Office Hours Development Team

23 March 2016

Software Engineering

Andrew McCracken, James Gruss, Naif Alotaibi, Tom Dylewski

**Change Table**

|  |  |  |
| --- | --- | --- |
| Change | Date | Team Member |
| Initial Draft | 3/23/16 | James Gruss |
| Sprint Dates starting at Sprint 2 end date | 4/12/16 | James Gruss |

**Proposed Software Development Life Cycle**

**Process Type**: Hybrid SCRUM

**Sprint Length**: 7 days (with planned exceptions)

**Sprint layout:** Tuesday to Tuesday

**Number of Sprints**: 4

**Rationale**

Our development team has a fairly short time period to complete this project. A process with a light-weight documentation style such as SCRUM will work well by building these activities into the development process. Furthermore, the full requirements for the project are not fully know at this time. Incremental development will allow for requirements change and further work to be completed by the requirements and project management team.

**General Process**

* Cyclical Definition
  + We will use the general model for the SCRUM process, but combine some tradition SCRUM activities into concurrent meetings and activities due to time constraints. We will keep a projected timeline of activities to estimate product delivery.
* General Cyclical Structure
  + Re/Prioritize Product Backlog → Sprint Planning → Sprint Execution → Sprint Wrap-up → Sprint Retrospective → <Repeat>

**Activities Defined**

* Prioritization: We will attempt to prioritize major features with their perceived sub-functionalities within the product backlog. Weighting will primarily be completed by estimations of project value and difficulty (time).
  + Reprioritization will include sharing of new information with members in person.
  + Deliverable: Updated product backlog on master backlog excel sheet (included) and snapshot of new backlog for comparison.
* Sprint Planning
  + Team Design Workshop: the team will analyze proposed features for the sprint and create models and diagrams to enhance coherent understanding of items and interfaces.
    - Deliverable: Pictures of hand-drawn models and diagrams marked with details of session date and item being described.
  + Sprint Backlog Prioritization:
    - Deliverable: Sprint backlog with more itemized sub-functionality detail than product backlog, but using same format in excel.
      * Includes item owners and metrics of priority and difficulty
* Sprint Execution: Here the team does the required work for the sprint.
  + Each team member is expected to work on his agreed upon tasks and ask for help from the team if necessary.
  + Sprint working meetings will be called by members on a need-be basis.
  + Members expected to use GitHub as a configuration management system, uploading his code and test cases by the end of the sprint.
  + Deliverable: Working code, summary of completed items, test cases for sub-functionality, and supplementary documentation for novel items.
* Sprint wrap-up
  + Compile summary of completed items into master backlog
  + Compile test cases
  + Run basic integration tests
  + Finalize master backlog as snapshot of current status
  + Readjust expected deliverable product via timeline review
* Sprint Retrospective
  + Assess effectiveness of SDLC process, especially the last-completed sprint

**Planned Timeline**

Sprint 1: 3/22 – 4/5

Sprint 2: 4/5 – 4/12

Sprint 3: 4/12 – 4/19

Sprint 4: 4/19 – 4/26

Sprint 5: 4/26 – 4/30

**Communication Structure**

* Tuesdays will include a 1:45 session that will be used for the Sprint wrap-up, sprint retrospective, and sprint planning,
  + Additional meetings can be scheduled, and this plan will be re-evaluated as time goes on.
* Thursdays include a 30 minute status update meeting
* Slack will be used for communication at a distance
  + Scheduling additional meetings
  + Asking questions or requesting help

**Sprint 1 Activities**

* Sprint 1 is over Easter break, and we need further documentation, so it is a longer sprint.
* Sprint 1 activities
  + Each member creates his own set of perceived project features and subtasks.
  + Each member estimates priority and difficulty of main features and subtasks
  + Each member learns unknown technologies
    - C#, ASP.NET, SQL, Git

**Quality Assurance**

* Team members will write test cases and complete testing for items that they own, submitting results at the sprint wrap-up.
* Heavily coupled functionalities will be tested during sprint by item owners
  + This will be planned at sprint planning
* Integration testing will take place at sprint wrap-up
* Excel sheets and automated testing tool reports are preferred deliverables

**Configuration Management**

* Git will be used as configuration management system.
* Branching Structure
  + Each sprint will branch from the master and be merged with the master at close
  + Each team member will create his own branch from the sprint branch, pushing and pulling to the sprint branch as needed to test interfaces with related features
* GitHub will be used on a day-to-day basis, and milestones will be submitted to GitLab