Chief Programmer |

**Section 6 - Management Processes**

**6.1 Start-up Plan**

The team has to deliver a product that meets the functional requirements specified by the client. In order to do so, the priorities need to be identified and a set schedule needs to be followed.

**6.1.1 Estimation Plan**

The estimation would be based on the breakdown of the tasks in the five workflows. Based on the project’s scope, there are higher-level tasks that can be further broken down within each workflow. Every low-level task will be delegated and estimated as follows:

1. Type of task: The task would be categorized as an inspection, meeting with client, general implementation in the workflow or just a task type.
2. For every task, we need estimates of the:

* Start and Finish Date
* Cost
* Effort Required
* Resources

Throughout the development plan, the budget would be allocated to internal operational costs and employee wages. The numbers would be estimated in the design phase based on the numbers of similar projects, given the same number of human capital available. In the implementation workflow, testing the compatibility of different hardware devices would account for a large portion of the money spent.

**6.1.2 Staffing Plan**

To be produced in a later document.

**6.1.3 Resource Acquisition Plan**

To be produced in a later document.

**6.1.4 Training Plan**

The staff is already trained in using CASE tools and has an adequate understanding of core software concepts required to set up the server and manage the front and back-end of eHarbinger’s website. A week in the analysis workflow needs to be given to train the staff about the attributes needed for the user profile that would make it more interactive and easy to use. In this, the staff would be trained on:

* Keeping the database(Postgres, Oracle) updated with the user profiles
* User Experience when designing the front end for the website

All other technical training has already been done by the staff. The team would hold a two-day mandatory session at the start of the project on "egoless programming" and conflict-resolution for all the four team members.

A day of training would cover “domain knowledge” in terms of impactful UX design; this would educate the team members on the specific user profiles being created and how to make them more impactful.

**6.2 Work Plan**

For the work plan, the project will use a schedule and resource allocation system to get the the tasks defined in its work activity done.

**6.2.1 Work Activities**

The overall project plan is given in Table 1 in the Appendix with the specified duration of each task. The plan indicates the major work activities and how they are interconnected for the entire duration of the project.

|  |  |  |
| --- | --- | --- |
| Duration | Activity | Status |
| Week 1 | Discussed the requirement artifacts with the client. Produced System Requirement Specification which was approved by the client. | Completed |
| Week 2, 3 | Produced the artifacts for analysis phase and discussed the consistency with the client's requirements. Worked on the Software Project Management Plan. | In Progress |
| Week 4, 5 | Produce and inspect design artifacts. | Upcoming |
| Week 6-8 | Implement the servers and perform testing. Check consistency throughout the documentation. | Upcoming |

**6.2.2 Schedule Allocation**

Refer to Section 12.3 for the Gantt chart explaining the time and milestones for the project.

**6.2.3 Resource Allocation**

|  |  |  |
| --- | --- | --- |
| **Work Activity** | **Team Members** | **Additional resources/factors** |
| Requirements | All 4 team members work together | Would need the understanding of the user/target segment |
| Analysis | 4 team members | Information should not be too technical at this stage |
| Design | 4 programmers | Produce the technical design. |
| Implementation | 4 programmers | Implement the design and monitor progress. |
| Testing | 3 programmers, 1 team member for editing the documentation | Perform thorough testing and document these changes to all the previous workflows. |

The team will work together for all the stages where the client's requirements will be reflected well in the analysis performed. In terms of meetings, there will be meetings at the end of every week where team members can raise potential queries. The potential users from the target group such as gamers from NYU’s Digital Media department can be contacted at the end of every workflow to make sure they are good with the progress and replication so far. The documentation needs to be updated each time a change is made to keep the information consistent.

**6.2.4 Budget Allocation**

To be produced in a later document.

**6.3 Control Plan**

In order to control the procedures required for the successful delivery of the product, all the sub-processes need to be monitored. These plans to be done incrementally and iteratively to ensure flexible improvements.

**6.3.1 Requirement Control and Traceability**

When changes are made to the requirements after the System Requirement Specifications(SRS) has been released, these changes need to be assessed. If it is possible to account for these changes without delaying the delivery date of the project, then the team can include these changes in the design. The time and resources needed to account for these additional changes should be decided by the team. We need to be accountable for:

* Traceability. Every artifact that is produced should be traced back to the requirements specification. The team would address this during review meetings as well as schedules inspections.
* Change control: All changes made after the System Requirement Specification(SRS) has been approved need to well documented and approved by all the team members. If this change is approved, then a revision needs to be made to the SRS.

**6.3.2 Schedule Tracking and Adjustment**

The project should move according to the set milestones on the Gantt chart where each phase has to be completed on the assigned date(Refer to section 12). In case the work in one workflow is not achieved by its set milestone, a part of the time from the successive milestone needs to be allocated to it. Special attention must be given to ensure that adjustment does not delay the completion of the present workflow. We need to measure the achievement of milestones to check if it conforms to the planned progress.

To ensure schedule tracking, the following methods will be used:

* Schedule performance reports: Each team member provides the status of the task allocated to them.
* Progress Variance Monitoring: Actual progress of project can vary from the planned progress, due to resource limitation and changes in requirements, these need to be monitored well.
* Progress Variance Resolution
* Corrective action to get back on schedule

**6.3.3 Budget Tracking and Adjustment**

Not needed in this document.

**6.3.4 Quality Control**

A software quality assurance plan has been made(Section 7.4) to monitor the quality of work processes and their products. There would be inspections in between the milestones to discuss the progress of the project and to detect faults in the work done so far. Inspecting the product's code would be a major way of detecting faults.

There would also be informal walkthroughs initiated by a team member in every phase to resolve any queries that might arise as the project progresses.

There is no separate software quality assurance(SQA) group in the project as all team members will be responsible for testing the production control requirements. In conducting these quality-adherence tests, the team would check that:

* Artifacts meet the set quality criteria
* Requirements should be traceable and verifiable
* Any defects in the earlier stages of the project should be immediately identified and removed.

**6.3.5 Reporting Mechanisms**

Not needed in this document.

**6.3.6 Metrics Collection Plan**

The metrics would involve recording the time spent on each of the project’s tasks. Later, during the implementation cycle, the defects need to be recorded as well. The three metrics that we will be working with are:

1. Project Metrics: Schedule, Cost, Productivity and Scope
2. Process Metrics: Phase-based effort, EDR per phase, Response time, Fixing quality
3. Product Metrics: Product Size, Cost Estimation, Quality Level

The team would follow the GQM(goal, question, metric) approach to set the metrics at conceptual, operational and quantitative levels. We need to set the team’s goals, establish related questions, and fix measurement parameters for these.

**6.4 Risk Management Plan**

There is no similar product similar to the functionality provided by eHarbinger. so it is not possible to test this new product with an existing one. The product should go over thorough and extensive testing to avoid any major flaws in the logic of its implementation. Since eHarbinger aims at interaction between users and focus on their profiles, the database of this information must be maintained well. It also needs to be updated with the latest registrations. The layout of the webpage needs to be user-friendly.

A thorough understanding of the problem needs to be present in order to come up with the most cost-effective solution. To get a deep understanding of the problem, the development team which comprises of gamers can interact with other gamers at the university and enquire about any challenges faced. During the design and implementation workflow, there is a high probability of a logical error in writing the code. The code needs to be reviewed by each team member to avoid any major design fault. The project must also be prepared for any hardware failure in the server. The data for the project needs to be backed up so that in case of any data erasure, no major documentation is lost and the project continues on time.

**6.5 Post Implementation Plan**

To be produced in a later document.

**Section 7 - Technical Processes**

**7.1 Process Model**

The life-cycle model used for this product is the evolution tree model. The project deliverables that are to be completed are listed below:

|  |  |
| --- | --- |
| Project proposal | **09/16/2015** |
| Software Project Management Plan (SPMP) | **09/23/2015** |
| Requirements and Analysis Documentation (RAS) | **09/30/2015** |
| Software Design Document (SDD) | **10/21/2015** |
| Design Document Final w/ Code | **11/18/2015** |

Each part of the project will be completed by the dates listed and seen as milestones for the projects. The day before each milestone, the documentations due will be reviewed by the team. In order for the project to be continued, each document that is posted must be approved and then fixed for any faults before it is posted again.

**7.2 Methods, Tools, and Techniques**

The databases will be designed in PostgreSQL, and contain a list of user profiles. The webpage itself will be designed in HTML, CSS, and Javascript. In order to complete the product on time, the work will be completed through divide-and-conquer, in which each problem will be divided into subproblems until the problem can be solved. Also, to help make each milestone, the product will be developed iteratively to allow for early testing, lower investment risks, and allow for short-term milestones to be completed within the team.

**7.3 Infrastructure Plan**

Each team member will be responsible for reviewing work of another. In order to test the webpage, the code will be copied into an open browser. If all the forum modeling, and services for the user worked are correct while testing then the team will move on to another aspect will be worked on. Each time a fault is detected in an artifact, a revision will be made. If there is a need for two different versions of the project in order to work for any operating system, then there would need to be multiple variations of the product. In order to keep all the work in order, a configuration control tool will be needed in which each baseline will be frozen to change artifacts and keep all originals/changes together.

**7.4 Product Acceptance and Migration Plan**

In order to test the product, it must be verified, making sure that the workflow has been done correctly, and validated, making sure the product meets all of the client’s needs. Each team member will have another review their work. This will be done through informal walkthroughs consisting of someone working on the current workflow and some team members set to work on the next workflow. Everyone will prepare by looking through the work done and seeing if there are any defects, or something not understood. Through the walkthrough, any defects found will be later corrected by the team members, but any solutions will be decided after the walkthrough to ensure every defect is gone through within a couple of hours. When the product is tested, the team will look at utility, how much of the client’s needs are met, reliability, how frequent and critical failures are, robustness, how the product works with incorrect inputs, performance, how much time and space the product takes up, and correctness, in which the outputs made work under each condition made.

**Section 8 - Supporting Processes Plans**

**8.1 Configuration Management Plan:**

In this subsection, details related to the configuration management of the project are shown. The configuration management, for the most part, will be handled by the team as a whole. Furthermore, the Software Configuration Management Plan (SCMP) will provide more information when it is created.

The configuration management tools for the documentation and code will be Google Docs and GitHub, respectively. These are used so that the team members may work on up-to-date documents and code at all times.

The identification for the documentation is provided ahead of time because each person in the group is assigned their own section to work on, followed by a final review of the document by the time as a whole. The identification tool for the code is provided in GitHub as a feature; whenever someone commits a piece of code, they must provide a comment showing what they changed and why they changed it.

The control of the documentation is provided by keeping older copies of each document in order to show the evolution of the document. The control of the code is provided by GitHub because GitHub tracks what changes are made and can revert if necessary.

Status accounting is done by reviewing the changes to the documentation or code.

Evaluation is done in a two-step process.

* The first step is for the person to check their own documentation or code and inspect it for any errors.
* The second step is for the team as a whole to read through the documentation and the code, if necessary, to look for any errors.

Release management for the documentation is determined by the due date of each document. Release management for the code will be decided by the team by implementing milestones; this is shown in Section 13.3.

Baselining is done procedurally because the product is being developed procedurally. The documentation is completed in the order it was assigned. When it comes to coding, if any faults are detected in the documentation, they are to be corrected as per the change control and then coding modifications are made. For documentation, when a baseline is completed, it is emailed to all team members before being submitted. For coding, baselines are to be emailed to all team members upon being completed.

Change requests, change control, and change tracking that are for substantial changes are done in a four-step process. Whether the change is substantial or not is up to the discretion of the requestor.

* The first step is an email is sent including all five team members requesting a certain change be made along with the reason for such changes, if the change is not a substantial change, it can be done immediately and step four may be completed, otherwise proceed to step two.
* The second step is a meeting time is arranged where all team members must attend.
* The third step is branched: if all team members unanimously agree to the change proceed to step four, otherwise, the change is discarded.
* The fourth step is to implement the change and document the status of the change as it is implemented.

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

In this subsection, the Verification and Validation (V&V) plans are shown. There will be more detail in the upcoming Software Verification and Validation Plan (SVVP). The team will do V&V individually and as a whole as needed.

The scope for the V&V of the project will be the:

* Software Requirements Specification (SRS)
* Software Project Management Proposal (SPMP)
* Software Analysis Document
* Software Design Document
* Database relational diagram
* All other documents
* Any and all software interfaces
* All other code

The standards tools for documentation and code will be used for V&V. A spell and grammar checker on Google Docs will be used for documentation and a web engine that supports PHP and JavaScript, such as Apache, will be used for the code.

The techniques used for V&V will be Black-box testing and Tracing. Black-box testing will be used to verify the quality of the code and tracing will be used to verify that the software matches the user requirements. Tracing will be done from User requirements to software requirements, from software requirements to interface requirements, from interface requirements to database requirements, and from acceptance tests to user requirements to create the full circle.

The responsibility of V&V will be to the team as a whole and the authors of said documents and code, specifically. The author will check over their own work after completion, then when they are ready, the team will look over the documents and codes for any extra errors.

Walkthroughs and peer reviews will be done periodically at the discretion of the author of the documentation or code. At the end of each milestone, reviews will be held in the form up stand-up meetings in order to answer the key questions: What did I accomplish? What will I accomplish? What obstacles stopped me? What obstacles will stop me?

Prototyping will be done at the end of each coding milestone and modeling will be used to extrapolate the prototype to its final form.

The techniques used for the validation plan will be testing, demonstration, analysis, and inspection. Testing will be done constantly throughout the software project development process and demonstration will be done on an as-needed basis. Analysis and inspection will be left to the end of each milestone’s work is completed.

**8.3 Documentation (library) Plan**

Described below is the documentation plan for each of the required documentation pieces.

The sources of information for all pieces of documentation should be from within the group or a previously agreed upon source, such as NewClasses. The generators and reviewers of the documents should be the members of the software development team.

For a piece of documentation to be presented, the following requirements must be met. The documentation should be:

* Tested by at least one other group member other than the author
* Reviewed by all team members
* Traced to previous documents and with future documents and code in mind
* If applicable, contain references to the associated architectural component it represents as well as its associated specifications

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Document Title | Document Generator | Document Reviewer | Document Approver | Date for Initial Version | Distribution List |
| Project proposal | Project Team | Project Team | Project Team | **09/16/2015** | Project Team |
| Software Project Management Plan (SPMP) | Project Team | Project Team | Project Team | **09/23/2015** | Project Team |
| Requirements and Analysis Documentation (RAS) | Project Team | Project Team | Project Team | **09/30/2014** | Project Team |
| Design Description (SDD) - Initial | Project Team | Project Team | Project Team | **10/21/2015** | Project Team |
| Software Design Document Final (w/Code) | Project Team | Project Team | Project Team | **11/18/2015** | Project Team |

**8.4 Quality Assurance Plan**

This subsection describes the Quality Assurance process for work products to insure that the quality is consistent for the entire project.

The scope of the Quality Assurance Plan is as follows:

* Software Project Management Proposal (SPMP)
* Requirements and Analysis Documentation (RAS)
* Software Design Document (SDD)
* All other documents
* Any and all software interfaces
* All other code
* Software binaries
* End-user documentation

In order to insure consistency, there will be testing, demonstration, analysis, and inspection. Testing will be done constantly throughout the software project development process and demonstration will be done on an as-needed basis. Analysis and inspection will be left to the end of each milestone. There will also be periodic reviews to make sure everything is on track to complete. At the end of each milestone, there will also be an audit an assessment done to verify the quality of the work of the software team.

The quality review sessions will be done in meetings where all of the software development team are in attendance. Reviews will be done to make sure documentation is correct and the code closely follows the documentation. The audit sessions will be appended at the end of the review sessions, so that all team members are present.

**8.5 Reviews and Audits Plan**

This subsection describes the schedule of review and audits along with the resources, methods, and procedures.

**Polytechnic University Management Reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review | Schedule | Resources | Method | Procedure |
| Proposal Review | **09/16/2015** | Associated documents in NewClasses site. Google Docs. | Verify all sections are completed according to document in NewClasses site. | 1. Submit document via NewClasses 2. Document is reviewed to see if it contains all required parts 3. Each part is checked for validity |
| SPMP Review | **09/23/2015** | Associated documents in NewClasses site. Google Docs. | Verify all sections are completed according to document in NewClasses site. | 1. Submit document via NewClasses 2. Document is reviewed to see if it contains all required parts 3. Each part is checked for validity |
| RAS Review | **09/30/2015** | Associated documents in NewClasses site. Google Docs. | Verify all sections are completed according to document in NewClasses site. | 1. Submit document via NewClasses 2. Document is reviewed to see if it contains all required parts 3. Each part is checked for validity |
| SDD Review (Initial) | **10/21/2015** | Associated documents in NewClasses site. Google Docs. | Verify all sections are completed according to document in NewClasses site. | 1. Submit document via NewClasses 2. Document is reviewed to see if it contains all required parts 3. Each part is checked for validity |
| SDD Review (final) | **11/18/2015** | Associated documents in NewClasses site. Google Docs. | Verify all sections are completed according to document in NewClasses site. | 1. Submit document via NewClasses 2. Document is reviewed to see if it contains all required parts 3. Each part is checked for validity |

**Developer Peer Reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review | Schedule | Resources | Method | Procedure |
| Requirements Peer Review | Weekly | Associated documents in NewClasses site. Google Docs. | Review status of document with all team members. | 1. Team schedules meeting time 2. Each team member discusses their current status and potential hardships 3. Questions are answered and group disperses. |
| Design Peer Review | Weekly | Associated documents in NewClasses site. Google Docs. | Review status of document with all team members. | 1. Team schedules meeting time 2. Each team member discusses their current status and potential hardships 3. Questions are answered and group disperses. |
| Implementation Peer Review | Weekly | Associated documents in NewClasses site. Google Docs. | Review status of document with all team members. | 1. Team schedules meeting time 2. Each team member discusses their current status and potential hardships 3. Questions are answered and group disperses. |

**Technical Reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review | Schedule | Resources | Method | Procedure |
| Technical Review | Bi-weekly | Google Docs. GitHub. | Discuss problems with coding or documentation. | 1. Discuss difficulties in coding or errors in documentation. 2. Discuss resolution to errors. |

**Walkthroughs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review | Schedule | Resources | Method | Procedure |
| Walkthrough | Weekly | Associated documents in NewClasses site. Google Docs. GitHub. | Discuss problems with coding or documentation. | 1. Discuss difficulties in coding or errors in documentation. 2. Discuss resolution to errors. |

**Inspections**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review | Schedule | Resources | Method | Procedure |
| Walkthrough | Weekly | Associated documents in NewClasses site. Google Docs. GitHub. | Discuss problems with coding or documentation. | 1. Discuss difficulties in coding or errors in documentation. 2. Discuss resolution to errors. |

**Audits**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review | Schedule | Resources | Method | Procedure |
| Document and Coding Review | Weekly | Associated documents in NewClasses site. Google Docs. GitHub. | Review deliverable documents and source code to determine if it meets requirements. | 1. Discuss difficulties in coding or errors in documentation. 2. Discuss resolution to errors. |

**8.6 Problem Resolution Plans**

The purpose of this subsection is to enumerate the resources, methods, tools, techniques, and procedures to report, analyze, prioritize, and process software problem reports.

The resources available are Google Docs and GitHub as well as the associated documents in NewClasses. In order to report problems, email will be used where all members of the group are in the conversation. The priority of the problem will be determined by the software development group as a whole.

These methods to be used are to review the documents and code and to discuss problems and desired changes. The analyzation and prioritization of such methods will be decided by the software development group as a whole.

The tool being used would be Apcahe. It will be used to execute code for backend development on the project. In addition to Apache, some frontend development will be created through Sublime Text Editor. Any and all problems with coding will be dealt with using Apache or Sublime by a designated member of the software development group. This is a high priority task.

The techniques being used are Black-box testing and tracing. Black-box testing will be done at the end of any task and will be a high priority task. Tracing will be a top priority task to insure that the software product closely matches the requirements.

**8.7 Environment Management Plans**

Environment for development will be collaborative with each person’s work available to anyone else on the team. In addition to having all the work to be reviewed by each other online, there will also be meeting times in which the team can put their work together to solve a problem faced in development. Initial testing will be done by the team, and later testing will be done by a focus group collected. This focus group will be given access to the web application and will have a feedback form to tell the team what is positive and negative about the project.

**8.8 Process Improvement Plan**

At each team meeting, each team member must discuss problems they are facing in their part of the project. Then the team will try to determine the root of the problem and find the best solution; this is essential as it is possible that one team member’s problem is caused by other team member’s work. In addition to these meetings, the team and materials will be available online in case there is an immediate problem within the project.

After each focus group, the team will meet after to determine problems found in the project by different users. Once these problems are identified, those in charge of each part of the project will improve their part for another focus group before the web application will be made online for everyone to access.

**Section 9 - Additional Plans**

Provisions must be made to accommodate other obligations. Some weekly meetings can be missed given the circumstances of the software development group.

**Section 10 - Index**

Please see table of contents for the location of information.

**Section 11 - Rationale**

This Software Project Management Plan (SPMP) lays down the foundation for the proposed project. It gives a very detailed overview of every contributing factor to the product. Included in the SPMP are the definitions of what’s to be used for our system. For example, the life cycle model, development language, and database system are all specified and defined. Estimation costs are outlined specifically for every workflow of the product. By referencing the given costs throughout the workflows, the team can stay within the budget and designated time. Training plans are outlined as well. These training plans are implemented so the staff workers will be familiar with all the systems and layouts that deal with our product. A well trained staff will result in the best production and development of the product by making sure everything works as intended. The resource allocation subsection shows a detailed scheduling of the resources to be used during the given workflows. By referencing this guideline during each workflow, the team can finish work on time without over exhausting resources.

The traceability subsection states that version control will be used with every new variation to pre existing documents or new documents. Version control is used to log all versions of documents, and every log should be able to be traced back to the requirements. Scheduling and tracking methods such as Progress Variance Monitoring and Progress Variance Resolution are used to meet milestones and deadlines. Quality control and risk management are dealt with through continuous testing of the software and hardware. Continuous testing makes sure our system is reliable and correct according to the specified requirements. Product acceptance is determined by conducting walkthroughs throughout every workflow. After all the faults are found for that given workflow, team members will go on to revise and correct those faults. Continuous walkthroughs will result in continual revisions and improvements, thus helping make sure the product is acceptable for the client.

The configuration management plan outlines how the documents will be documented and logged. All documents should be able to be traced. The documentation plan specifies how documents should be reviewed by team members and tested thoroughly, to ensure quality, The quality assurance plan makes sure the product is tested at every workflow and that the product is acceptable for the client. Walkthroughs, scheduled reviews, and audits are all used to achieve this. The problem resolution plan states the methods, tools, and techniques that are used in the workflows to implement our product. Schedule and defect tracking are to used to keep logs and metrics of all previous and current documents.

By adhering to these guidelines, our eHarbinger product can be implemented. Quality is insured due to meticulous planning, scheduling, and testing. There is a great demand for this product, and by adhering to this Software Project Management Plan, successfully implementing this product is feasible.

**Section 12 - Notes**

None.

**Section 13 - Appendices**

**13.1 Schedule Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated  (Hours) | Actual | Difference |
| Project Proposal | Brian | 5 | 4 | 1 |
| Project Proposal | Priyam | 2 | 4 | 2 |
| Project Proposal | Meghan | 3 | 2 | 1 |
| Project Proposal | Benson | 3 | 4 | 1 |
| Project Proposal | Team | 13 | 14 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated  (Hours) | Actual | Difference |
| SPMP | Brian | 2 | 2 | 0 |
| SPMP | Priyam | 3 | 3 | 0 |
| SPMP | Meghan | 4 | 3 | 1 |
| SPMP | Benson | 3 | 2 | 1 |
| SPMP | Team | 12 | 10 | 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated  (Hours) | Actual | Difference |
| RAS | Brian | 3 | NA | NA |
| RAS | Priyam | 3 | NA | NA |
| RAS | Meghan | 5 | NA | NA |
| RAS | Benson | 3 | NA | NA |
| RAS | Team | 14 | NA | NA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated  (Hours) | Actual | Difference |
| SDD Initial | Brian | 3 | NA | NA |
| SDD Initial | Priyam | 3 | NA | NA |
| SDD Initial | Meghan | 3 | NA | NA |
| SDD Initial | Benson | 3 | NA | NA |
| SSS Initial | Team | 12 | NA | NA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated  (Hours) | Actual | Difference |
| SDD Final | Brian | 5 | NA | NA |
| SDD Final | Priyam | 3 | NA | NA |
| SDD Final | Meghan | 4 | NA | NA |
| SDD Final | Benson | 4 | NA | NA |
| SDD Final | Team | 16 | NA | NA |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or Team) | Estimated  (Hours) | Actual | Difference |
| Brian | 18 | NA | AN |
| Priyam | 14 | NA | NA |
| Meghan | 19 | NA | NA |
| Benson | 16 | NA | NA |
| Team | 67 | NA | NA |

**13.2 Defect Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated | Actual | Difference |
| Project Proposal | Brian | 3 | 3 | 0 |
| Project Proposal | Priyam | 4 | 4 | 0 |
| Project Proposal | Meghan | 5 | 5 | 0 |
| Project Proposal | Benson | 5 | 5 | 0 |
| Project Proposal | Team | 19 | 19 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated | Actual | Difference |
| SPMP | Brian | 2 | NA | NA |
| SPMP | Priyam | 4 | NA | NA |
| SPMP | Meghan | 6 | NA | NA |
| SPMP | Benson | 4 | NA | NA |
| SPMP | Team | 18 | NA | NA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated | Actual | Difference |
| RAS | Brian | 3 | N/A | N/A |
| RAS | Priyam | 2 | N/A | N/A |
| RAS | Meghan | 4 | N/A | N/A |
| RAS | Benson | 3 | N/A | N/A |
| RAS | Team | 16 | N/A | N/A |

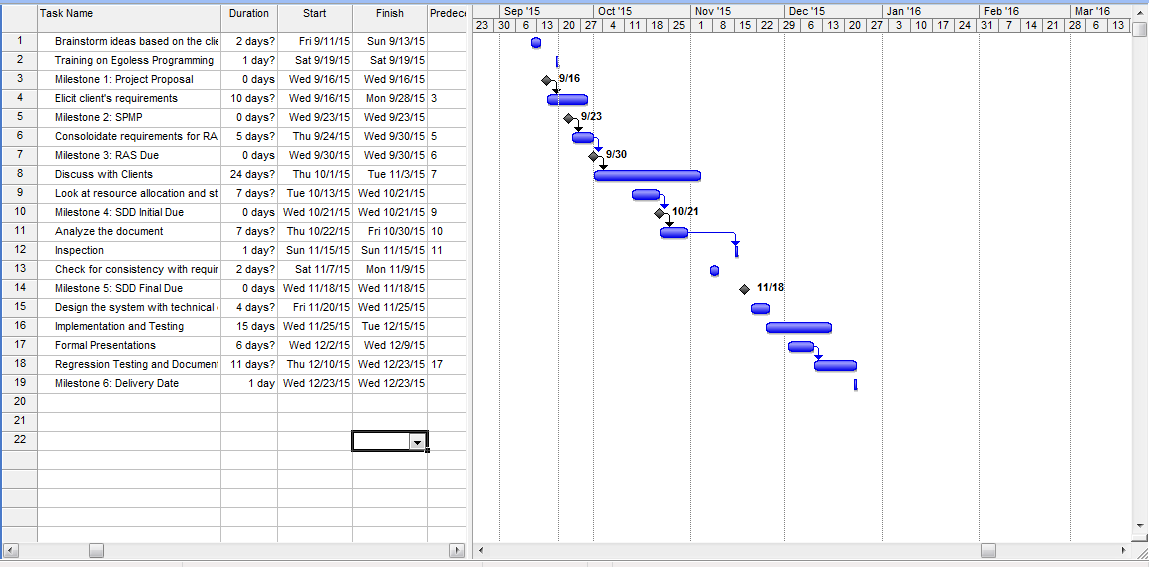
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated  (Hours) | Actual | Difference |
| SDD Initial | Brian | 4 | NA | NA |
| SDD Initial | Priyam | 3 | NA | NA |
| SDD Initial | Meghan | 4 | NA | NA |
| SDD Initial | Benson | 3 | NA | NA |
| SDD Initial | Team | 14 | NA | NA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or  Deliverable | Who (individual  or Team) | Estimated  (Hours) | Actual | Difference |
| SDD Final | Brian | 3 | NA | NA |
| SDD Final | Priyam | 4 | NA | NA |
| SDD Final | Meghan | 3 | NA | NA |
| SDD Final | Benson | 4 | NA | NA |
| SDD Final | Team | 14 | NA | NA |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or Team) | Estimated | Actual | Difference |
| Brian | 15 | N/A | N/A |
| Priyam | 17 | N/A | N/A |
| Meghan | 22 | N/A | N/A |
| Benson | 19 | N/A | N/A |
| Team Summary | 73 | N/A | N/A |

**13.3 Gantt Chart/Microsoft Project Schedule**



|  |  |
| --- | --- |
| **Tasks** | **Time Span** |
| **Brainstorm ideas based on the client** | **9/11/15 - 9/13/15** |
| **Milestone 1: Project Proposal** | **9/16/15** |
| **Training on Egoless Programming** | **9/19/15** |
| **Elicit client's requirements** | **9/16/15 - 9/28/15** |
| **Milestone 2: SPMP** | **9/23/15** |
| **Consolidate requirements for RAS** | **9/24/15 - 9/30/15** |
| **Milestone 3: RAS Due** | **9/30/15** |
| **Discuss with Clients** | **10/1/15 - 11/3/15** |
| **Look at resource allocation and start assembling SSD** | **10/13/15 - 10/21/15** |
| **Look at resource allocation and start assembling the SPMP** | **10/23/15 - 11/8/15** |
| **Milestone 4: SSD Initial Due** | **10/21/15** |
| **Analyze the document** | **10/22/15 - 10/30/15** |
| **Check for consistency with requirements** | **11/7/15 - 11/9/15** |
| **Inspection** | **11/15/15** |
| **Milestone 5: SDD Final Due** | **11/18/15** |
| **Design the system with technical details** | **11/20/15 - 11/25/15** |
| **Implementation and Testing** | **11/25/15 - 12/15/15** |
| **Regression Testing and Documentation-Updates** | **12/10/15 - 12/23/15** |
| **Milestone 6: Delivery Date** | **12/23/15** |