Software Requirements Specification

for

CubeSat

Version 1.0b approved

Prepared by:
Isaac Cano,
Chung An Chen,
A. Darren Ferahyan,
Giovanni A. Garcia,
Tan Luong

Aerospace Corporation

December 04, 2018
# Table of Contents

Table of Contents 2
Revision History 3

1. Introduction 4
   1.1 Purpose 4
   1.2 Document Conventions 5
   1.3 Intended Audience and Reading Suggestions 5
   1.4 Product Scope 6
   1.5 References 6

2. Overall Description 7
   2.1 Product Perspective 7
   2.2 Product Functions 8
   2.3 User Classes and Characteristics 8
   2.4 Operating Environment 9
   2.5 Design and Implementation Constraints 9
   2.6 User Documentation 9
   2.7 Assumptions and Dependencies 10

3. External Interface Requirements 11
   3.1 User Interfaces 11
   3.2 Hardware Interfaces 12
   3.3 Software Interfaces 12
   3.4 Communications Interfaces 12

4. System Features 13
   4.1 CubeSat 13
   4.2 Ground System 13
   4.3 SimPlat 13

5. Other Nonfunctional Requirements 14
   5.1 Performance Requirements 14
   5.2 Safety Requirements 14
   5.3 Security Requirements 14
   5.4 Software Quality Attributes 15
   5.5 Business Rules 15

6. Other Requirements 16

Appendix A: Glossary 16

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.
# Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason For Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team CubeSat</td>
<td>9/18/2018</td>
<td>1st Draft</td>
<td>1.0a</td>
</tr>
<tr>
<td>Team CubeSat</td>
<td>10/27/2018</td>
<td>2nd Draft</td>
<td>1.0b</td>
</tr>
</tbody>
</table>

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.
1. **Introduction**

1.1 **Purpose**

1. Outline the Software Capabilities of CubeSat
   a. Receive commands from the Ground System (GS)
   b. Execute commands
   c. Process sensor data to autonomously determine actions
   d. Interpret sensor/image data
   e. Service another CubeSat

2. Detail the all components related to CubeSat:
   a. *Ground System (GS)*:
      *Shall* -
      i. Receive data from the CubeSat
      ii. Send commands to the CubeSat
      iii. Display a User Interface (UI) to manage/visualize data

   b. *Space Simulator (SimPlat)*:
      *Shall* -
      i. Receive move commands from the CubeSat
      ii. In case of an emergency, receive a stop cmd from GS

3. Provide additional information for Other Non-Functional Requirements:
   a. *Hardware*:
      i. Processing Device:
         1. Raspberry Pi 3 Model B+
      ii. Sensor Types:
         1. Proximity
         2. Camera
      iii. Communication Types:
         1. WiFi
         2. LoRaWAN/Zigbee
   b. Performance
   c. Security
1.2 Document Conventions

<table>
<thead>
<tr>
<th>Typographical Conventions:</th>
<th>Details:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolded Font</td>
<td>Introduces a new section</td>
</tr>
<tr>
<td>Italicized Font</td>
<td>Important components inside a hierarchy</td>
</tr>
<tr>
<td>Underline Word</td>
<td>Emphasizes importance</td>
</tr>
<tr>
<td>Bullet Points</td>
<td>Explain the details to a given hierarchy</td>
</tr>
</tbody>
</table>

1.3 Intended Audience and Reading Suggestions

<table>
<thead>
<tr>
<th>Intended Audience:</th>
<th>Refer to: (Section #)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Managers (PM)</td>
<td>1.X - 6.X</td>
</tr>
<tr>
<td>Developers (Devs)</td>
<td>2.X - 4.X</td>
</tr>
<tr>
<td>Testers</td>
<td>2.X - 4.X</td>
</tr>
<tr>
<td>Users</td>
<td>2.6</td>
</tr>
</tbody>
</table>

*This document was meant to be read in sequential order*
1.4 Product Scope

1. The Software Products Produced:

**GS** - will be used to control the CubeSat in Low Earth Orbit (LEO) to de-orbit decommissioned satellites (Target) and space debrises

a. GS’s Functionality:
   i. Interpret proximity sensor data to approach/avoid objects, process camera image data and understand what it’s looking at, send/receive thruster status, control deorbit mechanism, communicate status and receive commands from GS.
   ii. Provide one-way communication between GS and CubeSat
   iii. Compare action/response data of CubeSat
   iv. Provide telemetry and control of CubeSat. Ability to compare data from both to evaluate accuracy of motion and synchronicity.

**SimPlat** - will use CubeSat’s commands to simulate motion in space.

b. SimPlat’s Functionality:
   i. Provide one-way communication from CubeSat
   ii. Read real-time signals from CubeSat:
       1. Interpret intended motion
       2. Move to simulate motion in space

1.5 References

Aerospace Corporate University Affiliate Program - CubeSat Test Platform Project Description
2. Overall Description

2.1 Product Perspective
2.2 Product Functions

<table>
<thead>
<tr>
<th>Function Name:</th>
<th>Sub-Components:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>Display</td>
<td>Displays CubeSat’s Controls and Statuses</td>
</tr>
<tr>
<td></td>
<td>Movement Prediction</td>
<td>Interprets Sensor Data to predict CubeSat’s next action</td>
</tr>
<tr>
<td></td>
<td>Image Recognition</td>
<td>Interpret live camera feed to detect objects</td>
</tr>
<tr>
<td></td>
<td>Data Transfer</td>
<td>Send commands to the CubeSat</td>
</tr>
</tbody>
</table>

2.3 User Classes and Characteristics

Basic knowledge of using computers is necessary to use this web application. Knowledge of how to use a mouse or keyboard and internet browser is necessary to navigate the user interface.

As of right now only the administrators interact with the web portal. They manage the overall system.

*In the future, functionality of this system will be divided among multiple users. Admin will spread the responsibility of controlling the CubeSat to different user groups. Each user group will have specific components that they are responsible for. User groups will be added as needed.*

<table>
<thead>
<tr>
<th>User Type:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin</td>
<td>Controls entire system</td>
</tr>
<tr>
<td>Co-Admin</td>
<td>Maintains system controls</td>
</tr>
</tbody>
</table>
2.4 **Operating Environment**

The software will be web-based, meaning it will be accessible by browser.

2.5 **Design and Implementation Constraints**

<table>
<thead>
<tr>
<th>Constraint:</th>
<th>Component Type:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimPlat</td>
<td>Hardware</td>
<td>Attempts to imitate movement in space.</td>
</tr>
<tr>
<td>Camera</td>
<td>Hardware</td>
<td>1080p quality may be too high resolution. Down-scaling may need to be applied.</td>
</tr>
<tr>
<td>Image Recog.</td>
<td>Software</td>
<td>Lack of data due to the reduced frame rate or pixel density.</td>
</tr>
<tr>
<td>Data Transfer</td>
<td>Hardware</td>
<td>Data transfer speeds may be too low to stream 1080p video. May need to reduce fps.</td>
</tr>
<tr>
<td>Time Delay</td>
<td>Hardware</td>
<td>There will be a delay in time. Actions may be impacted.</td>
</tr>
<tr>
<td>Thrusters</td>
<td>Hardware</td>
<td>Thrusters will be simulated by lights. No actual propulsion system will be used.</td>
</tr>
</tbody>
</table>

2.6 **User Documentation**


*Not a working link*
2.7 Assumptions and Dependencies

<table>
<thead>
<tr>
<th>Assumptions/Dependencies</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Connectivity</td>
<td>WiFi, Ethernet, Tethering, etc.</td>
</tr>
<tr>
<td>Modern Internet Browser</td>
<td>Google Chrome, Firefox, Edge</td>
</tr>
<tr>
<td>Computer &amp; Computer Peripherals</td>
<td>Computer, Mouse, Keyboard, Monitor</td>
</tr>
</tbody>
</table>
3. External Interface Requirements

3.1 User Interface

This software only requires an up-to-date web application on a device connected to the internet.
The user interface will consist of clickable/tappable buttons used to navigate. It is intended to be usable on any device with web connectivity and internet browser, so everything should be navigable through a tap or click.
There will be four different screen to show different data receiving from CubeSat and camera.
There will be a message display location anytime when trying to move CubeSat.
3.2 **Hardware Interfaces**

<table>
<thead>
<tr>
<th>Component:</th>
<th>Software Interaction:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Pi 3 Model 3B+ (1)</td>
<td>Runs the motion software for the SimPlat.</td>
</tr>
<tr>
<td>Raspberry Pi 3 Model 3B+ (2)</td>
<td>Runs the Machine Learning Algorithm for the CubeSat.</td>
</tr>
<tr>
<td>Camera</td>
<td>Attaches to (2) and records video for ML Algorithm.</td>
</tr>
<tr>
<td>WiFi Module</td>
<td>Built into (2) and sends image data to the GS.</td>
</tr>
<tr>
<td>Sensors (Proximity, Motion)</td>
<td>Attaches to (2) and provide data for the Machine Learning Algorithm.</td>
</tr>
<tr>
<td>LoraWAN/Zigbee</td>
<td>Attaches to (2) and sends sensor data to the GS.</td>
</tr>
</tbody>
</table>

3.3 **Software Interfaces**

Software shall store information to database such as location data and video data.
Software shall communicate with CubeSat to control it.
Software shall communicate with camera for receiving video data.
Software shall make connection between itself and CubeSat with wifi.
Software shall allow user to control CubeSat in any secured environment.
Software shall provide detail about CubeSat when CubeSat is working.

3.4 **Communications Interfaces**

The GS system shall use wifi for communication over the internet/raspberry pi and for the internet communication will be through the connection signal.
4. System Features

4.1 CubeSat

4.1.1 Shall receive commands from GS.
4.1.2 Shall send telemetry data to GS.
4.1.3 Shall send commands to SimPlat.
4.1.4 Shall autonomously navigate to target specified by GS.
  4.1.4.1 Shall avoid obstacles.
  4.1.4.2 Shall identify specified target.
4.1.5 Shall attach to the specified target.
4.1.6 Should service specified target.
4.1.7 Should deorbit specified target.

4.2 Ground System - GS

4.2.1 Shall send commands to the CubeSat.
  4.2.1.1 Shall specify a target.
  4.2.1.2 Shall send retrieval commands.
4.2.2 Shall send SimPlat emergency STOP command.
4.2.3 Shall receive telemetry data from the CubeSat.
4.2.4 Shall store telemetry data in a database.
4.2.5 Shall have an intuitive User Interface.
  4.2.5.1 Shall display sensor data.
  4.2.5.2 Shall display image data.
  4.2.5.3 Shall display telemetry data.
  4.2.5.4 Shall display CubeSat Diagnostics.

4.3 Space Dynamics Simulation Platform (SimPlat)

4.3.1 Shall simulate motion in space.
4.3.2 Shall receive commands from CubeSat.
4.3.3 Shall receive emergency STOP command from GS.
5. **Other Nonfunctional Requirements**

5.1 **Performance Requirements**

- The transferring of data between the CubeSat and GS must be no less than 8 MBps.
- This allows the the CubeSat and GS to communicate and transfer video from the camera in the CubeSat to the GS.
- This will also allow both to exchange information and communicate with lower delay.

5.2 **Safety Requirements**

Malfunctioning of this kind of software, could not possibly cause direct or indirect, human or environmental damage. Therefore, there are no safety requirements for this project. (ADF)

5.3 **Security Requirements**

- The server must authenticate every request accessing the restricted Web pages.
- After authenticating the browser, the server must determine whether that browser is authorized to access the requested restricted Web pages.
- The system must have security controls to protect against attacks.
- The behavior of the software must be correct and predictable.
- The software must ensure the integrity of the user information. (ADF)

5.4 **Software Quality Attributes**

- Correctness:
  - Agreement of program code with specifications
  - Independence of the actual application of the software system
- Reliability: The probability that this system fulfills a function for a number of input trials under specified input conditions in a time interval (assuming that hardware and input are free of errors).
- Learnability: The user interface should present information as close to reality as possible and permit efficient utilization of the software’s failures.
• Efficiency: Ability of a software system to fulfill its purpose with the best possible utilization of all necessary resources (time, storage, transmission, channels, and peripherals). (ADF)

5.5 Business Rules

The system shall have roles assigned for every individual. Such roles shall be: administrators, moderators, and users. Each role shall have different access level to the system.

1. Administrators shall have the ability to:
   a. view the GS,
   b. send commands to the CubeSat from GS,
   c. access system data,
   d. modify system data,
   e. view system log,
   f. change the roles of other users,
   g. add new users

2. Moderators shall have the ability to:
   a. view the GS,
   b. access system data,
   c. view system log,
   d. add new users

3. Users shall have the ability to:
   a. view the GS
6. Other Requirements

Appendix A: Glossary

<table>
<thead>
<tr>
<th>Abbreviation/Acronym</th>
<th>Phrase</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CubeSat</td>
<td>CubeSat</td>
<td>A type of miniaturized satellite for space research that is made up of multiples of 10×10×10 cm cubic units.</td>
</tr>
<tr>
<td>GS</td>
<td>Ground System</td>
<td>A centralized venue for managing some operation.</td>
</tr>
<tr>
<td>LEO</td>
<td>Lower Earth Orbit</td>
<td>An orbit around Earth with an altitude above Earth's surface of 2,000 kilometers (1,200 mi), and an orbital period between about 84 and 127 minutes.</td>
</tr>
<tr>
<td>SimPlat</td>
<td>Space Dynamics Simulation Platform</td>
<td>A vehicle with platform and CubeSat mount that simulates motion in space.</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirements Specification</td>
<td>A description of a software system to be developed.</td>
</tr>
<tr>
<td>Target</td>
<td>Decommissioned/Wounded Satellite</td>
<td>Satellites that are no longer functional or operable.</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
<td>The means by which the user and the cubesat interact, in particular the use of input device and software.</td>
</tr>
</tbody>
</table>