

**Computer Science and Engineering**

**NYU Student Club Event Planner**

**Software Project Management Plan**

**Version 1.0**

Document Number: SPMP-001

Project Number: A10

Project Members:

Gryczka, Patrick

Han, Qiao Yang

Huang, Alan

Papa, Kostaq

**REVIEW AND APPROVALS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Printed Name**  **and Title** | **Function (Author, Reviewer, Approval)** | **Date** | **Signature** |
| Professor Strauss | Instructor | April 10, 2014 |  |
| Patrick Gryczka | Author | April 10, 2014 | On file |
| Qiao Yang Han | Author | April 10, 2014 | On file |
| Alan Huang | Author | April 10, 2014 | On file |
| Kostaq Papa | Author | April 10, 2014 | On file |

**REVISION LEVEL**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
| **April 9, 2014** | **Version 1.0** | **Initial Release** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**TABLE OF CONTENTS**

**1. INTRODUCTION............................................................................................1**

**1.1 PROJECT SUMMARY...................................................................................1**

**1.2 PURPOSE, SCOPE, AND OBJECTIVES………………………………………….......1**

**1.3 ASSUMPTIONS AND CONSTRAINTS………………………………………..........1**

**1.4 PROJECT DELIVERABLE…………………………………………………………..........2**

**1.5 SCHEDULE AND BUDGET SUMMARY……………………………………………...2**

**1.6 EVOLUTION OF THE PLAN……………………………………………………………...3**

**2. REFERENCES................................................................................................3**

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

**4. PROJECT ORGANIZATION............................................................................3**

**4.1 EXTERNAL INTERFACES………………………………………………….................3**

**4.2 INTERNAL STRUCTURE……………………………………………………...............4**

**4.3 ROLES AND RESPONSIBILITIES……………………………………………….........4**

**5. MANAGEMENT PROCESSES........................................................................4**

**5.1 START-UP PLAN…………………………………………………………….................4**

**5.1.1 ESTIMATION PLAN………………………………………………...................4**

**5.1.2 STAFFING PLAN………………………………………................................4**

**5.1.3 RESOURCE ACQUISITION PLAN……………………………………............5**

**5.1.4 TRAINING PLAN……………………………………………………………..........5**

**5.2 WORK PLAN……………………………………………………………………..............5**

**5.2.1 WORK ACTIVITIES…………………………………………….......................5**

**5.2.2 SCHEDULE ALLOCATION………………………………...........................6**

**5.2.3 RESOURCE ALLOCATION…………………………………........................8**

**5.3 CONTROL PLAN………………………………………………………......................9**

**5.3.1 REQUIREMENT CONTROL AND TRACEABILITY………..................9**

**5.3.2 SCHEDULE TRACKING AND ADJUSTMENT……………...................10**

**5.3.3 BUDGET TRACKING AND ADJUSTMENTS..................................10**

**5.3.4 QUALITY CONTROL…………………................................................10**

**5.3.5 REPORTING MECHANISMS.......................................................10**

**5.3.6 METRICS COLLECTION PLAN…………………………..........................11**

**5.4 RISK MANAGEMENT PLAN……………………………………......................11**

**5.5 POST IMPLEMENTATION PLAN………………………...............................11**

**6. TECHNICAL PROCESSES..............................................................................11**

**6.1 PROCESS MODEL.....................................................................................11**

**6.2 METHODS, TOOLS, AND TECHNIQUES.....................................................12**

**6 .3 INFRASTURCTURE PLAN..........................................................................13**

**6.4 PRODUCT ACCEPTANCE AND MIGRATION PLAN……………………….............13**

**7. SUPPORTING PROCESSES PLANS...............................................................13**

**7.1 CONFIGURATION MANAGEMENT PLAN..................................................13**

**7.2 QUALIFICATION (VERIFICATION AND VALIDATION) PLAN.......................13**

**7.3 DOCUMENTATION (LIBRARY) PLAN........................................................13**

**7.4 QUALITY ASSURANCE PLAN....................................................................14**

**7.5 REVIEWS AND AUDITS............................................................................14**

**7.6 PROBLEM RESOLUTION PLANS...............................................................14**

**7.7 ENVIRONMENT MANAGEMENT PLANS...................................................14**

**7.8 PROCESS IMPROVEMENT PLAN..............................................................14**

**8. ADDITIONAL PLANS..................................................................................15**

**9. INDEX.......................................................................................................15**

**10. REQUIREMENTS TRACEABILITY...............................................................15**

**11. RATIONALE.............................................................................................15**

**12. APPENDICES...........................................................................................15**

**12.1 SCHEDULE TRACKING………………………….......................................15**

**12.2 DEFECT TRACING…………………......................................................16**

**1. OVERVIEW**

**1.1 Project Summary**

The motivation for creating the NYU Student Club Event Planner stems from the dichotomy between how well established NYU student clubs are and how poorly they are able to communicate their actions to the student body. This document is meant to cement the management planning for the implementation of this software project. The intended audience of this document are both the project managers and the investors of this project, the NYU student clubs.

**1.2 Purpose, Scope, and Objectives**

The purpose of the NYU Student Club Event Planner is to create a better means of communication between the clubs at NYU and the NYU student body. This software proposes to fill this perceived deficit with a social infrastructure that should be able to both connect students with each other and keep them informed on events.

All activities directly related to the purpose are considered to be within the scope. All activities not directly related to the purposes are considered to be out of scope. For example, issues concerning the finance and administration of student club are not within the scope of this project.

The objectives of the project are as follows:

* + Complete the project by the project due date
  + Complete the project within budget
  + Provide all deliverables identified in Section 1.3 by the project due date
  + Fulfill all stated requirements, as in the SRS, of the software product deliverable, which fall into one of the following categories:
    - Internal database modifications
    - Web pages of user interface
    - Student club management
    - Club event management

**1.3 Assumptions and Constraints**

The project will be planned with the following assumptions:

* + This project will be Linux Based using Apache Server.
  + That student clubs at NYU will be willing to pay a fee to host their club information on our site and make their clubs more accessible to the student body
  + Initial estimates for the project as provided in this SPMP are +/- 40%.

The project will be planned with the following constraints:

* + Budget
    - $2,000
  + Time
    - Six months
  + Staff
    - The outside consultants from student clubs will be required to assist in the implementation phase of the project.
  + Maintenance
    - The project will have to be designed such that maintenance expenses do not exceed $400 per year.

**1.4 Project Deliverables**

**Deliverables**  **Due Dates:**

* + Software Requirement Specification (SRS) - 4/8/2014
  + Software Project Management Plan (SPMP) - 4/9/2014
  + Software Analysis Specification (SAS) - 4/14/2014
  + Software Design Document (SDD) - 4/28/2014

**1.5 Schedule and Budget Summary**

The project has the following high-level schedule:

* + Delivery of baseline project plan:
  + Software products ready for operation:

The project has a budget of $2,000. Once the software product is delivered, annual maintenance costs should be no larger than $400.

**1.6 Evolution of the Plan**

The plan is considered to be a dynamic document and will be updated monthly by default and on an unscheduled basis as necessary. Scheduled updates to the plan will occur once every month on the last business day of the month.

Notification of scheduled and unscheduled updates to the plan will be communicated via e-mail to all project participants according to the Reporting Plan

Once the initial plan is finalized, a baseline of the plan will be created. Changes to the plan will take place against this baseline. The plan will only receive further baselines if significant change in scope occurs.

**2. REFERENCES**

NYU Student Clubs Event Planner, Team A10, Project Proposal, Version 1.0, 4/1/2014

NYU Student Clubs Event Planner, Team A10, System Requirement Specifications, Version 1.0, 4/4/2014

**3. DEFINITIONS**

DH – Development Head

NYU SCEP – NYU Student Club Event Planner

SQA – Software Quality Assurance

**4. PROJECT ORGANIZATION**

**4.1 External Interfaces**

The biggest external interface we will be communicating with the student clubs that will be using our product. They will feed us data such as clubs' names, clubs' administration teams, clubs' events, and clubs' member data. Our application will sort this data out in a way that end users can browse them clearly.

**4.2 Internal Structure**

Our software development team will make use of the chief programmer team structure. This structure may change early on in the development cycles if necessary.

**4.3 Roles and Responsibilities**

* Programming: Entire team
* Testing: Patrick Gryczka and Qiao Yang Han
* Software Quality Assurance: Kostaq Papa
* Chief Programmer: Qiao Yang Han
* Backup Programmer: Kostaq Papa
* Programming Secretary: Patrick Gryczka
* Post-delivery Maintenance: Kostaq Papa

**5. MANAGEMENT PROCESS**

**5.1 Start-up Plan**

**5.1.1 Estimation Plan**

Microsoft Project is used to keep track of the milestones, the deliverables of each iteration, and the team meetings. Microsoft Word is used to document and modify the requirement specifications, and the project management plan. The cost of the project is determined when the client signs off on the requirements specifications document. The requirements specification document is a contract between the development team and the client.

**5.1.2 Staffing Plan**

Currently the development team includes one project manager, four developers who will be programming the project and performing quality assurance. If a team member were to withdraw from the project for any reason, a new person will added to the team through an interview process. All members of the team must have a strong programming background. They must be familiar with one or more object oriented programming languages such as Java, C++, and C#. They must be familiar with database concepts, and web development technologies such as HTML, CSS, and JavaScript.

**5.1.3 Resource Acquisition Plan**

Currently, there is no need to hire any new members to the team.

All the software and hardware is provided by NYU School of Engineering. The lab stations will hold the code, database, and web servers needed for the project. If any additional resources are needed, Professor Fred Strauss will be contacted.

Personal information such as the emails of all the current NYU students will need to be obtained from the university registrar.

**5.1.4 Training Plan**

Necessary courses will be available to members of the team if they are not familiar with any concepts. Courses in Java, C++, object oriented programming, data structures, databases, and web development will be available. As a result, everyone in the team will be able to utilize the concepts involved in three-tier architecture (user interface, application server, and database).

**5.2 Work Plan**

**5.2.1 Work Activities**

1. Project Proposal

1.1 Write and Review Proposal

1.2 Submit Proposal

2. Requirements Specification Document

2.1. Initial SRS

2.2 Review and correct SRS

2.3 Final SRS

2.4 Submit SRS

3. Software Project Management Plan

3.1 Initial SPMP

3.2 Review and Correct SPMP

3.3 Submit SPMP

4. Requirements/Analysis Document

4.1 Initial Requirements/Analysis Document

4.2 Review and Correct RAS

4.3 Submit RAS

5. Design Document

5.1 Initial Design Document

5.3 Review Design Document

5.4 Submit Design Document

6. Final Design

6.1 Correct and modify initial design document

6.2 Review Design Document

6.3 Submit final design

7. Implementation

7.1 Implement all components of the system

7.2 Conduct unit and product testing

7.3 Submit source code

8. Implementation Demo

8.1 Practice implementation demo

8.2 Conduct demo

**5.2.2 Schedule Allocation**

|  |  |  |  |
| --- | --- | --- | --- |
| Task: | Start Date: | End Date: | Due Date: |
|  |  |  |  |
| Initial Project Proposal | February 19, 2014 |  | March 24, 2014 |
| Write and Review Proposal | February 19, 2014 | March 05, 2014 | March 24, 2014 |
| Submit Proposal |  |  | March 24, 2014 |
| Initial Software Requirements Specification | March 06, 2014 |  | March 24, 2014 |
| Initial SRS | March 06, 2014 | March 10, 2014 | March 24, 2014 |
| Review and Correct SRS | March 10, 2014 | March 24, 2014 | March 24, 2014 |
| Final SRS |  | March 24, 2014 | March 25, 2014 |
| Submit SRS |  |  | March 24, 2014 |
| New Initial Project Proposal | March 24, 2014 |  | March 24, 2014 |
| Write and Review New Project Proposal | March 24, 2014 | March 25, 2014 | March 24, 2014 |
| Submit New Project Proposal | March 25, 2014 |  | March 24, 2014 |
| New Initial Software Requirement Specification | March 25, 2014 |  | March 25, 2014 |
| New Initial SRS | March 25, 2014 | March 31, 2014 | March 25, 2014 |
| Review and Correct New SRS | March 31, 2014 | April 4, 2014 | March 25, 2014 |
| New Final SRS | April 4, 2014 |  | March 25, 2014 |
| Submit New SRS |  | April 4, 2014 | March 25, 2014 |
| Software Management Plan | March 17, 2014 |  | March 24, 2014 |
| Initial SPMP | April 05, 2014 | April 09, 2014 | March 24, 2014 |
| Review and Correct SPMP | April 09, 2014 | April 10, 2014 | March 24, 2014 |
| Submit SPMP |  | April 10, 2014 | March 25, 2014 |
| Requirements/Analysis Documentation | April 05,2014 |  | April 15, 2014 |
| Initial Requirements/Analysis Documentation | April 05, 2014 | April 12, 2014 | April 15, 2014 |
| Review And Correct RAS | April 12, 2014 | April 14, 2014 | April 15, 2014 |
| Submit RAS |  | April 15, 2014 | April 15, 2014 |

**5.2.3 Resource Allocation**

|  |  |  |  |
| --- | --- | --- | --- |
| WBS | Task Name | Resource | Team Member |
| 1 | **Project Proposal** |  |  |
| 1.1 | Write and Review Proposal | Microsoft Word | All members |
| 1.2 | Submit Proposal |  | Any member |
| 2 | **Requirements Specification Document** |  |  |
| 2.1 | Initial SRS | Microsoft Word | All members |
| 2.2 | Review and correct SRS | Microsoft Word | All members |
| 2.3 | Final SRS | Microsoft Word | All members |
| 2.4 | Submit SRS |  | Any member |
| 3 | **Software Project Management Plan** |  |  |
| 3.1 | Initial SPMP | Microsoft Word | All members |
| 3.2 | Review and Correct SPMP | Microsoft Word | All members |
| 3.3 | Submit SPMP |  | Any member |
| 4 | **Requirements/Analysis Document** |  |  |
| 4.1 | Initial Requirements/Analysis Document | Microsoft Word | All members |
| 4.2 | Review and Correct RAS | Microsoft Word | All members |
| 4.3 | Submit RAS |  | Any member |

**5.3 Control Plan**

**5.3.1 Requirement Control and Traceability**

Any changes that affect the schedule of the project have to be reviewed and approved by all members of the development team and documented in the NOTES section of this document, as well as trace back any related changes to the SRS. The traceability mentioned before will be accomplished by numbering all requirements.

**5.3.2 Schedule Tacking and Adjustment**

During meetings, team members will discuss their status and any issues that they encountered in order to resolve them. In addition, members must alert one another of any addition problems through email. All team members are responsible for staying on schedule. The overall progress of the team will be measured against the schedule elaborated in the SPMP. Also, the work produced will be evaluated for its compatibility with the defined requirements. If the initial duration of a task is prolonged from the predicted schedule, the entire team will be responsible for finding reasons for the prolonged duration and adjust early on by either reassigning tasks to other programmers or modifying the predicted schedule. The schedule tracking will be performed thoroughly during the entire development of the product including after each milestone is completed.

**5.3.3 Budget Tracking and Adjustment**

Equipment and hosting costs will be handled early on in the project and should match predictions quite closely. Man hour costs are predicted to be a much more likely source of variance. Costs will be analyzed at the end of each project milestone and the budget will be predicted accordingly. Initial adjustments for going over budget on man hours will involve analyzing work practices to see if anything can be changed to increase streamlining, and maximize efficiency. If man hours continue to be over budget, adjustments will be made to the features. Features will need to be evaluated for expandability.

**5.3.4 Quality Control**

The quality of the product will be maintained by having peer and group reviews. During group reviews, the item (not including code) being reviewed will be read out loud, and the team will find faults and correct them. Code will be reviewed as a group as well as self-reviewed. Faults will be discovered during reviews, but they will be fixed at a later date.

**5.3.5 Reporting Mechanisms**

Reporting will be done through a standardized form, which will standardize required information for reports and allow for quicker appraisal of reports.

**5.3.6 Metrics Collection Plan**

Metrics for the assessment of the project will be collected after each task has been completed and after a milestone has been reached and analyzed by the entire team. Those metrics include the duration of a task or milestone. The duration of the task or milestone will be constantly checked against initial projections. The efforts put into the task or milestone will be another collected metrics. The last metric collected will be the quality of the task or milestone. This quality will be evaluated relatively to the number of faults per line of code.

**5.4 Risk Management Plan**

Throughout the software development process many risks may arise. One risk is that a team member may withdraw from the class before the withdrawal deadline. This may lead to a resource acquisition plan. Another risk is having a fellow team member get sick for a prolonged duration, resulting in one of the active members taking the workload of that team member. Hardware failure is also a risk to consider; therefore, frequent backups should be conducted.

**5.5 Post Implementation Review**

The project’s materials, including documents and codes, will be archived and made available to the team for maintenance. The post implementation review will take place once the project goes live. The review will include an assessment of the overall project as well as an assessment of the effectiveness of the project team, duration tracking, and the communication mechanisms; in addition, lessons will be learned from the assessment for future projects.

**6. TECHNICAL PROCESSES**

**6.1 Process Model**

The development process model for this project is Agile Unified Process. Agile Unified Process allow developers to enter the implementation stage early. As we start working on the actual project, we will get a clearer picture of the project design. Then we can quickly update the changes occurred in the requirement and the analysis. Our project requires strong collaboration between developers and student club representatives. Agile Unified Process allows frequent interactions between developers and clients. Feedbacks from student club representatives is important to our project design and implementation.

In the process of implementation, the project is broken into subsystems and each subsystem is broken into several components. Every two weeks, a milestone is set. Each team member is assigned to complete some of the components. One subsystem of the project should be completed in two weeks. Each subsystem is tested to ensure its correct matching to the design and efficiency, before we move to the next milestone. After completing the milestones, we integrate the subsystems into a whole. A comprehensive testing is done on the system. This system will then be the baseline for the next feature to add onto.

In the beginning of the construction phrase, a version control tool account will be setup to allow collaboration among developers. The coding environment and programming tools will be installed and configured. The developers will meet up twice every week to communicate.

After the completion of the project, the product will then be released to the client.

**6.2 Methods, Tools, and Techniques**

The Agile Unified Process is our development model. We develop the project in small components using github. The methodology used to develop this project is object-oriented software design. Our project involves building high-availability heavy-duty dynamic web sites. We decide to use LAMP software bundle. Linux is the operating system and Apache HTTP Server is the web server. The internal database is written in MySQL. PHP is the server-side scripting language that will be used to handle server side processing, creating dynamic web pages that are generated from information within a database. The web pages of user interface is written in HTML and javascript. Javascript allows client-side scripts to interact with the user on the browser. CSS is used for the front-end design. C++ will be used to create an API to the MySQL database. The C++ API consists of classes correspond to the data in the database. PHP can extract information from the database through the API.

The Standards for programming is set. Use liberally blank line within the code for readability. Use indentation organize the code. Write comments to clarify the functionality of certain methods. Do not exceed 120 characters in a single line. Use bracket immediately after the declaration of class. For example,

Class A{

};

Do not use goto statement.

**6.3 Infrastructure Plans**

The team has three Linux workstations available for this project. The Apache HTTP Server and MySQL server are also available in each Linux workstations. Microsoft Visual Studio 2010 are also available in each Linux workstations. Github is a web-based hosting service for software development projects. We Github for file sharing, file editing and synchronization.

**6.4 Product Acceptance and Migration Plan**

Every milestone of the project will be accepted formally by the client by signing appropriate acceptance documentation. At the end of every phase the client will install the product and perform an acceptance test. This may result in additional requests for change and improvements. The details about the change plan are given in Section 5.3.1

**7. SUPPORTING PROCESSES PLANS**

**7.1 Configuration Management Plan**

Version control will be handled by the synchronization features of dropbox. These features will keep our files up to date and ensure that each team member’s project files are the same. Identification will be self-maintained. Developers will be responsible for updating a log file when they make changes to the project. This log file should contain the time of changes the purpose of changes and their location within the project.

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

See Section 6.4.

**7.3 Documentation (Library) Plan**

Documentation will begin before code is written. Within the design process, documentation will be written for each necessary function in the system. These function documentations are intended to aid in function implementation and will be extended as necessary during implementation.

**7.4 Quality Assurance Plan**

QA will be testing throughout the lifecycle of the NYU SCEP project. It will begin with logical tests on the systems design, and continue into practical testing once implementation begins.

QA will have a standardized defect reporting method to make defect reports easier to understand.

**7.5 Reviews and Audits**

Reviews will mostly take the form of inspections and walkthroughs. Inspections will take place during every milestone, and whenever non-trivial defects are found and fixed by the SQA team and the DHs. Walkthroughs will be more common. Walkthroughs should occur at least once a month with additional ones scheduled whenever defect occurrences increase by more than 25% between two weeks. In addition to inspections and

**7.6 Problem Resolution Plans**

Defects found and reported by the SQA team will be sent to a subdivision of the development team responsible for verifying that the reported defects are in fact defects and not a misinterpretation on the part of the SQA team. Once verified as a problem the defects will be addressed in a group meeting of all DHs and those developers responsible for the modeling and planning of the software. Any defects deemed trivial or having to do with implementation rather than the quality of the design will be sent straight to the developers to fix, while issues with the software systems design will be more seriously analyzed by the design team to solve without causing larger problems.

Once corrections have been made and tested they will be appropriately documented and the working software will be updated to include the changes.

**7.7 Environment Management Plans**

There will be three environments. A design management environment, an implementation environment, and a testing environment. Design management will be for keeping the system design up to date and verifying the logic behind the design. The implementation environment will be where coding is done. And the testing environment is where QA will manage testing of the implementation.

**7.8 Process Improvement Plan**

As stated in section 7.6, in addition to regularly scheduled inspections and walkthroughs, we will be performing additional reviews and audits whenever tracked defect occurrences increase by more than 25% between weeks. This practice is intended to identify problems within our development cycle and within the NYU SCEP’s design, and allow the team to solve them.

**8. ADDITIONAL PLANS**

None as of the writing of this document.

**9. INDEX**

**10. RATIONALE**

The reasoning behind this system, is that the social dynamics of NYU can be improved by improving the communication between clubs and their current and potential members. We believe that by providing a centralized platform of access to club activities for students, we can make it easier for students to keep up with the club events and meetings and hopefully increase participation

**11. NOTES**

**12. APPENDICES**

**12.1 Schedule Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact or Deliverable** | **Who** | **Estimated** | **Actual** | **Difference** |
| SPMP | Patrick Gryczka | 4 hours | 5.5 hours | 1.5 hours |
| SPMP | Qiao Yang Han | 4 hours | 4.0 hours | 0.0 hours |
| SPMP | Alan Huang | 4 hours | 2.5 hours | 1.5 hours |
| SPMP | Kostaq Papa | 4 hours | 4.0 hours | 0.0 hours |
| SPMP | Summary | 16 hours | 16 hours | 3.0 hours |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| **Who** | **Estimated** | **Actual** | **Difference** |
| Patrick Gryczka | 4 hours | 5.5 hours | 1.5 hours |
| Qiao Yang Han | 4 hours | 4.0 hours | 0.0 hours |
| Alan Huang | 4 hours | 2.5 hours | 1.5 hours |
| Kostaq Papa | 4 hours | 4.0 hours | 0.0 hours |

**12.2 Defect Tracking**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Artifact or Deliverable** | **Who** | **Estimated** | **Actual** | **Difference** | **Number of Defects Detected** |
| SPMP | Patrick Gryczka | 5 | 4 | 1 | 7 |
| SPMP | Kevin Han | 5 | 3 | 2 | 4 |
| SPMP | Alan Huang | 5 | 2 | 3 | 3 |
| SPMP | Kostaq Papa | 5 | 4 | 1 | 5 |
| Total Defects |  |  |  |  | 19 |

**Cumulative**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Who** | **Estimated** | **Actual** | **Difference** | **Number of Defects Detected** |
| Patrick Gryczka | 5 | 4 | 2 | 7 |
| Qiao Yang Han | 5 | 3 | 1 | 4 |
| Alan Huang | 5 | 2 | 3 | 3 |
| Kostaq Papa | 5 | 4 | 2 | 5 |
| Total Defects |  |  |  | 19 |