

SE 3XA3: Software Requirements Specification ScrumBot

Team 304, ScrumBot
Arkin Modi, modial
Leon So, sol4
Timothy Choy, choyt2

Last Updated: February 3, 2020

Contents

1	Project Drivers	1
1.1	The Purpose of the Project	1
1.2	The Stakeholders	1
1.2.1	The Client	1
1.2.2	The Customer	2
1.2.3	The Users (Roles in Scrum process)	2
1.2.4	Other stakeholders	3
1.3	Mandated Constraints	3
1.3.1	Solution Constraints	3
1.3.2	Implementation Environment of the Current System	3
1.3.3	Partner or Collaborative Applications	3
1.3.4	Off-the-Shelf Software	3
1.3.5	Anticipated Workplace Environment	3
1.3.6	Schedule Constraints	4
1.3.7	Budget Constraints	4
1.3.8	Enterprise Constraints	4
1.4	Naming Conventions and Terminology	4
1.5	Relevant Facts and Assumptions	4
2	Functional Requirements	5
2.1	The Scope of the Work and the Product	5
2.1.1	The Context of the Work	5
2.1.2	Work Partitioning	5
2.1.3	Individual Product Use Cases	5
2.2	Functional Requirements	5
3	Non-functional Requirements	5
3.1	Look and Feel Requirements	5
3.2	Usability and Humanity Requirements	5
3.3	Performance Requirements	5
3.3.1	Speed and Latency Requirements	5
3.3.2	Precision or Accuracy Requirements	5
3.4	Operational and Environmental Requirements	5
3.4.1	Expected Environment	5
3.4.2	Requirements for Interfacing with Adjacent Systems	6
3.5	Maintainability and Support Requirements	6

3.6	Security Requirements	6
3.7	Cultural Requirements	6
3.8	Legal Requirements	6
3.9	Health and Safety Requirements	6
4	Project Issues	7
4.1	Open Issues	7
4.2	Off-the-Shelf Solutions	7
4.3	New Problems	7
4.4	Tasks	7
4.5	Migration to the New Product	7
4.6	Risks	7
4.7	Costs	7
4.8	User Documentation and Training	7
4.9	Waiting Room	7
4.10	Ideas for Solutions	7
5	Appendix	8
5.1	Symbolic Parameters	8

List of Tables

1	Revision History	ii
---	----------------------------	----

List of Figures

Table 1: Revision History

Date	Developer(s)	Change
January 23, 2020	Arkin Modi	Copy template
February 3, 2020	Leon So	Worked on Stakeholders section
February 4, 2020	Leon So	Worked on Purpose, Non-functional Requirements sections
February 8, 2020	Leon So	Formatted and added non-functional requirements

This document describes the requirements for Scrumbot. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (Robertson and Robertson, 2012). If you make further modifications to the template, you should explicitly state what modifications were made.

1 Project Drivers

1.1 The Purpose of the Project

Scrum is an agile process framework widely used in industry for managing and coordinating collaborative projects. Scrum being a process based on the agile development method, follows a highly iterative process and often has heavy customer involvement, therefore it can be often be complex. It is often difficult to manage communication and coordination within the project due to the complexity of agile development processes. With Discord being a popular communication tool used by many teams of software developers today, Scrumbot provides a solution that directly integrates the management of a scrum development cycle into the communication channels. Scrumbot will allow for better management and organization of retrospectives, stand-ups, and other scrum/agile processes used by software teams during routine communication.

As a result, Scrumbot, a well-integrated scrum management system, can reduce inefficiencies surrounding the management of projects using the scrum framework, as well as strengthen communication within the agile process. This will help software development firms reduce costs and achieve better workflow.

1.2 The Stakeholders

This solution will be used in an environment where the development team uses the scrum agile framework and uses Discord as their routine communication channel.

1.2.1 The Client

- Dr. Ashgar Bokhari (3XA3)

The client for this project is Dr. Ashgar Bokhari - professor of the 3XA3 course.

1.2.2 The Customer

The customer of the ScrumBot is any firm looking to improve efficiency in communication and organization during the Scrum process of an agile software development cycle using the Scrum framework. This customer will likely be a software development firm (or the development department of a firm).

- Software Development Firm

The development firm will benefit through centralizing project resources, therefore improving the organization and management of the project. The development firm will therefore be able to reduce any costs associated with inefficiencies during the development process. This will also improve communication between members of the development firm.

1.2.3 The Users (Roles in Scrum process)

- Product Owner(s)

The product owner(s) will benefit from the reduced development time through improved communication and efficient management of the project. The product owner(s) will also benefit as their ideas and requirements will be better organized, thus helping the development team more efficiently achieve the goals of the product owner(s).

- Scrum Master

The scrum master will be better able to coordinate scrum plans and division of tasks, as well as routine communication (i.e. retrospectives and stand-ups). With a scrum management system directly implemented into the communication channels, the scrum master will be able to directly manage the meeting information and their scrum development cycle during communication.

- Development Team

The development team will be able to more easily follow through on tasks assigned by the scrum master. The development team may easily refer to any relevant articles or information regarding their project. This will reduce inefficiencies and speedup communication within the development team. It will also reduce confusion, as all information concerning the development cycle will be available through their routine communication channel (Discord).

1.2.4 Other stakeholders

1.3 Mandated Constraints

1.3.1 Solution Constraints

1.3.2 Implementation Environment of the Current System

The application will be installed on the user's local machine or a server, and added to a Discord server. To function, the application be started on the machine and the user will interact with it through the respective Discord server.

1.3.3 Partner or Collaborative Applications

1.3.4 Off-the-Shelf Software

The following off-the-shelf software will utilized:

- Discord (Available at <https://discordapp.com/>)
- Discord Server (Available through Discord)
- Python 3 (Available at <https://www.python.org/downloads/>)
- discord.py (Available at <https://pypi.org/project/discord.py/>)

All off-the-shelf software are available for free.

1.3.5 Anticipated Workplace Environment

The anticipated workplace environment for this application is within a software development team's Discord server. This application can be used from anywhere, as long as the user has access to the respective Discord server.

1.3.6 Schedule Constraints

The project deliverable must be completed by their respective deadline. The remaining deadline include:

- Proof of Concept (February 13, 2020, 1:00 PM)
- Test Plan, Revision 0 (February 28, 2020, 11:30 PM)
- Design & Document, Revision 0 (March 13, 2020, 11:30 PM)
- Demonstration, Revision 0 (March 17, 2020)
- Demonstration, Revision 1 (March 31, 2020)
- Final Documentation, Revision 1 (April 6, 2020, 11:30 PM)

1.3.7 Budget Constraints

This project has no monetary budget. If there are any necessary purchases for development, the cost shall be paid by the project members. All resources to re-create and upgrade the existing project are provided.

1.3.8 Enterprise Constraints

This application will be available for free to any user that has access to a Discord server and a machine with Python 3 (and the dependencies) installed.

1.4 Naming Conventions and Terminology

1.5 Relevant Facts and Assumptions

User characteristics should go under assumptions.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

2.1.2 Work Partitioning

2.1.3 Individual Product Use Cases

2.2 Functional Requirements

3 Non-functional Requirements

3.1 Look and Feel Requirements

- The system will output the

3.2 Usability and Humanity Requirements

- The system shall have intuitive user commands

3.3 Performance Requirements

3.3.1 Speed and Latency Requirements

- The system shall respond to user commands within 2 seconds.

3.3.2 Precision or Accuracy Requirements

- The system shall have a meeting locations and schedules accuracy greater than 70%.

3.4 Operational and Environmental Requirements

3.4.1 Expected Environment

- The system shall be able to operated and be executed within the Discord application

3.4.2 Requirements for Interfacing with Adjacent Systems

- The system shall connect to Google API services

3.5 Maintainability and Support Requirements

- The code shall be documented using comments
- The code shall be documented using Doxygen
- The code documentation shall be easy to understand

3.6 Security Requirements

- The connection between the system and the APIs shall use HTTPS for security

3.7 Cultural Requirements

- The system shall use Canadian English spelling

3.8 Legal Requirements

3.9 Health and Safety Requirements

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

4 Project Issues

4.1 Open Issues

4.2 Off-the-Shelf Solutions

4.3 New Problems

4.4 Tasks

4.5 Migration to the New Product

4.6 Risks

4.7 Costs

4.8 User Documentation and Training

4.9 Waiting Room

4.10 Ideas for Solutions

References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.