

SOFTWARE REQUIREMENTS SPECIFICATION

SOFTWARE ENGINEERING AND PROJECT

UNIVERSITY OF ADELAIDE

Prospector Sea Floor Mapping System

Navdeep Singh (a1660360)

Liang Yuan (a1679380)

Zeqi Fu (a1680895)

Tao Zhang (a1680974)

Lili Wu (a1683229)

Yi Lin (a1682781)

Yann Frizenschaf
(a1162562)

Semester 2, 2016

| Revision History | | | | |
|------------------|------------------|------------|---|---------|
| Date | Name | Student ID | Updates | Version |
| 21st Aug | Yann Frizenschaf | 1162562 | Added 3 operate models for the movement system in Description Section | 0.1 |
| 21st Aug | Yi Lin | 1682781 | Added GUI statement in Description Section | 0.2 |
| 21st Aug | Yann Frizenschaf | 1162562 | Added 2.1 Features Part and 2.1.1 Primary Systems Part in Description Section; Added FRC 015, FRC 016, FRE 004, FRE 005 in 4.3 Vision Section | 0.3 |
| 25th Aug | Yann Frizenschaf | 1162562 | Draft Release | 1.0 |
| 20th Oct | Liang Yuan | 1679380 | update 3.2.1,4.2.2,5.1.1 combined FRC0015,FRC0016 | 1.1 |
| 24th Oct | Yi Lin | 1682781 | added content about safety requirements | 1.2 |
| 25th Oct | Yi Lin | 1682781 | update 1.2, 1.4 and section 2 | 1.3 |
| 26th Oct | Yi Lin | 1682781 | update section 3 and 5 | 1.4 |
| 27th Oct | Yi Lin | 1682781 | update section 4 | 1.5 |
| 27th Oct | Tao Zhang | 1680974 | Separate section of Assumption and Constraint | 1.6 |
| 27th Oct | Yi Lin | 1682781 | Final Release | 2.0 |

Contents

| | | |
|----------|------------------------------------|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Conventions | 1 |
| 1.3 | Intended Audience | 2 |
| 1.4 | Scope | 2 |
| 2 | System Description | 3 |
| 2.1 | Features | 3 |
| 2.1.1 | Component Systems: | 3 |
| 2.2 | Users | 5 |
| 2.3 | Environment | 5 |
| 2.4 | Assumptions | 5 |
| 2.5 | Constraints | 6 |
| 3 | User Requirements | 6 |
| 3.1 | Operations GUI | 6 |
| 3.1.1 | Core Requirements | 6 |
| 3.1.2 | Extension Requirements | 9 |
| 3.2 | Mapping Output | 10 |
| 3.2.1 | Core Requirements | 10 |
| 3.2.2 | Extension Requirements | 11 |
| 4 | Functional Requirements | 11 |
| 4.1 | Movement | 11 |
| 4.1.1 | Core Requirements | 11 |
| 4.1.2 | Extension Requirements | 13 |
| 4.2 | Mapping | 14 |
| 4.2.1 | Core Requirements | 14 |
| 4.2.2 | Extension Requirements | 16 |
| 4.3 | Sensors | 16 |
| 4.3.1 | Core Requirements | 16 |
| 4.3.2 | Extension Requirements | 17 |
| 5 | Non-functional Requirements | 17 |
| 5.1 | Environment | 17 |
| 5.1.1 | Core Requirements | 17 |
| 5.1.2 | Extension Requirements | 18 |
| 5.2 | Safety | 18 |
| 5.2.1 | Core Requirements | 18 |

| | | |
|----------|----------------------------------|-----------|
| 5.2.2 | Extension Requirements | 18 |
| 5.3 | Quality assurance | 19 |
| 5.3.1 | Core Requirements | 19 |
| 5.3.2 | Extension Requirements | 19 |
| 6 | Glossary | 19 |
| 7 | References | 20 |

1 Introduction

1.1 Motivation

The sea floor is considered a significant frontier for development of many industries including, but not limited to, scientific research, mining, transport, and maintenance of communications infrastructure. A key consideration for any activity taking place on or near the sea floor is a sound understanding of the geology and topography of the area under consideration.

This document comprises the Software Requirements Specification for the software component of the Sea Floor Mapping System(SFM) to be developed by the Prospector team for SeaFaults. The requirements for the software system are described in detail in the following sections.

1.2 Conventions

Requirements of the system are organised into three categories:

User requirements (UR): define the operation of software components which are user-facing, and require some interaction by the user.

Functional requirements (FR): define the externally quantifiable functions the system must include, measured as a set of inputs, observable behavior, and outputs of the software.

Non-functional requirements (NR): define all internal requirements placed upon the system, which may not have an observable functional effect, but are important for reasons of maintainability, extensibility, and/or resource and environment constraints. Generally, the functional requirements are related to system design, and the non-functional requirements are related to system architecture and operating environment.

Each of the above requirement types are further sorted into two classes: core requirements, which must be met in order for the system to meet its contractually obligated purpose, and extension requirements, which are not strictly required but are desirable to have in order to improve the system's efficiency, usability or accuracy above the strictly required level.

For each of the above requirement types, each individual requirement is designated by a prefix (UR, FR or NR) corresponding to its requirement type, a requirement

class (C for core, E for extension), and a three digit number identifier. For example, the first core user requirement is designated URC001 and the second functional extension requirement is designated FRE002. Finally, each requirement is designated with a priority (High, Medium, or Low) indicating the requirement's relative importance to the system's core mission.

A glossary of all important terms and abbreviations can be found in Section 6.

1.3 Intended Audience

This document is intended for consumption by all key stakeholders within the SeaFaults organisation and their clients, as well as the Prospector software development team as a high-level reference for the SFM system's required software functionality.

For SeaFaults stakeholders, this document provides an overview of the functionality which may reasonably be expected from the software system developed by Prospector, as well as defining the scope and range of functionality. Such stakeholders should read the entire document, with an emphasis on the scope, user requirements and functional requirements.

For Prospector software developers, this document describes a high-level overview of the requirements of the software to be developed in non-technical language. While this gives room for interpretation in terms of technical implementation of the requirements, developers should consistently refer to this document to ensure their work remains focussed on the core requirements and within the project's overall scope definition.

1.4 Scope

This document defines the requirements for the software component of the SFM system only. While hardware considerations are touched on in the form of non-functional requirements, the hardware to be used has been defined and consideration of this (beyond its effect on the requirements of the software system) are beyond the scope of both this document and the project as a whole.

The software for which the requirements are defined herein is intended to enable effective mapping of a defined area of sea floor in order to determine its suitability for use by a SeaFaults client. It will do this by mapping the sea floor as accurately and efficiently as possible, and providing outputs pertaining to the mapping operation once the operation is complete. The system itself makes no determination of the

suitability of an area for a particular application, rather, it produces quantitative map data which can be utilised by SeaFaults and/or its clients to assess the mapped area for suitability. All known environmental and external constraints on the system and its operations are contained within this document, largely in the form of functional and non-functional requirements.

2 System Description

2.1 Features

The basic operation of the Prospector SFM system is to be as follows[1] (these requirements are formalised in Sections 3, 4 and 5): the SeaFaults Mapping SeaFaults Mapping Robot will be placed at a predefined point in the survey area, and will subsequently traverse the entire survey area (where possible), synthesising sensor data with an estimate of its current location in order to build a map of the survey area in real-time. Features to be mapped include faultlines (two-dimensional, coloured), obstacles (three-dimensional) and survey area boundaries.

During the mapping operation, the SeaFaults Mapping SeaFaults Mapping Robot must move autonomously and avoid obstacles and No-Gone-Zones(NGZ) under the observation of an operator, who may intervene in an emergency. When the mapping operation is complete, the SeaFaults Mapping SeaFaults Mapping Robot must reach the predefined extraction point.

After extraction, the map data model must be extracted from the SeaTank's on-board data and saved to an XML file. A partially completed map in the form XML file can also be loaded and subsequently completed by the SeaTank.

The main features of the software component of the Prospector SFM system are detailed below. Note that Section 2.1.1 contains descriptive information only as a reference, which do not constitute part of the formal requirements.

2.1.1 Component Systems:

Movement/Geolocation: The software must be capable of controlling the hardware in such a way that the SeaFaults Mapping SeaFaults Mapping Robot is able to move to a nominated location in a reasonable time, incorporating sensor feedback via the sensor system in order to make informed decisions about direction changes due to obstacles or site boundaries. It must also be capable

of synthesising multiple inputs including motor readings and inputs from the sensor system in order to maintain an estimate of its current position for both navigation output to the mapping system. The movement system must be able to operate in three distinct modes:

- Fully manual mode, in which the user explicitly controls the movement of the SeaFaults Mapping SeaFaults Mapping Robot (forward, reverse and rotation).
- Move-to-point mode, in which the user specifies a point in survey-area coordinates (see Section 2.4) and the SeaFaults Mapping SeaFaults Mapping Robot moves to the specified point, avoiding obstacles as required.
- Fully autonomous mode, in which the user instructs the SeaFaults Mapping SeaFaults Mapping Robot to begin the mapping operation, after which the SeaFaults Mapping SeaFaults Mapping Robot will traverse and map the survey area as completely as possible in a systematic fashion.

The detailed requirements of each of these modes are listed in Section 4.

Sensors: The software must be capable interpreting inputs from hardware sensors in order to identify obstacles, site boundaries and fault lines, and feed this information to both the Movement System and the Mapping System.

Mapping: The software must be capable of combining inputs from both the Movement System (current location and orientation) and Sensor System (currently visible topographical features) in order to build a map of the survey area in real time. It must also be capable of resuming operation appropriately when given a partial or incomplete map of the current survey area, and exporting map data in an appropriate format after the mapping operation is completed.

GUI: A user-friendly GUI must be provided to enable control and oversight of mapping operations. The GUI must include controls and visual feedback for the operator as defined in Section 3.

Communications: The software must allow for communication between the operations centre on the surface and the SeaFaults Mapping SeaFaults Mapping Robot on the sea floor, in order to issue commands and monitor the state of the system. The communication functionality is provided by the LeJOS platform.

Emergency: At any time, the operator must be able to bring the SeaFaults Mapping SeaFaults Mapping Robot to a complete and immediate stop if the current state of the system is deemed to be hazardous.

2.2 Users

The intended users of this system fall into two categories: system operators and map consumers. The manner in which these two user groups interact with the software system and its interfaces differs in several key aspects.

System operators oversee the real-time operations of the SeaFaults Mapping SeaFaults Mapping Robot while the sea floor mapping is being conducted. These users interact with the Human-Machine Interface (HMI) during the operation in order to define the target location for the SeaTank, monitor the state of the system for features of interest or hazards, and determine when the mapping operation is complete and the SeaFaults Mapping SeaFaults Mapping Robot should be returned to the surface. The needs of these users should be considered when designing and implementing the system's primary HMI including real-time control and monitoring features.

Map consumers have no interaction with the system during operation, but rather are concerned with the quality, accuracy, completeness and file format of the system outputs in the form of sea floor maps. The needs of these users should be considered when designing and implementing the conversion and delivery of map data in the required XML format.

2.3 Environment

The Prospector SFM software must be implemented for the provided hardware and firmware platform, that is, the Lego Mindstorms EV3 and LeJOS. The user-interface application must be capable of running on Windows and Mac operating systems (see Section 5).

2.4 Assumptions

- Operational area will not exceed A1 paper size.
- Boundaries and faultlines will have well-defined colours, and these will not change across operations.
- The software is not expected to function correctly in the event of damaged hardware; protection of the system or recovery from external damage is a hardware responsibility and is therefore beyond the scope of this document and project.

- The operator can maintain remote visual contact with the SeaFaults Mapping SeaFaults Mapping Robot at all times.
- The SeaFaults Mapping SeaFaults Mapping Robot can enter the extraction zone from any direction.

2.5 Constraints

- The coordinates of the SeaFaults Mapping SeaFaults Mapping Robot at the start of every mapping operation will be known in order to provide a baseline for future position estimates.
- The coordinates of the SeaFaults Mapping SeaFaults Mapping Robot and all relevant mapped features will be specified in survey-area coordinates, measured from the South-West corner of the survey area and measured in centimetres.

3 User Requirements

3.1 Operations GUI

3.1.1 Core Requirements

URC001: GUI Mode Change

Description: The software must include a GUI (hereafter referred to as the Operations GUI) which enables the user to change the SeaFaults Mapping SeaFaults Mapping Robot operation mode (see Section 2.1.1) and perform all actions required for each operation mode.

Rationale: Encapsulates control and monitoring interface for operations users.

Priority: High.

Status: Implemented.

Source: Project specification [1]

URC002: GUI Manual Mode

Description: When in manual control mode, the Operations GUI must allow the user to take manual control of the SeaFaults Mapping SeaFaults Mapping

Robot, including the ability to move forward and backward (in the SeaFaults Mapping SeaFaults Mapping Robot's frame of reference) and rotate in-place.

Rationale: Allows user-controlled navigation of the SeaFaults Mapping SeaFaults Mapping Robot in manual mode.

Priority: High.

Status: Implemented.

Source: Project specification [1]

URC003: GUI Move-to-Point Mode

Description: When in move-to-point mode, the Operations GUI must allow the user to specify a target destination for the SeaFaults Mapping SeaFaults Mapping Robot in survey-area coordinates, and commence SeaFaults Mapping SeaFaults Mapping Robot movement.

Rationale: Allows the user to define the parameters for and commence semi-autonomous movement of the SeaFaults Mapping SeaFaults Mapping Robot.

Priority: High.

Status: Implemented.

Source: Project specification [1]

URC004: GUI Autonomous Mode

Description: When in autonomous mode, the GUI must allow the user to initiate and halt the survey operation.

Rationale: Allows the user to commence and halt fully autonomous SeaFaults Mapping SeaFaults Mapping Robot movement.

Priority: High.

Status: Implemented.

Source: Project specification [1]

URC005: Mapping Toggle

Description: The software must include as part of the Operations GUI the ability to start and stop recording map data in any of the above operational modes.

Rationale: Allows the operator to prevent gathering of erroneous map data if and when manual repositioning of the SeaFaults Mapping SeaFaults Mapping Robot is required.

Priority: High.

Status: Implemented.

Source: Project specification [1]

URC006: Position Indicator

Description: The software should include in the GUI display an indication of the current position and orientation of the SeaFaults Mapping SeaFaults Mapping Robot relative to the survey area bounds.

Rationale: Allows the operator to track the SeaFaults Mapping SeaFaults Mapping Robot's position in the absence of line-of-site to the SeaFaults Mapping SeaFaults Mapping Robot's current position.

Priority: High.

Status: Implemented.

Source: Project specification [1]

URC007: Map View

Description: The software must include as part of the Operations GUI a basic, top-down overview of the currently mapped area, including known and estimated survey area boundaries and previously mapped obstacles and faultlines, as well as any pre-defined NGZs and operator specified NGZs during exploration.

Rationale: Provides reference to the user regarding the remaining area to be mapped and also SeaFaults Mapping SeaFaults Mapping Robot position relative to significant features in the survey area.

Priority: High.

Status: Implemented.

Source: Requirements elicitation meeting.

URC008: Emergency Stop

Description: The software must include as part of the Operations GUI a emergency stop button, functional in all control modes.

Rationale: Allows the user to intervene and halt SeaFaults Mapping SeaFaults Mapping Robot movement in the event of an emergency.

Priority: High.

Status: Implemented.

Source: Project specification [1]

URC009: Load XML GUI

Description: The software must include as part of the Operations GUI a pop-up dialog which allows loading of an XML file defining a full or partial map of the survey area.

Rationale: Allows loading of map data for subsequent completion.

Priority: High.

Status: Implemented.

Source: Requirements elicitation meeting.

3.1.2 Extension Requirements

URE001: Stop Notification

Description: The software should indicate to the user (via the Operations GUI) when an emergency stop has been triggered due to obstacle proximity.

Rationale: Prevents user confusion in the event of an unexpected stop.

Priority: Low.

Status: Unimplemented.

Source: Requirements elicitation meeting.

URE002: Manual Correction

Description: The Operations GUI should include the ability to manually specify the position and orientation of the SeaFaults Mapping SeaFaults Mapping Robot on the real-time display based on a visual estimate of the current position by the operator.

Rationale: Allows manual correction of the SeaFaults Mapping SeaFaults Mapping Robot's position estimate by the user.

Priority: Medium.

Status: Implemented.

Source: Requirements elicitation meeting.

URE003: Mapping Complete Notification

Description: The Operations GUI should indicate to the user when (in fully autonomous mode) the mapping operation is complete.

Rationale: Eases the user's cognitive burden by providing a simple indication when the mapping operation is completed.

Priority: Medium.

Status: Implemented.

Source: Requirements elicitation meeting.

URE004: Unreachable Area Notification

Description: The Operations GUI should indicate to the user which sections of the survey area have been determined to be unreachable due to obstacle placement, if any.

Rationale: Provides visual feedback to the user with respect to unreachable areas.

Priority: Medium.

Status: Unimplemented.

Source: Requirements elicitation meeting.

3.2 Mapping Output

3.2.1 Core Requirements

URC010: Map Conversion

Description: The software must include a function(a button show on GUI) to convert the real time data-model gathered during the operational phase and exported from the SeaFaults Mapping SeaFaults Mapping Robot to the required format(s).

Rationale: Allows the conversion to be performed by a user rather than a software developer.

Priority: Higg.

Status: Implemented.

Source: Project specification [1]

3.2.2 Extension Requirements

URE005: GUI Map Conversion

Description: The software should include a GUI tool to convert and display the real time data-model gathered during the operational phase and exported from the SeaFaults Mapping SeaFaults Mapping Robot to the required format(s).

Rationale: Allows the conversion to be performed by a novice user rather than a software developer.

Priority: Low.

Status: Implemented.

Source: Requirements elicitation meeting.

4 Functional Requirements

4.1 Movement

4.1.1 Core Requirements

FRC001: Position Estimation

Description: The software must be able to maintain an estimate of the SeaTank's current position based on a known starting position and an estimate of its movement since the start of the survey.

Rationale: Enables geolocation of mapped features.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC002: Exiting Survey Area Avoidance

Description: When in move-to-point or fully autonomous mode, the software must prevent the Sea Faults Mapping SeaFaults Mapping Robot from exiting the survey area by a significant margin. However, if the operator want to move the SeaFaults Mapping Robot out of the boundary, the mode must switch into the manual mode.

Rationale: Prevents inefficiency in mapping and damage to the Sea Tank.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC003: Collision Avoidance

Description: When in move-to-point or fully autonomous mode, the software must prevent the SeaFaults Mapping SeaFaults Mapping Robot from colliding with an obstacle with significant force by stopping the SeaFaults Mapping SeaFaults Mapping Robot at an appropriate distance.

Rationale: Prevents damage to the hardware.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC004: No-Go-Zone Avoidance

Description: When in move-to-point or fully autonomous mode, the software must prevent the SeaFaults Mapping SeaFaults Mapping Robot from entering any predefined NGZ.

Rationale: Prevents damage to the hardware.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC005: Manual Operation

Description: The software must allow the SeaFaults Mapping SeaFaults Mapping Robot to operate in manual mode, in which movement only occurs due to explicit user input from the Operations GUI.

Rationale: Allows fully manual operation of the SeaTank.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC006: Semi-Autonomous Operation

Description: The software must allow the SeaFaults Mapping SeaFaults Mapping Robot to operate in move-to-point mode, in which the SeaFaults Mapping SeaFaults Mapping Robot travels to a point specified by the user, avoiding any intervening obstacles.

Rationale: Allows semi-autonomous operation of the SeaTank.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC007: Autonomous Operation

Description: The software must allow the SeaFaults Mapping SeaFaults Mapping Robot to operate in fully autonomous mode, in which the SeaFaults Mapping SeaFaults Mapping Robot traverses the survey area without user intervention.

Rationale: Allows fully autonomous operation of the SeaTank.

Priority: High.

Status: Implemented.

Source: Project specification [1]

4.1.2 Extension Requirements**FRE001: Autonomous Correction**

Description: The software should allow for correction of the SeaTank's estimated position based on sensor inputs (for example, detection of the survey area boundary).

Rationale: Enables automatic correction of any accumulated error in the position estimate.

Priority: Low.

Status: Unimplemented.

Source: Requirements elicitation meeting.

FRE002: Manual Correction

Description: The software should allow for correction of the SeaTank's estimated position based on user input.

Rationale: Enables manual correction of any accumulated error in the position estimate.

Priority: High.

Status: Implemented.

Source: Project specification [1]

4.2 Mapping

4.2.1 Core Requirements

FRC008: Map Model

Description: The software must synthesise sensor input with the current position and orientation estimate of the SeaFaults Mapping SeaFaults Mapping Robot in order to build a data model representing the mapped area in real time.

Rationale: Enables the output of map data and informs the navigation of the SeaTank.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC009: Real-time Data Publishing

Description: The software must provide the current map data model to the Operations GUI for real time display such that the display can be updated at a rate no slower than once per second.

Rationale: Enables near real-time display of current map data.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC010: Time Limit

Description: The mapping operation must take no more than 20 minutes from commencement (user-specified start time) to completion (SeaFaults Mapping SeaFaults Mapping Robot reaches extraction area). This does not include exporting map data to file.

Rationale: Defines completion of mapping operation within a reasonable time frame.

Priority: Medium.

Status: Implemented.

Source: Project specification [1]

FRC011: Map Conversion

Description: The software must provide a means to export the real-time data model from the SeaFaults Mapping SeaFaults Mapping Robot to a workstation once the mapping operation has completed.

Rationale: Enables the conversion of the data model to the appropriate mapping user format(s).

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC012: Map Export

Description: The software must be able to convert the real-time data model created during the operational phase to a suitable output format within a reasonable amount of time (< 1 minute) after the mapping operation has completed. This phase of the operation can take place off-board. The output format (XML as defined by a DTD document) is to be provided by a third party.

Rationale: Enables generation of the required map data for mapping users.

Priority: High.

Status: Implemented.

Source: Project specification [1]

FRC013: Load XML File

Description: The software must be able load map data in the form of an XML file in order to update the onboard map data model. Loaded data must take

precedence over any map data generated by the SeaTank.

Rationale: Enables loading and subsequent completion of existing map data.

Priority: High.

Status: Implemented.

Source: Requirements elicitation meeting.

FRC014: Data Precision

Description: The error in mapped location of obstacle edges, faultlines and boundaries must not exceed 7 cm. This degree of error is expected due to accumulated error in location estimate and sensor precision.

Rationale: Defines required precision of mapped data.

Priority: High.

Status: Implemented.

Source: Project specification [1]

4.2.2 Extension Requirements

Nil.

4.3 Sensors

4.3.1 Core Requirements

FRC0015: Faultline Detection

Description: The SeaFaults Mapping SeaFaults Mapping Robot must be able to detect faultlines and distinguish these from survey area boundaries or three-dimensional obstacles. In addition, The software must be able to assign a depth value to faultlines based on a provided colour range mapping and the colour value input read from the appropriate sensor.

Rationale: Enables both the mapping functionality and obstacle avoidance.

Priority: High.

Status: Implemented.

Source: Project specification [1]

4.3.2 Extension Requirements

FRE003: Flexible Configuration

Description: The mapping of faultlines to colour value ranges should be configurable prior to software compile time.

Rationale: Improves maintainability and flexibility of the sensor system.

Priority: Low.

Status: Unimplemented.

Source: Requirements elicitation meeting.

FRE004: Discontinuity Detection

Description: The software should be able to detect and interpret discontinuities in the survey area boundary based on the relevant sensor input.

Rationale: Enhances the system's mapping ability and avoids damage to the hardware and/or operator intervention.

Priority: Medium.

Status: Unimplemented.

Source: Project specification [1]

5 Non-functional Requirements

5.1 Environment

5.1.1 Core Requirements

NRC001: Operation Environment Support

Description: The GUI and off-board conversion tools should support 64-bit Windows (7/8/10) and Mac (OSX10) operating systems, with the Java 8 Runtime installed.

Rationale: Allows for operation in the most common end-user environments.

Priority: Medium.

Status: Implemented.

Source: Project specification [1]

5.1.2 Extension Requirements

Nil.

5.2 Safety

5.2.1 Core Requirements

NRC002: Human Safety

Description: Operator workflows should ensure user safety.

Rationale: Compliance with relevant workplace health and safety guidelines [2].

Priority: High.

Status: Implemented.

Source: Project specification [1]

NRC003: Materials Safety

Description: The operator workflows and hardware handling should prevent damage to the hardware.

Rationale: Hardware is not the property of the development team, and should be protected accordingly.

Priority: High.

Status: Implemented.

Source: Project specification [1]

5.2.2 Extension Requirements

Nil.

5.3 Quality assurance

5.3.1 Core Requirements

NRC004: Testing

Description: Automated (unit) testing should be implemented where appropriate, and configured to run when the code is built.

Rationale: Automated testing assists in the detection of bugs introduced with code changes or misconfiguration.

Priority: High.

Status: Implemented.

Source: Project specification [1]

5.3.2 Extension Requirements

Nil.

6 Glossary

FRC Core functional requirement prefix

FRE Extension functional requirement prefix

GUI Graphical User Interface

HMI Human-machine interface

LeJOS The Lego Java Operating System

NGZ No-go-zone, forbidden to enter

NRC Core non-functional requirement prefix

NRE Extension non-functional requirement prefix

SFM Sea Floor Mapping

SRS Software Requirements Specification

XML eXtensible Markup Language

URC Core user requirement prefix

URE Extension user requirement prefix

WHS Workplace Health and Safety

7 References

- [1] Milanese, D and Weerasinghe, A 2016. *Software Engineering and Project - SeaFaults Mapping SeaFaults Mapping Robot*. Project specification version 1.0
- [2] Safe Work Australia 2016. *Guide to the model Work Health and Safety Act*. Accessed online 25/10/16
<<http://www.safeworkaustralia.gov.au/sites/SWA/about/Publications/Documents/717/Guide-to-the-WHS-Act-at-21-March-2016.pdf>>