Developer Guide

Rebecca DeMarco

August 26, 2016

**International Space Station Data Visualization Project Developer Guide**

### Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 8/25/2016 | draft | Establishing draft | Rebecca DeMarco |
| 12/8/2016 | v1.0 | submission of final deliverable | Rebecca DeMarco |

### Introduction

#### 1.1 Purpose of Document

This document is to help future developers pick up the work on this project. It is meant as a resource to orientate them to the project. Here the new developer(s) can find a written status of where the project was last left off, the tools that are being used, and what documentation is left behind.

#### 1.2 Intended Auidience

The intended audience of this document is people who are assigned to work on the project.

#### 1.3 Document Conventions

Words that are italicized in section 2.1 Crash Course on Scrum mean that they will be defined in the definition part of that section and is a term you should be familiar with if you choose to fully implement scrum.

### 2.0 Development Process

The development process established for the Fall 2016 iteration of the web application was Scrum. Scrum is an implementation of Agile development.Due to difficulty in implementation, most of the process was implementing the artifacts (documentation) of Scrum. [Please read the Agile Manifesto](http://agilemanifesto.org/)

#### 2.1 Crash Course on Scrum

**Roles in Scrum**

* There are a few different roles in Scrum.
  + **Product Owner**: The product owner is usually the customer, the person that you are building the product for
  + **ScrumMaster**: The ScrumMaster is the person who leads the development team in following the process of Scrum
  + **Development team**: These are people who work on the project - i.e write code, build the hardware system
  + **Stakeholders**: Stakeholders include end users and anybody who is involved in the project

**Role Responsibilities**

* Each role in Scrum has things that they should do...
  + **Product Owner** - the product owner helps the team come up with the *product backlog* and should complete *backlog grooming* at the end of every *sprint*. Additionally, at the end of every sprint, the product owner should be updated on the project. The product owner usually has an opportunity to give feedback at the end of sprint, and this can be documented as part of the *sprint retrospective*.
  + **ScrumMaster** - The ScrumMaster is the person who makes sure that the development team follows the processes of Scrum. The ScrumMaster is also thought as a servant-leader, as they often manage the project and work on the project. The ScrumMaster is in charge of the *scrumboard* and helping to remove any *roadblocks* for development team members so that they can get their tasks done.
  + **Development team** - people on the development team are responsible for completing *user stories*. These are the people with technical skills who are doing things like writing code or creating a hardware system design.
  + **Stakeholders** - Stakeholders can influence product feedback and what the product owner wants the product to do.

**Starting the Process of Scrum**

1. Establish the need for the project
2. Work with the product owner to have *User Story* Elicitation
   * During *User Story* Elicitation, you work with the product owner to come up with an initial *product backlog*. After getting the initial \* product backlog\* set up, review the *user stories* and refine them into more well written user stories.
3. *Backlog grooming*
   * work with your product owner to *prioritize the backlog*
4. *Sprint planning*
   * decide what *user stories* you and your team are to accomplish during your *sprint*
5. Start the *sprint*
6. Work on *user stories*
7. Close/end *sprint*
8. *Demo*
   * at the end of sprint, show your current product iteration to your product owner if applicable
9. *Retrospective*
   * What went well
   * What can be improved
   * Product owner feedback
10. Repeat steps 3-9

**The life of a User Story**

* Things that should happen in the lifetime of a user story:
  + The user story should be written in the format of "As a , I want to so that ."
  + Each user story should have acceptance criteria to accompany it
  + Each user story should be given a story estimate (*story points*)
  + User stories should be assigned to a member of the development team
  + User stories should have work logged to them when they are being worked on. It is recommended that you log work as you go.

When is a user story considered done?

* A user story is considered done when the following things occur:
  + The acceptance criteria have been met
  + Any code that has been written has been reviewed and merged into master
  + Any code that has been written has been tested and documented in the test document
  + Work has been logged

##### **Words to Know - Definitions**

**Backlog Grooming**: Also known as backlog prioritization, prioritizing the backlog, or grooming the backlog. Backlog grooming is when the product owner looks over the backlog and indicates to the team which user stories are the most important ones. In some cases, this means that the product owner actually looks at each and every user story and ranks them. In other cases it means that the product owner tells the ScrumMaster (and sometimes team) what features are the most important to get down first and the ScrumMaster grooms the backlog accordingly.

**Burndown Chart**: A burndown chart is a way of graphically representing the ideal story points completed over time versus when the story points were actually completed over time.

**Daily Scrum**: Also known as daily standups or a standup meeting. During daily scrum, members of the development team share three things: work that they've completed, work that they're working on, and roadblocks that are stopping them from doing work.

**Demo**: Also known as end of sprint demo or product demonstration. A demo is when the current iteration of you product is presented to your product owners (and sometimes stakeholders). Demos usually happen at the end of the sprint, but sometimes they don't happen because no significant visual progress has been made or because the product owner is unavailable. Normally demos happen at the end of the sprint and the team showcases all the work that they have accomplished during the sprint.

**Product backlog**: The product backlog is a backlog of all of the user stories that still need to be done and not currently in an active sprint backlog. All user stories start in the product backlog. Then, during sprint planning, some of the user stories are pulled from the product backlog and placed into the sprint backlog. If a user story was being worked on during a sprint, but then got roadblocked and can't be completed until later, at the end of that sprint it can be placed back into the product backlog. Once a user story is complete, it is taken out of the product backlog and is documented as complete.

**Retrospective**: A retrospective is something that happens at the end of the sprint or when the sprint ends. During a retrospective each member goes over the work that the did and didn't do during the sprint and the team discusses what went well and what could be improved upon. Depending on organizational practices, justification as to why a user story is being pushed to the next sprint might be documented. Additionally, the product owner might give feedback about the progress made.

**Roadblock**: A roadblock is an impediment stopping a member of the development team from completing a particular task.

**Scrumboard**: A scrumboard is a mechanism used to keep track of user stories during each sprint. Depending on organizational practices, you may be using either a physical scrumboard or a virtual scrumboard. Scrumboard typically contain three categories: to-do, working on, and done. Regardless of the type of scrumboard, the card should indicate who is assigned to that user story and the story estimates.

**Sprint backlog**: The sprint backlog contains the user stories that the team as decided to accomplish during the sprint. After a sprint has started, do not add more user stories to the sprint backlog.

**Sprints**: Sprints are periods of time that are dedicated to accomplishing user stories. The purpose of a sprint is to build a functional, potentially shippable part of the product by the end of the sprint. After a sprint has started, do not add more user stories to the sprint backlog.

**Sprint Planning**: Sprint planning is when the ScrumMaster works with the development team to decide which user stories can be accomplished during that sprint. Usually, user stories are taken from the top of the product backlog, assigned to the appropriate team members, and put into the sprint backlog. After a sprint has started, do not add more user stories to the sprint backlog.

**Story points**: Also known as story estimates. Story points are the estimated number of hours of work needed to be done in order for the user story to be completed.

**User Stories**: User stories are a format of stating what needs to be done to build the product and what the product needs to do. User stories are written in the format of "As a , I want to so that ." User stories are accompanied by acceptance criteria. Acceptance criteria state what conditions need to be met in order for the user story to be considered done.

#### 2.2 Additional Reading on Scrum

* Supplemental Information on Scrum:
  + [Information about Scrum from Mountain Goat Software](https://www.mountaingoatsoftware.com/agile/scrum)
  + [Video: Learn About Scrum from Scrum Alliance](https://www.scrumalliance.org/why-scrum)
  + [Video: The Wrong way to do Agile: Specifications by Atlassian](https://www.youtube.com/watch?v=l1yWusiaLCM)
  + [Video: The Wrong way to do Agile: Team Structure by Atlassian](https://www.youtube.com/watch?v=HsB0UZR7XvE)

#### 2.3 How Scrum Was Used for this Project

**Scrum Implementation During Fall 2016** Initially, I tried to fully implement Scrum. Without a virtual scrumboard it was difficult to log work and keep track of user stories. Additionally, the lack of a virtual toolset made it harder to produce burndown charts. It was also difficult to implement because too many last minute meetings, tours and lectures, and time spent learning things I didn't know made it hard to correctly estimate what could be done during a sprint (I tried doing 1 week sprints). Since the project is in the early stages of development, much time was spent on setting up project documentation, processes, and UI design. Due to difficulty in implementation, most of the process was implementing the artifacts (documentation) of Scrum. For future implementation, I would recommend just working on the user stories and doing test driven design of the acceptance criteria.

### 3.0 Development Tools

* R 3.3.0 or greater
* Python 3.X via Anaconda
* Neo4j
* MongoDB

### 4.0 Project Documentation

* Needs Statement
  + Rebecca DeMarco (Fall 2016 Intern) worked with David to determine the needs of the project
* User Story Specification Document
  + This document is your go to document to see what the product is required to do. From the needs statement, Rebecca derived user stories and acceptance criteria.
* Design Specification Document
  + In this document you will find explanations of the design of the web application. You will find information on the software architecture, technical environment, software design, and user interface design. This document greatly captures the human factors efforts into the design of the user interface.
* Test Document
  + This document shows all of the verification testing that occurred to verify that the user stories have been satisfied.
* Developer Guide
  + This document helps the new project member understand the current status of the project and what tools and processes are used for the project.

### 5.0 Project Status

**Project Status as of December 9, 2016**

### 6.0 Project Resources

Here is a list of websites and videos that can help learn about data science and data visualization:

* [webdesignerdepot.com: 50 Great Examples of Data Visualization](http://www.webdesignerdepot.com/2009/06/50-great-examples-of-data-visualization/)
  + lists a number of different data vis tools that exist... you can do a number of visualizations shown with different packages in R
* [Wikipedia page for Data Visualization](https://en.wikipedia.org/wiki/Data_visualization)
  + good overview of the basic concepts of what data visualization entails
* [Good Charts by Scott Berinato](https://www.amazon.com/Good-Charts-Smarter-Persuasive-Visualizations/dp/1633690709/ref=sr_1_1?s=books&ie=UTF8&qid=1475613237&sr=1-1&keywords=good+charts+scott+berinato)
  + Book recommended during a Neo4j webinar on usability
* [Data Visualization Best Practices](http://www.slideshare.net/idigdata/data-visualization-best-practices-2013)
  + informative powerpoint on slideshare by a business intelligence and analytics professional
* [Data Visualization 101: How to Design Charts and Graphs](http://www.slideshare.net/Visage/data-visualization-101-how-to-design-chartsandgraphs)
  + great powerpoint on slideshare that shows what effective charts and graphs should look like
* [A gentle introduction to topic modeling using R](https://eight2late.wordpress.com/2015/09/29/a-gentle-introduction-to-topic-modeling-using-r/)