**Project Management Plan**

version 0.5

PAS Customization Project

**Team 7**

Stephanie Greene (Team Leader)

Matthew Boydston

Eric DeShazer

Javier Ochoa

David Rivera

SE 4485.001

Software Engineering Project

Fall 2013

**Document Version Control Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Purpose/Change | Author | Date |
| 0.1 | Initial draft | Eric DeShazer | 9/8/2013 |
| 0.2 | Added Sprint Iteration Pointing Meeting, adjusted the number of Weekly Scrum Meetings, adjusted duration of Friday team meetings, added work and educational info for Stephanie and David. | Eric DeShazer | 9/9/2013 |
| 0.3 | Added work and educational info for Javier, updated the Table of Contents | Eric DeShazer | 9/10/2013 |
| 0.4 | Added work and educational info for Matthew, relocated the Document Version Control Table, updated the Table of Contents | Eric DeShazer | 9/13/2013 |
| 0.5 | Updated new weekly meeting days, added IEEE reference, included details on expected team behavior, updated the Table of Contents, added Configuration Mgmt to section 7.0 | Eric DeShazer | 9/13/2013 |

Abstract

The PAS (Patient Accounting System) Customization Project was commissioned by MedAssets to provide a greater level of control to customers using the Charge Capture Audit application. The purpose of the PAS Customization Project is to provide an easy to use interface through which MedAssets customers can customize the structure of the corrected charge export data file from the Charge Capture Audit application, as well as generate and download the export file.

**Table of Contents**

Section 1.0 – Introduction 5

Section 2.0 – Project Organization 6

Section 3.0 – Lifecycle Model 7

Section 4.0 – Risk Analysis 8

Section 5.0 – Hardware and Software Resource Requirements 13

Section 6.0 – Deliverables, Schedule 14

Section 7.0 – Monitoring, Reporting, and Controlling Mechanisms 20

Section 8.0 – Professional Standards 22

References 24

**Section 1.0 - Introduction**

The PAS (Patient Accounting System) Customization Project was commissioned by MedAssets to provide a greater level of control to customers utilizing one of their flagship applications, Charge Capture Audit. Charge Capture Audit provides automated auditing of inpatient and outpatient billing to improve billing accuracy and increase customer revenues. An important feature of Charge Capture Audit is the ability to export corrected charge data from the MedAssets system for consumption by the customer’s internal record keeping system. This data is exported in a CSV file. Because different customers use different record keeping systems, the structure of this CSV file must be tailored to each customer’s requirements. Customization of the export data file is currently performed by MedAssets staff.

The purpose and scope of the PAS Customization Project is to provide an easy to use interface through which MedAssets customers can customize the structure of the corrected charge export data file as well as generate and download the resulting CSV file. The final deliverable will be a stand-alone Windows application that will include the interface and a backend database of patient records from which the data will be exported. The programming language used to develop the interface and backend database connections will be Java. The database will be created using the MySQL relational database management system. Project team members will collaborate to plan the project, identify system components, allocate resources, and create the individual project deliverables. The project must be completed in 15 weeks.

The delivered application will provide a model to implement a new standard feature of the Charge Capture Audit application. The delivered interface must allow each customer to specify the name of the export data file, the number of fields to export, the order of the fields, how a debit or credit is represented, whether column headers are included, and finally to download the customized export data file. Each customer must have the ability to save multiple instances of their preferences.

Agile software development techniques will be utilized to ensure the successful completion of the project. Each phase of the project is documented in this Project Management Plan. The phases include Pre-Sprint Requirements Gathering, Sprint Iterations (Design, Implementation, Validation, Documentation), and Customer Delivery. The overall project will be managed using the Scrum framework for agile software development. Each iteration of this Scrum project is documented in Section 6.0 – Deliverables, Schedule.

**Section 2.0 - Project Organization**

*This section describes the way in which the development team is organized, the people involved and their roles on the project, and the rationales in assigning roles.*

**2.1 – Developers**

2.1.1 – Matthew Boydston

*Experience:*

Matthew has experience in writing code in Java, C++, and C. Additionally, Matthew has experience using Sencha Touch and Ext JS frameworks for developing web based and mobile applications. Matthew held a summer internship position at Sabre Holdings where he worked on the End to End Quality Assurance team. As part of the internship program, Matthew competed in a week long hackathon as a Project Manager/Developer to develop a web application. This gave Matthew working experience with applying the concepts he had learned throughout his coursework at the University of Texas at Dallas.

2.1.2 – Eric DeShazer

*Experience:*

Eric has extensive experience authoring code in Java, C, C++, and MIPS Assembly for project deliverables meeting the requirements of University of Texas at Dallas Software Engineering courses. He has working experience developing applications in Java and MicroChip PIC Assembly languages for his current employer, ThyssenKrupp Elevator Company. Eric also has experience in many aspects of Software Engineering, including requirements analysis, system and software architecture and design, testing, validation, as well as training and technical documentation.

2.1.3 - Stephanie Greene

*Experience:*

Stephanie currently works in the software QA department for Beck Technology. She has extensive experience using Scrum for the last 2.5 years in her current position at Beck. In addition, she is the software QA lead and liaison for Beck’s QA team in India. Stephanie has extensive experience authoring code in Java, C, C++, and C# in her work environment as well as meeting the requirements of University of Texas at Dallas Software Engineering Courses. Stephanie also has recent experience in database implementation, having completed a database course last semester.

2.1.4 - Javier Ochoa

*Experience:*

Javier has experience coding in Java, C++, Python, and MIPS Assembly for project deliverables meeting the requirements of University of Texas at Dallas Software Engineering courses. Javier also has recent experience in database implementation, having completed a database course last year. He has working experience as a Project Manager, working through the software development life cycle using Waterfall process for projects on the Reference Data Systems and Governance (RDS&G) department for the Consuming Banking Technology and Operations (CBT&O) line of business of Bank of America.

2.1.5 – David Rivera

*Experience:*

David has extensive experience programming in C/C++, and Java languages. He also has experience using the Visual Studio IDE and recent experience with Mysql, having completed a database course last semester. In addition, David has Android development experience from an advanced requirements project and client-server application development from advanced software architecture.

**2.2 – Product Owner / Project Manager / Project Leader**

Stephanie Greene

*Rationale:*

Stephanie’s experience using Scrum techniques in her workplace make her the ideal candidate for Product Owner as well as Project Leader.

**2.3 – Scrum Master**

Javier Ochoa

*Rationale:*

Javier’s experience with his current employer, his education, and his communication skills made him a good choice for Scrum Master.

**2.4 – Requirements Analysts**

2.4.1 – Matthew Boydston

*Rationale:*

Matthew’s experience from his Requirements Engineering course project as well as his experience from his internship hackathon made him a good choice for Requirements Analyst.

2.4.2 – David Rivera

*Rationale:*

David’s experience in his advanced requirements project made him a good choice for Requirements Analyst.

**2.5 – Testers**

2.5.1 – Matthew Boydston

*Rationale:*

To support the Agile techniques chosen by the group, we felt it necessary to have a single tester for every developer.

2.5.2 – Eric DeShazer

*Rationale:*

To support the Agile techniques chosen by the group, we felt it necessary to have a single tester for every developer.

2.5.3 - Stephanie Greene

*Rationale:*

To support the Agile techniques chosen by the group, we felt it necessary to have a single tester for every developer.

2.5.4 - Javier Ochoa

*Rationale:*

To support the Agile techniques chosen by the group, we felt it necessary to have a single tester for every developer.

2.5.5 – David Rivera

*Rationale:*

To support the Agile techniques chosen by the group, we felt it necessary to have a single tester for every developer.

**2.5 – Technical Writers**

2.5.1 – Eric DeShazer

*Rationale:*

Eric’s experience authoring technical manuals for his current employer made him a good choice for Technical Writer.

2.5.2 – Javier Ochoa

*Rationale:*

Javier’s experience with Bank of America made him a good choice for Technical Writer.

2.5.3 – David Rivera

*Rationale:*

David’s experience in his advanced requirements project made him a good choice for Technical Writer.

**2.5 – Trainers**

2.5.1 – Matthew Boydston

*Rationale:*

Matthew has done extensive work in new member education and officer transitions with a student organization on campus which made him a good choice for Trainer.

2.5.2 – Javier Ochoa

*Rationale:*

Javier’s work and educational experience made him a good choice for Trainer.

**Section 3.0 - Lifecycle Model**

*This section describes the lifecycle model used, including rationale*

The Scrum framework for agile software development was chosen as the life cycle model for this project because this is the preferred method utilized by MedAssets in their development projects and multiple team members have working experience in this model. Those of us experienced in the use of Scrum feel it is also best suited to this project because it will aid our team in handling many of the project unknowns.

The implementation of the various Scrum processes is as follows:

* Initial Customer Contact
  + Team gathers high level use cases and requirements. The Product Owner creates the Product Backlog. (lasting 2 weeks)
* Sprints
  + There will be 3 Sprints, each iteration lasting 4 weeks
    - Each Sprint begins with a Sprint Planning Meeting and a Sprint Iteration Pointing Meeting.
    - During each 4 week iteration period, there will be a Weekly Scrum Meeting. There will also be a Weekly Customer Meeting with the project mentor.
    - Each Sprint ends with Sprint Review and Scrum Retrospective Meetings
* Demonstration of functional features
  + In the wrap up phase of each Sprint, if functional features are complete, they will be demonstrated to the customer to elicit feedback.
* After the last Sprint is complete, the Team will use the last week remaining to finalize the customer deliver package and present the completed project to MedAssets and UT Dallas.

**Section 4.0 - Risk Analysis**

*This section describes the possible project risks, the likelihood of these risks arising, and the risk reduction strategies proposed.*

**4.1 - Incorrect Specifications/Scope**

*Description:* The completed product fails to achieve the customer's intended requirements.

*Likelihood:* Low

*Impact:* High

*Detection Difficulty:* Medium

*Anticipation Plan:*

- Intensive feasibility study and stakeholder interviews will attempt to clarify the project scope and requirements to minimize doubt

- The project's status will be evaluated at each milestone to ensure that the completed progress is within the project scope and adheres to the defined requirements

*Contingency Plan:*

- If the project's progress at any milestone strays from the scope of the project, the requirements and specifications should be reevaluated and the completed modules should be reconsidered.

**4.2 – Scope Creep**

*Description:* The requirements or features of the project grow and interfere with meeting project deadlines.

*Likelihood:* Low

*Impact:* Medium

*Detection Difficulty:* Low

*Anticipation Plan:*

- The Product Backlog and Sprint Backlogs will be continuously monitored and compared with the project deadlines to ensure timely completion of activities.

- The estimations of completion times have been increased 30%

- Requested additional features will be thoroughly discussed and evaluated with the customer to ensure successful project completion according to the original deadline.

*Contingency Plan:*

- Schedule extra time for team members

- Re-allocate other team members time to complete needed activities

- Defer additional features to a future project

**4.3 – Schedule Creep**

*Description:* The project's planned value exceeds its earned value at any point in time.

*Likelihood:* Medium

*Impact:* Dependent on schedule variance

*Detection Difficulty:* Low

*Anticipation Plan:*

- The project's schedule variance will be updated frequently at milestones in the schedule to encourage timely completion of activities

- The estimations of completion times have been increased 30%

- Weekly Scrum Meetings will be held to ensure team member contributions are according to the project plan

- Performance will be evaluated by a committee comprised of the Product Owner and all Team members and quick resolutions to issues will be top priority

*Contingency Plan:*

- Schedule extra time for team members

- Re-allocate other team members time to complete needed activities

- Continual poor performance by any Team member will be made known to UT Dallas staff for additional evaluation

**4.4 – Poor Validation Testing**

*Description:* Project features are inadequately tested, resulting in a poorly performing product.

*Likelihood:* Medium

*Impact:* High

*Detection Difficulty:* Medium

*Anticipation Plan:*

- Conduct thorough code walk-throughs after code completion

- Perform unit tests and correct errors discovered

- Design test cases with the purpose of finding errors

- Ensure all tests are based on detailed use cases and traceable to customer requirements

*Contingency Plan:*

- Schedule time for additional testing

**4.5 - Lack of Necessary Skills**

*Description:* The expertise of the team members limits the progress of activities and interferes with meeting the project requirements.

*Likelihood:* Low

*Impact:* Medium

*Detection Difficulty:* Medium

*Anticipation Plan:*

- Team member’s responsibilities have been assigned through a selective process that accounts for scholastic success, previous project experience, and thorough evaluations of work history

*Contingency Plan:*

- Schedule extra time for team members

- Re-allocate other team members time to complete needed activities

- Assign an experienced team member to provide training to less experienced team member.

**4.6 - Lack of Necessary Resources**

*Description:* Available hardware, software, and technical support limits the progress of activities and interferes with meeting the project requirements.

*Likelihood:* Low

*Impact:* Medium

*Detection Difficulty:* Medium

*Anticipation Plan:*

- The required hardware, software, and available resources will be assessed during the feasibility phase and addressed before any progress is made on the project itself

*Contingency Plan:*

- Consult with UT Dallas and MedAssets staff to acquire necessary resources

**4.7 - Hardware or Software Malfunctions**

*Description:* Problems with hardware or software interfere with meeting project requirements.

*Likelihood:* Low

*Impact:* Medium

*Detection Difficulty:* High

*Anticipation Plan:*

- Utilize reliable equipment and applications, measured by experience

- Frequently back up data

- Utilize configuration management tools to ensure that untainted work can be restored to a known revision level

- Identify additional hardware to be used in case of failure

*Contingency Plan:*

- If identified additional hardware are not sufficient, consult with UT Dallas and MedAssets staff to acquire necessary resources

**4.8 – Poorly Implemented Technical Documentation**

*Description:* Technical documentation does not properly explain functional project features.

*Likelihood:* Medium

*Impact:* Medium

*Detection Difficulty:* Low

*Anticipation Plan:*

- Technical writers and software developers work closely together to ensure accurate and complete documentation

- Peer review and revision of technical documents is performed before finalization

- Regularly review technical documentation with MedAssets staff

*Contingency Plan:*

- Utilize outside party to review alignment between documents and functional project features

**Section 5.0 - Hardware and Software Resource Requirements**

*This section describes the hardware and software required to carry out this project*

**5.1 – Hardware (5 desktop computers)**

5.1.1 – Processor: Intel Pentium processor or equivalent running at 233 Mhz or faster

5.1.2 – RAM: 128MB or greater available system RAM

5.1.3 – Hard Disk: 20GB or greater available hard disk space

5.1.4 – Input Devices: standard keyboard and Microsoft compatible mouse

5.1.5 – Video: video adapter and monitor with Super VGA (800x600) or higher resolution

5.1.6 – Sound card: standard PC

5.1.7 – Speakers or headphone: standard PC

**5.2 – Software**

5.2.1 – Operating System: Windows XP Professional or better

5.2.2 – Internet browser: Internet Explorer 8, Firefox, Google Chrome, or equivalent compatible with the resident OS.

5.2.3 – Integrated Debugging Environment – Latest Netbeans, Eclipse, or equivalent IDE compatible with the resident OS.

5.2.4 – Programming Language: Latest Java JRE and SDK compatible with the resident OS.

5.2.5 – RDBMS: Latest production version of MySQL compatible with the resident OS.

**Section 6.0 - Deliverables, Schedule**

*This section describes the activities, dependencies between activities, the estimated time required to reach each milestone, and the allocation of people to activities.*

**Start of Project**

*Start date:* August 30, 2013.

*Planned completion date:*  December 13, 2013.

**6.1 – Pre-Sprint Requirements Gathering**

6.1.1 - Stakeholder interview

*Person(s):* Product Owner, Development Team

*Predecessor(s):* Start of Project

*Duration:* 9/4/13 - 9/4/13

*Description:*

Meet with the stakeholder and gather all of the necessary requirements and specifications. Incongruities between the initial project presentation and the stakeholder's intended product should be addressed here.

6.1.2 - Perform feasibility study

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.1.1

*Duration:* 8/30/13 - 9/6/13

*Description:*

Determine the technical feasibilities of the project. Gather necessary information and knowledge to understand the problem and the resources necessary to provide a solution.

6.1.3 - Define project scope

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.1.2

*Duration:* 9/7/13 - 9/9/13

*Description:*

Define the project scope, including an analysis of the problem, feasibility, and general constraints that apply.

6.1.4 - Develop user stories, create product backlog

*Person(s):* Product Owner, Analyst 1, Analyst 2

*Predecessor(s):* 6.1.3

*Duration:* 9/10/13 - 9/13/13

*Description:*

Develop high level user stories that affirm the stakeholder’s goals, formally define the product backlog, and construct project plan targeted towards the stakeholder for further review.

**6.2 – Sprint One**

6.2.1 – Sprint Planning Meeting

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.1.4

*Duration:* 30 min - 9/13/13

*Description:*

Product Owner and Development Team meet to plan the tasks and deliverables of the first Sprint. The Product Owner presents the Product Backlog to the Development Team. Sprint Backlog (goals) is created.

6.2.2 – 4 Weekly Sprint Iteration Pointing Meetings

*Person(s):* Product Owner, Scrum Master, Development Team

*Predecessor(s):* 6.2.1 and previous Weekly Sprint Iteration Pointing Meeting

*Duration:* 30 min - 9/16/13, 9/23/13, 9/30/13, 10/7/13

*Description:*

User stories are assigned points according to the effort required for implementation. Points are used to allocate resources.

6.2.3 – Implement user stories, create tests and documentation

*Person(s):* Development Team

*Predecessor(s):* 6.2.2

*Duration:* 9/16/13 - 10/14/13

*Description:*

Developers implement user stories in collaboration with testers and tech writers to produce tested documented code that satisfies the goals of the stakeholders. Automated tests are developed from realistic examples of user goals. Implemented and tested user stories are delivered to the tech writers upon completion so that technical documentation can be created.

6.2.4 – 4 Weekly Customer Meetings

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.2.1 and previous Weekly Customer Meeting

*Duration:* 1 hour - 9/18/13, 9/25/13, 10/2/13, 10/9/13

*Description:*

Team members report their progress since the last Weekly Customer Meeting, discuss plans for tasks to be completed by the next Weekly Customer Meeting, demo any completed user stories, and elicit advice and feedback from the project mentor.

6.2.5 – 4 Weekly Scrum Meetings

*Person(s):* Product Owner, Scrum Master, Development Team

*Predecessor(s):* 6.2.1 and previous Weekly Scrum

*Duration:* 20 min - 9/23/13, 9/30/13, 10/7/13, 10/14/13

*Description:*

Team members report their progress since the last Weekly Scrum, discuss plans for tasks to be completed by the next Weekly Scrum, and discuss obstacles that stand in the way of progress on the Sprint Backlog. Scrum Master facilitates meeting and removes any obstacles impeding the team’s progress.

6.2.6 – Sprint Review Meeting

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.2.1 and all previous Weekly Meetings

*Duration:* 1 hour, 10/14/13

*Description:*

Team members present completed user stories to the Product owner. Sprint Backlog is updated according to the successful completed deliverables. Unfinished tasks are shifted to the next iteration.

6.2.7 – Scrum Retrospective Meeting

*Person(s):* Scrum Master, Development Team

*Predecessor(s):* 6.2.6

*Duration:* 30 min - 10/14/13

*Description:*

Team members and the Scrum Master discuss how the Team can adjust the development process to make it more effective and fulfilling for the next Sprint.

**6.3 – Sprint Two**

6.3.1 – Sprint Planning Meeting

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.2.7

*Duration:* 30 min - 10/14/13

*Description:*

Product Owner and Development Team meet to plan the tasks and deliverables of the first Sprint. The Product Owner presents the Product Backlog to the Development Team. Sprint Backlog (goals) is created.

6.3.2 – 4 Weekly Sprint Iteration Pointing Meetings

*Person(s):* Product Owner, Scrum Master, Development Team

*Predecessor(s):* 6.3.1 and previous Weekly Sprint Iteration Pointing Meeting

*Duration:* 30 min - 10/14/13, 10/21/13, 10/28/13, 11/4/13

*Description:*

User stories are assigned points according to the effort required for implementation. Points are used to allocate resources.

6.3.3 – Implement user stories, create tests and documentation

*Person(s):* Development Team

*Predecessor(s):* 6.3.2

*Duration:* 10/14/13 - 11/11/13

*Description:*

Developers implement user stories in collaboration with testers and tech writers to produce tested documented code that satisfies the goals of the stakeholders. Automated tests are developed from realistic examples of user goals. Implemented and tested user stories are delivered to the tech writers upon completion so that technical documentation can be created.

6.3.4 – 4 Weekly Customer Meetings

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.3.1 and previous Weekly Customer Meeting

*Duration:* 1 hour - 10/16/13, 10/23/13, 10/30/13, 11/6/13

*Description:*

Team members report their progress since the last Weekly Customer Meeting, discuss plans for tasks to be completed by the next Weekly Customer Meeting, demo any completed user stories, and elicit advice and feedback from the project mentor.

6.3.5 – 4 Weekly Scrum Meetings

*Person(s):* Product Owner, Scrum Master, Development Team

*Predecessor(s):* 6.3.1 and previous Weekly Scrum

*Duration:* 20 min - 10/21/13, 10/28/13, 11/4/13, 11/11/13

*Description:*

Team members report their progress since the last Weekly Scrum, discuss plans for tasks to be completed by the next Weekly Scrum, and discuss obstacles that stand in the way of progress on the Sprint Backlog. Scrum Master facilitates meeting and removes any obstacles impeding the team’s progress.

6.3.6 – Sprint Review Meeting

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.3.1 and all previous Weekly Meetings

*Duration:* 1 hour, 11/11/13

*Description:*

Team members present completed user stories to the Product owner. Sprint Backlog is updated according to the successful completed deliverables. Unfinished tasks are shifted to the next iteration.

6.3.7 – Scrum Retrospective Meeting

*Person(s):* Scrum Master, Development Team

*Predecessor(s):* 6.3.6

*Duration:* 30 min, 11/11/13

*Description:*

Team members and the Scrum Master discuss how the Team can adjust the development process to make it more effective and fulfilling for the next Sprint.

**6.4 – Sprint Three**

6.4.1 – Sprint Planning Meeting

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.3.7

*Duration:* 30 min - 11/11/13

*Description:*

Product Owner and Development Team meet to plan the tasks and deliverables of the first Sprint. The Product Owner presents the Product Backlog to the Development Team. Sprint Backlog (goals) is created.

6.4.2 – 4 Weekly Sprint Iteration Pointing Meetings

*Person(s):* Product Owner, Scrum Master, Development Team

*Predecessor(s):* 6.4.1 and previous Weekly Sprint Iteration Pointing Meeting

*Duration:* 30 min - 11/11/13, 11/18/13, 11/25/13, 12/2/13

*Description:*

User stories are assigned points according to the effort required for implementation. Points are used to allocate resources.

6.4.3 – Implement user stories, create tests and documentation

*Person(s):* Development Team

*Predecessor(s):* 6.4.2

*Duration:* 11/11/13 - 12/9/13

*Description:*

Developers implement user stories in collaboration with testers and tech writers to produce tested documented code that satisfies the goals of the stakeholders. Automated tests are developed from realistic examples of user goals. Implemented and tested user stories are delivered to the tech writers upon completion so that technical documentation can be created.

6.4.4 – 4 Weekly Customer Meetings

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.4.1 and previous Weekly Customer Meeting

*Duration:* 1 hour - 11/13/13, 11/20/13, 11/27/13, 12/4/13

*Description:*

Team members report their progress since the last Weekly Customer Meeting, discuss plans for tasks to be completed by the next Weekly Customer Meeting, demo any completed user stories, and elicit advice and feedback from the project mentor.

6.4.5 – 4 Weekly Scrum Meetings

*Person(s):* Product Owner, Scrum Master, Development Team

*Predecessor(s):* 6.4.1 and previous Weekly Scrum

*Duration:* 20 min - 11/18/13, 11/25/13, 12/2/13, 12/9/13

*Description:*

Team members report their progress since the last Weekly Scrum, discuss plans for tasks to be completed by the next Weekly Scrum, and discuss obstacles that stand in the way of progress on the Sprint Backlog. Scrum Master facilitates meeting and removes any obstacles impeding the team’s progress.

6.4.6 – Sprint Review Meeting

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.4.1 and all previous Weekly Meetings

*Duration:* 1 hour, 12/9/13

*Description:*

Team members present completed user stories to the Product owner. Sprint Backlog is updated according to the successful completed deliverables. Unfinished tasks are shifted to the next iteration.

6.4.7 – Scrum Retrospective Meeting

*Person(s):* Scrum Master, Development Team

*Predecessor(s):* 6.4.6

*Duration:* 30 min, 12/9/13

*Description:*

Team members and the Scrum Master discuss how the Team can adjust the development process to make it more effective and fulfilling for the next Sprint.

**6.5 - Customer Delivery**

6.5.1 – Finalize customer delivery package

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.4.7

*Duration:* 12/9/13 - 12/11/13

6.5.2 - Present delivery package to customer

*Person(s):* Product Owner, Development Team

*Predecessor(s):* 6.5.1

*Duration:* 1 hour – 12/11/13

*Description:*

Team members demonstrate final deliverables to customer, provide training in use of application and backend database, deliver technical documentation, and gather feedback from customer.

**6.6 – Senior Design Day**

6.6.1 – Final Project Report

*Person(s):* Product Owner

*Predecessor(s):* 6.5.2

*Duration:* 15 min – 12/13/13

*Description:*

Final project report is presented to UT Dallas staff.

6.6.2 – Final Project Demonstration

*Person(s):* Product Owner

*Predecessor(s):* 6.5.2

*Duration:* 15 min – 12/13/13

*Description:*

Final project demonstration for UT Dallas staff is performed.

**Section 7.0 - Monitoring, Reporting, and Controlling Mechanisms**

*This section describes the management reports that will be produced, the frequency of reporting, and the project monitoring and control mechanisms used.*

**7.1 – Project Management Plan**

*Frequency:*

Once

*Rationale:*

The Project Management Plan provides a template for the execution of the project and a means to evaluate the overall execution of the project upon completion.

**7.2 – Product Backlog**

*Frequency:*

Once before the 1st Sprint and updated/maintained by the Product Owner after each Sprint is complete.

*Rationale:*

The Product Backlog is produced initially to provide as complete a list of project tasks as possible at the beginning of the project. Subsequent Sprint Backlogs are dependent on this list for their content.

**7.3 – Sprint Planning Meeting Report**

*Frequency:*

Every 4 weeks

*Rationale:*

The Sprint Planning Meeting Report consists of the minutes of the Sprint Planning Meeting and is used to determine the success of the Sprint during the Sprint Review Meeting.

**7.4 – Sprint Backlog**

*Frequency:*

Every 4 weeks

*Rationale:*

The Sprint Backlog is produced to provide a complete list of tasks for each Sprint. The content of subsequent Sprint Backlog items is dependent on the tasks completed from each previous Sprint Backlog.

**7.5 – Weekly Scrum Meeting Report**

*Frequency:*

Weekly

*Rationale:*

The Weekly Scrum Meeting Report consists of the minutes of the Weekly Scrum Meeting and is used to document progress and next steps, as well as obstacles encountered, on a weekly basis.

**7.6 – Weekly Customer Meeting Report**

*Frequency:*

Weekly

*Rationale:*

The Weekly Customer Meeting Report consists of the minutes of the Weekly Customer Meeting and is used to document progress and next steps, customer demonstrations and feedback, as well as project mentor advice, on a weekly basis.

**7.7 – Sprint Review Meeting Report**

*Frequency:*

Every 4 weeks

*Rationale:*

The Sprint Review Meeting Report consists of the minutes of the Sprint Review Meeting and is used to document the progress and success of the Sprint.

**7.8 – Scrum Retrospective Meeting Report**

*Frequency:*

Every 4 weeks

*Rationale:*

The Scrum Retrospective Meeting Report consists of the minutes of the Scrum Retrospective Meeting and is used to document the progress, next steps, areas needing improvement, and success of the Team’s implementation of Scrum processes.

**7.8 – Final Project Report**

*Frequency:*

Once

*Rationale:*

The Final Project Report provides detailed explanation of the project execution and the final status of all project deliverables.

**7.9 – Configuration Management with GitHub**

*Frequency:*

Continuous version control of all documents and code

*Rationale:*

To ensure restorability of all jointly developed project deliverables, GitHub was chosen.

**Section 8.0 - Professional Standards**

*This section describes the expected behavior of the team members related to scholastic dishonesty, meeting schedule and quality expectations for tasks and deliverables.*

**8.1 – Standards of Behavior**

8.1.1 – Scholastic Honesty:

All team members agree to act with integrity and professionalism and at all times obey and uphold the currently posted rules of scholastic honesty at UT Dallas.

8.1.1 – Teamwork:

All team members agree to respect other team members’ opinions, and to ensure that all criticism is constructive and targeted at the idea not the person. All team members agree that our responsibilities to our fellow team members, to the project deliverables, UT Dallas staff, and our project mentor are our top priorities.

8.1.2 – Meetings:

All team members agree that if they cannot attend a scheduled meeting that they will contact the other team members and explain the circumstances at the earliest opportunity to minimize the effects of their absence on the project.

8.1.3 – Communications:

All team members agree that any project related email, voice mail, or text requiring a response will be answered in 24 hours or less.

8.1.4 – Deliverables:

All team members agree that if the timely submission of a deliverable to either the team, UT Dallas, or the project mentor is in jeopardy, they will notify all other team members a week prior to the deliverable due date or at the earliest possible opportunity so that other team members can act to resolve the issue before it adversely affects project success.

**8.2 – Remedies for Unacceptable Behavior**

8.2.1 – 1st Offense:

On the first occurrence of unacceptable behavior, determine the circumstances involved, resolve the problem, and document the event in the meeting minutes.

8.2.2 - 2nd Offense:

On a second occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.

8.2.3 – 3rd Offense:

On a third occurrence, again notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem. At this point, the team will have the \*option\* of removing the team member. If removed, then the team member receives a pro-rated grade based on the number of weeks they have participated in the group.

8.2.4 – Definition of unacceptable behavior:

Examples of unacceptable behavior may include not delivering on time, delivering poor quality work, missing team meetings, being unprepared for team meetings, disrespectful or rude behavior, etc. Reasons such as "too busy" or "I forgot", or "my dog ate my design model" are unacceptable.

Valid reasons that must be considered include those listed for obtaining an incomplete standing in a course (illness, death in the family, travel for business or academic reasons, etc.)

**References**

[1] Kendra Cooper. (2011, August 28), *Project Management Plan* (*1st edition*) [pdf file]. Available : http://www.utdallas.edu/~ewong/SE4485-2013-Fall/Template/1-managment-plan.pdf