

# SE 3XA3: Test Plan

## ScrumBot

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

Last Updated: February 27, 2020

# Contents

<b>1</b>	<b>General Information</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.2	Scope . . . . .	1
1.3	Acronyms, Abbreviations, and Symbols . . . . .	1
1.4	Overview of Document . . . . .	1
<b>2</b>	<b>Plan</b>	<b>1</b>
2.1	Software Description . . . . .	1
2.2	Test Team . . . . .	2
2.3	Automated Testing Approach . . . . .	2
2.4	Testing Tools . . . . .	2
2.5	Testing Schedule . . . . .	2
<b>3</b>	<b>System Test Description</b>	<b>2</b>
3.1	Tests for Functional Requirements . . . . .	2
3.1.1	Area of Testing1 . . . . .	2
3.1.2	Area of Testing2 . . . . .	3
3.2	Tests for Nonfunctional Requirements . . . . .	3
3.2.1	Area of Testing1 . . . . .	3
3.2.2	Area of Testing2 . . . . .	3
3.3	Traceability Between Test Cases and Requirements . . . . .	3
<b>4</b>	<b>Tests for Proof of Concept</b>	<b>3</b>
4.1	Area of Testing1 . . . . .	3
4.2	Area of Testing2 . . . . .	4
<b>5</b>	<b>Comparison to Existing Implementation</b>	<b>4</b>
<b>6</b>	<b>Unit Testing Plan</b>	<b>4</b>
6.1	Unit testing of internal functions . . . . .	4
6.2	Unit testing of output files . . . . .	4
<b>7</b>	<b>Appendix</b>	<b>5</b>
7.1	Symbolic Parameters . . . . .	5
7.2	Usability Survey Questions? . . . . .	5

# List of Tables

1	Revision History . . . . .	iii
2	Table of Abbreviations . . . . .	1
3	Table of Definitions . . . . .	1

## List of Figures

Table 1: Revision History

Date	Developer(s)	Change
January 23, 2020	Arkin Modi	Copy template
February 20, 2020	Arkin Modi	Created the Purpose Section
February 27, 2020	Timothy Choy, Leon So	Updated General Information section
February 27, 2020	Arkin Modi	Worked on Test Team, Automated Testing Approach, and Testing Tools

# 1 General Information

## 1.1 Purpose

The purpose of this document is to outline the testing, validation, and verification process of the functional and non-functional requirements, for the ScrumBot project. These test cases were conceived before the implementation and therefore will be used by the project members for future reference during the development process.

## 1.2 Scope

## 1.3 Acronyms, Abbreviations, and Symbols

Table 2: Table of Abbreviations

Abbreviation	Definition
POC	Proof of Concept
SRS	Software Requirements Specification

Table 3: Table of Definitions

Term	Definition
Pytest	A unit testing framework for Python
Term2	Definition2

## 1.4 Overview of Document

This document outlines a test plan that fully encompasses all requirements of ScrumBot, as stated in the SRS. This document includes relevant information concerning: test team, automated testing, testing tools, testing schedule, unit-testing, and test cases.

# 2 Plan

## 2.1 Software Description

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. 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 stages used by software teams within their routine communication channel.

## 2.2 Test Team

The test team will consist of all the members of the project: Arkin Modi, Leon So, and Timothy Choy.

## 2.3 Automated Testing Approach

Testing shall be automated with the use of the GitLab's CI/CD tool. The tests will be run every time a commit is pushed to the repository.

## 2.4 Testing Tools

The unit tests will be written using the Pytest framework. Static testing will be done with the use of Pylint.

## 2.5 Testing Schedule

See Gantt Chart at the following url, <https://gitlab.cas.mcmaster.ca/modia1/ScrumBot/-/blob/master/ProjectSchedule/>.

# 3 System Test Description

## 3.1 Tests for Functional Requirements

### 3.1.1 Area of Testing1

#### Title for Test

1. test-id1

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

### **3.1.2 Area of Testing2**

...

## **3.2 Tests for Nonfunctional Requirements**

### **3.2.1 Area of Testing1**

**Title for Test**

1. test-id1

Type:

Initial State:

Input/Condition:

Output/Result:

How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

### **3.2.2 Area of Testing2**

...

## **3.3 Traceability Between Test Cases and Requirements**

# **4 Tests for Proof of Concept**

### **4.1 Area of Testing1**

**Title for Test**

1. test-id1

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

## **4.2 Area of Testing2**

...

## **5 Comparison to Existing Implementation**

## **6 Unit Testing Plan**

### **6.1 Unit testing of internal functions**

### **6.2 Unit testing of output files**



## **7 Appendix**

This is where you can place additional information.

### **7.1 Symbolic Parameters**

The definition of the test cases will call for SYMBOLIC\_CONSTANTS. Their values are defined in this section for easy maintenance.

### **7.2 Usability Survey Questions?**

This is a section that would be appropriate for some teams.