

# Project Title: System Verification and Validation Plan for The Nursery Project

Aaron Billones, billonea  
Gillian Ford, fordg  
Juan Moncada, moncadaj  
Steven Ramundi, ramundis

November 2, 2022

Date	Version	Notes
2022-11-02	Juan Moncada, Aaron Billones, Steven Ramundi, Gillian Ford	Initial release

## Contents

## List of Tables

# 1 Symbols, Abbreviations and Acronyms

symbol	description
CR	Conveyor Functional Requirement
NFR	Non-Functional Requirement
PDR	Pot Dispensing Functional Requirement
TDR	Tray Dispensing Functional Requirement
VR	Verification Functional Requirement
SRS	Software Requirements Specification
TDST	Tray Dispenser Subsystem Test
PDST	Pot Dispenser Subsystem Test
CST	Conveyor Subsystem Test
VST	Verification Subsystem Test
SCT	Safety Critical Test
PT	Precision Test
RT	Reliability Test
EPET	Expected Physical Environment Test
LED	Light Emitting Diode
LCD	Liquid-Crystal Display
SRT	Speed Requirements Test
ART	Accessibility Requirements Test
LRT	Learning Requirements Test

Table 1: Acronyms

This document is the Verification & Validation Plan of the Pot-Pulator. This machine will be built for Sheridan Nurseries, filling trays with pots, preparing them to be populated with soil and seeds, to reduce the manual labour of the workers at the farm. The Verification & Validation Plan will consider the testing of each part of the Pot-Pulator, including the respective tray and pot droppers, the sensor verification section, and the conveyor belt.

## **2 General Information**

### **2.1 Summary**

The testing of the software in this report will be with C++ and Python. These will be used to perform specific tasks based on groups of related written statements. Testing with these types of software will involve specific testing involving multiple functions.

### **2.2 Objectives**

The main objective of this report is to build confidence in the software used for the Pot-Pulator. This is preparation for any errors in testing, so there is an opportunity to come up with solutions. When every possible error is worked through in advance, it increases the likelihood of the project's success.

### **2.3 Relevant Documentation**

Reports referenced or corresponding to this plan include the Pot-Pulator's Software Requirements Specification (SRS), Development Plan, Management Information Systems (MIS) and Module Guide (MG).

## **3 Plan**

The following section will outline the plan for system testing. The testing will be divided into 5 categories; the 4 separate subsystems that make up the system and the system.

### 3.1 Verification and Validation Team

The testing will be distributed evenly amongst the members of the team. The interdisciplinary aspect of this project means that certain members focused more on certain parts of the capstone, thus verification will be done evenly but there will be an emphasis on testing sections in which each member was not a major contributing partner. The following table shows the main point of focus of each group member and their area of focus on the project.

Team member	Area of Focus	Verification Area of Focus
Gillian Ford	Pot dropping and Verification	Conveyor and tray dispensing
Aaron Billones	Conveyor and tray dispensing	Pot dropping and Verification
Steven Ramundi	Conveyor and tray dispensing	Pot dropping and Verification
Juan Moncada	Pot dropping and Verification	Conveyor and tray dispensing

### 3.2 Milestones

The following are milestones for the corresponding testing including the date of expected completion.

Testing	expected date of completion
Pot dropping Subsystem	December 5th 2022
Tray dispensing Subsystem	December 5th 2022
Conveyor Subsystem	December 5th 2022
Verification Subsystem	January 10th 2023
Whole System	January 21th 2023

### 3.3 SRS Verification Plan

The design verification plan will be comprised of the verification of 4 of the core subsystems. Each of these subsystems will be tested individually to include all possible scenarios, once individual testing has been completed the system will be put together and tested as a whole.

The verification of system will take place alongside the end user. Sheridan Nurseries has been scheduled to have an early look at the unfinished system

so that all functional and non-functional requirements may be looked over and made sure to be fulfilled. This process will also allow for the foundations of new requirements that could be added before the final build of the system. There will also be multiple review sessions including by group members and peers. Peer review sessions will be to happen after every milestone has been reached

The review will go over the following checklist to make sure that the system is meeting all functional and non-functional requirement. The following is the SRS validation checklist.

Requirement Validate	Pass or fail
System interface should be clear, legible, and obvious. Subsystems are communicating with integration board. System is running smoothly, without the presence or audible or visual errors. There are no loose or exposed wires or electronics. There are no exposed gears, chains or moving parts.	

### 3.4 Design Verification Plan

Each of the subsystem will be tested individually. The following section will outline the individual testing for each individual testing subsystem. The following are the design validation checklists for the

further development of the project will warrant the addition to the following checklist. Checklist will be updated as new items are brought up.

#### 3.4.1 Conveyor

Requirement Validate	Pass or fail
Conveyor is stable so that movements from subsystems do not cause any vibration, tipping or other unwanted movement.	

### 3.4.2 Tray Dispenser

Requirement Validate	Pass or fail
Tray dispenser runs smoothly and is not translating any unwanted motion into the system.	

### 3.4.3 Pot Dispenser

Requirement Validate	Pass or fail
pot dispenser runs smoothly and is not translating any unwanted motion into the system.	

### 3.4.4 Verification

Requirement Validate	Pass or fail
Verification is able to display distinct subsystem failures verification operates consistently and is able to identify initiation and operation failures distinctly	

## 3.5 Verification and Validation Plan Verification Plan

As the project progresses there will be the requirement to add, remove or alter verification and Validation steps/procedures. In order to capture this as the system progresses in its completion when major verification and validation milestones are hit, they will be accompanied with the review of the verification and validation plan. This review will trigger any updates that will need to be put through.

## 3.6 Implementation Verification Plan

Implementation verification will take place through code walkthroughs. these will take place between members of the same subsystem focus. When there is a code to be uploaded and implemented into the system, the member which created the code will go through a in depth walk through with the other member working on the same subsystem. This is done for two main



reasons. The first, all members of the same subsystem focus will have in depth knowledge of all of the code that that subsystem has implemented into the system. Second, the code will be checked by a member who is of equal knowledge of the subsystem, its characteristics and requirements thus making inspection of the code more fruitful.

### **3.7 Automated Testing and Verification Tools**

Automated testing will take place through the use of a linter, most specifically the Python PEP 8 linter per the development plan. The code will be ran through the pep 8 linter before any milestone to emphasize the use of the most condensed and verified code before milestones. this was eluded to in the development plan for this project.

## **4 System Test Description**

### **4.1 Tests for Functional Requirements**

The following section includes system test cases for functional requirements. The tests are designed in such a way to ensure that all the functional requirements are met. For reference of the functional requirements, please review the SRS document.

#### **4.1.1 Pot-pulator Complete System Testing**

##### **1. PCST-01: Tray Dispenser Operation**

Control: Manual

Initial State: Some amount of trays in the stack.

Input: Sensor reads the status of tray stack.

Output: System operates or displays an error and stops all system operation.

Test Case Derivation: If the tray dispenser is empty or malfunctions, the system will output an error on a screen (tray dispenser malfunction/empty). If this error occurs, it will stop all sections of the machine. Otherwise, operate normally.

How test will be performed: Trays will be removed from the stack, then trays will be put in the stack. Operation and results on the screen will be observed.

## 2. PCST-02: **On Switch for Tray Dispenser Error**

Control: Manual

Initial State: Some number of trays in the stack, pot-pulator switch is set to off.

Input: Pot-pulator switch is set to on.

Output: System operates or remains idle and displays an error.

Test Case Derivation: If the tray dispenser is empty or malfunctions, the system will output an error on a screen (tray dispenser malfunction/empty). If this error occurs, the machine will not start. Otherwise, operate normally.

How test will be performed: Trays will be removed from the stack, then trays will be put in the stack. Operation and results on the screen will be observed.

## 3. PCST-03: **Tray Dispenser Reset**

Control: Manual

Initial State: Some number of trays in the stack, an error is displayed on the screen.

Input: Pot-pulator reset button is pressed.

Output: System operates or remains idle and displays an error.

Test Case Derivation: When there is a malfunction, the reset button will be pressed. If the malfunction is not solved, the system will output an error on the screen (tray dispenser malfunction/empty). Otherwise, system will start again normally.

How test will be performed: Reset button will be pressed during an error. Operation and results on the screen will be observed.

4. **PCST-04: Pot Dispenser Operation** Control: Manual

Initial State: Some amount of pots in the stack.

Input: Sensor reads the status of pot stack.

Output: System operates or displays an error and stops all system operation.

Test Case Derivation: If the pot dispenser is empty or malfunctions, the system will output an error on a screen (pot dispenser malfunction/empty). If this error occurs, it will stop all sections of the machine. Otherwise, operate normally.

How test will be performed: Pots will be removed from the stack, then trays will be put in the stack. Operation and results on the screen will be observed.

5. **PCST-05: On Switch for Pot Dispenser Error**

Control: Manual

Initial State: Some number of pots in the stack, pot-pulator switch is set to off.

Input: Pot-pulator switch is set to on.

Output: System operates or remains idle and displays an error.

Test Case Derivation: If the pot dispenser is empty or malfunctions, the system will output an error on a screen (pot dispenser malfunction/empty). If this error occurs, the machine will not start. Otherwise, operate normally.

How test will be performed: pots will be removed from the stack, then pots will be put in the stack. Operation and results on the screen will be observed.

6. **PCST-06: Pot Dispenser Reset**

Control: Manual

Initial State: Some number of pots in the stack, an error is displayed on the screen.

Input: Pot-pulator reset button is pressed.

Output: System operates or remains idle and displays an error.

Test Case Derivation: When there is a malfunction, the reset button will be pressed. If the malfunction is not solved, the system will output an error on the screen (pot dispenser malfunction/empty). Otherwise, system will start again normally.

How test will be performed: Reset button will be pressed during an error. Operation and results on the screen will be observed.

## 7. PCST-07: **Conveyer Operation**

Control: Manual

Initial State: Tray put on conveyer.

Input: Sensor reads the status of tray on conveyer.

Output: System operates or displays an error and stops all the system operation.

Test Case Derivation: If a component of the conveyer malfunctions, the system will output an error on a screen (conveyer malfunction). If this error occurs, it will stop all sections of the machine. Otherwise, operate normally.

How test will be performed: Trays will be fed onto the conveyer correctly, then trays will be placed stuck on purpose. Operation and results on the screen will be observed.

## 8. PCST-08: **Conveyer On Button**

Control: Manual

Initial State: Some trays are put on conveyer, switch is set to off.

Input: Pot-pulator switch is set to on.

Output: System operates or remains idle and displays an error.

Test Case Derivation: If the conveyer malfunctions, the system will output an error on a screen (conveyer malfunction) If this error occurs, the machine will not start. Otherwise, operate normally.

How test will be performed: Trays will be placed on the conveyer in incorrect positions, then in correct positions. Operation and results on

the screen will be observed.

#### 9. PCST-09: **Conveyer Reset**

Control: Manual

Initial State: Some trays are put on conveyer, switch is set to off.

Input: Pot-pulator switch is set to on.

Output: System operates or remains idle and displays an error.

Test Case Derivation: When there is a malfunction, the reset button will be pressed. If the malfunction is not solved, the system will output an error on the screen (conveyer malfunction). Otherwise, system will start again normally.

How test will be performed: Reset button will be pressed during an error. Operation and results on the screen will be observed.

#### 4.1.2 **Tray Dispenser Subsystem Testing**

#### 10. TDST-01: **Tray Stack Detection**

Control: Static, Manual

Initial State: No trays present in the stack. Trays present in the stack.

Input: Sensor reads the status of tray stack.

Output: Sends a signal/bit to microprocessor that tells the system there are/aren't trays present.

Test Case Derivation: The observed signal/bit is the expected value. The subsystem does not operate when no trays are present.

How test will be performed: All other sensors and subsystems will be switched off. All trays will be removed from the stack. The detection bit will be observed. Then trays will be placed in the stack, and the detection bit will be observed.

#### 11. TDST-02: **Operation from Tray Stack Detection**

Control: Dynamic, Manual

Initial State: Some amount of trays in the stack.

Input: Sensor reads the status of tray stack.

Output: Subsystem operates or remains idle.

Test Case Derivation: If no trays are present, the subsystem will not operate and remain ready in the idle state. Otherwise, operate normally.

How test will be performed: All other sensors and subsystems will be switched off. Trays will be removed from the stack and operation will be observed. Trays will be put in the stack and operation will be observed.

## 12. TDST-03: **Tray from Stack to Conveyor**

Control: Dynamic, Manual

Initial State: There is a stack of trays beside the vacant conveyor with the subsystem in idle position.

Input: Stack of trays.

Output: One tray from the stack is placed onto the end of the conveyor and returns to idle position.

Test Case Derivation: There is a tray in the correct designated position. The subsystem moves into the ready idle state to retrieve more trays.

How test will be performed: All other sensors and subsystems will be switched off. The system will be manually activated to retrieve one tray from its stack. The success or failure will be observed.

## 13. TDST-04: **Verify Tray Status on Conveyor**

Control: Dynamic, Manual

Initial State: Tray put on conveyor.

Input: Sensor reads the status of tray on conveyor.

Output: Subsystem continues operation or stops.

Test Case Derivation: Subsystem continues operation (when successful) or stops (when tray is stuck/fails to move on conveyor).

How test will be performed: Trays will be fed onto the conveyor correctly. Results will be observed. then trays will be placed stuck on purpose. Results will be observed.

#### 4.1.3 Pot Dispenser Subsystem Testing

##### 1. PDST-01: **Pot from Stack to Tray**

Control: Dynamic, Manual

Initial State: Pot dispenser loaded with two pots

Input: Simulated sensor input, two pot locations of tray directly below pot dispenser

Output: Pot dispenser will dispense two pots into designated pot locations on tray

Test Case Derivation: Pot dispenser will dispense pots into correctly positioned tray as it is prompted to

How test will be performed: Tray will be manually placed directly below pot dispenser with pot locations directly below pot stack. Machine will be turned on. Once pots are dispensed, pot dispenser will queue next two pots and tray will be removed.

##### 2. PDST-02: **Tray Sensing**

Control: Dynamic, Manual

Initial State: Mounted sensor with no object being sensed

Input: Manual placement of trays in front of sensor

Output: Sensor will output a signal when the presence of a tray is sensed

Test Case Derivation: Sensor will recognize that a tray is beneath the pot dispenser

How test will be performed: Tray will be manually placed directly in front of the mounted sensor. Signal output from sensor will be analyzed to determine sensor is aware of tray presence. Tray will then be moved forward and output from sensor will be analyzed to confirm sensor is aware that tray is moving. Tray will then be moved forward out of view

of sensor and output from sensor will be analyzed to confirm sensor is aware that tray is no longer present.

3. PDST-03: **Ability to Dispense 4" Diameter Pots**

Control: Static, Manual

How test will be performed: All specifications of pot dispenser will ensure that a 4" diameter pot is able to be dispensed. Measurements and reviews will be conducted by another member of the group any time a change is made to the dispenser during design and build phases. During build phase, test will be conducted on both pot dispensers.

4. PDST-04: **Ability to Store/Dispense Multiple Pots**

Control: Dynamic, Manual

Initial State: Pot dispenser loaded with pots

Input: Ten pots, simulated sensor input

Output: Pot dispenser will dispense two pots, reload with two pots from stack, dispense two pots, etc. until pot storage is empty

Test Case Derivation: Pot dispenser will complete 5 cycles of dispensing, storing and dispensing 10 pots in total

How test will be performed: Pot dispenser will be loaded with 10 pots, 5 per side. Sensor input will be simulated to indicate presence of tray. Pot dispenser will complete 5 cycles of dispensing, at which point pot storage will be spent.

5. PDST-05: **Pot Storage Detection**

Control: Dynamic, Manual

Initial State: Pot dispenser with no pots in storage

Input: N/A

Output: Pot storage sensor will output a signal when no trays are detected in pot storage

Test Case Derivation: Sensor will recognize that no pots are sensed in pot storage



How test will be performed: All pots will be removed from pot storage. Signal output from sensor will be analyzed to confirm sensor is aware that pot storage is empty.

#### **4.1.4 Conveyor Subsystem Testing**

##### **1. CST-01: Conveyor Ability to Move Trays**

Control: Dynamic, Manual

Initial State: Conveyor with tray placed at start

Input: Simulated inputs indicating conveyor can start

Output: Constant speed of conveyor motor and belt

Test Case Derivation: Conveyor will recognize tray is present on belt and able to move forward

How test will be performed: A single tray will be placed at the start point on the conveyor belt. The conveyor will receive signals indicating that there are no issues with any other subsystems and the tray can be moved forward. Behaviour of conveyor will be observed to confirm conveyor has moved tray from start to end with no stopping. Test will be interrupted if tray is unable to move forward due to physical interference or if conveyor stops.

##### **2. CST-02: Conveyor Ability to Stop**

Control: Dynamic, Manual

Initial State: Conveyor moving tray along belt

Input: Simulated signals from pot dispenser indicating tray is beneath pot dispenser

Output: Conveyor motor and belt come to a stop

Test Case Derivation: Conveyor will receive signal from pot dispenser, indicating the tray is beneath the pot dispenser, and stop movement of tray

How test will be performed: A single tray will be placed on the conveyor while conveyor is moving. A signal will be sent to the conveyor,

simulating a signal from the pot dispenser sensor which indicates that the tray is beneath the pot dispenser. Behaviour of conveyor will be observed o confirm conveyor brings tray to a stop when signal is recognized.

### 3. CST-03: **Conveyor Belt Friction**

Control: Static, Manual

Input: Mass of tray, tilt angle of conveyor belt

Output: Maximum acceleration of conveyor belt

Test Case Derivation: Maximum acceleration based on friction between conveyor belt and tray will be calculated and set acceleration/decceleration values will be determined

How test will be performed: 6 trays will be weighed and the mean mass will be calculated. Each tray will be placed on the conveyor belt one by one. For each tray, the conveyor belt will be tilted until the tray begins to slip, at which point the angle at which the belt is tilted will be recorded. The mean of these 6 angles will be calculated. These values will then be used to approximately determine the maximum acceleration the trays can undergo without slipping, and the acceleration of the conveyor motor will be set to not exceed 70% of this value.

#### 4.1.5 **Verification Subsystem Testing**

##### 1. VST-01: **Verify Correct Number of Pots in Tray**

Control: Dynamic, Manual

Initial State: One tray filled with some pots placed on the conveyor.

Input: Tray filled with a number of pots.

Output: Returns a count of the number of pots in the tray.

Test Case Derivation: The count read by the subsystem matches the actual number of pots in the tray.

How test will be performed: All other sensors and subsystems will be switched off. The subsystem will be manually activated to count the number of pots in the given tray as it moves on the conveyor. The

success or failure will be observed.

## 2. VST-02: **Operation from Verification Status**

Control: Dynamic, Manual

Initial State: Tray has completed counting the number of pots in the tray and deemed it success or fail.

Input: Status bit for success or fail of the pot verification step.

Output: Signal to tell the system to continue/stop operation based on status bit.

Test Case Derivation: The subsystem should signal the main processor to turn off other subsystems when there is a problem in verifying the number of pots (ie. *actual*  $\neq$  *target*).

How test will be performed: All other sensors and subsystems will be switched off. The subsystem will be manually activated to count the number of pots in the given tray as it moves on the conveyor. The success or failure will send a status bit to the main processor. The status bit will be observed.

## 4.2 Tests for Nonfunctional Requirements

### 4.2.1 Safety Critical Testing

#### 1. SCT-01: **Tray Dispenser Failure**

Type: Dynamic, Manual

Initial State: Tray dispenser functioning normally

Input: Tray dispenser disconnect

Output: System flags tray dispenser failure

How test will be performed: Tray dispenser will be manually disconnected from the system during operation. System response will be analyzed to confirm that a failure in the tray dispenser is recognized.

#### 2. SCT-02: **Pot Dispenser Failure**

Type: Dynamic, Manual

Initial State: Pot dispenser functioning normally

Input: Pot dispenser disconnect

Output: System flags pot dispenser failure

How test will be performed: Pot dispenser will be manually disconnected from the system during operation. System response will be analyzed to confirm that a failure in the pot dispenser is recognized.

### 3. SCT-03: **Conveyor Failure**

Type: Dynamic, Manual

Initial State: Conveyor functioning normally

Input: Conveyor disconnect

Output: System flags conveyor failure

How test will be performed: Conveyor will be manually disconnected from the system during operation. System response will be analyzed to confirm that a failure in the conveyor is recognized.

### 4. SCT-04: **Verification Failure**

Type: Dynamic, Manual

Initial State: Verification functioning normally

Input: Verification disconnect

Output: System flags verification failure

How test will be performed: Verification will be manually disconnected from the system during operation. System response will be analyzed to confirm that a failure in the verification is recognized.

## 4.2.2 Precision Testing

### 1. PT-01: **Tray Dispenser Precision**

Type: Static, Manual

How test will be performed: Tray storage will be filled to maximum capacity. Tray dispenser will place tray onto conveyor belt. Centre line of tray will be established and measured relative to centre line of

conveyor. Test will be repeated for all trays until tray storage is empty. Average offset measurement will be calculated.

## 2. PT-02: **Pot Dispenser Precision**

Type: Static, Manual

How test will be performed: Pot storage will be filled to 50% capacity. Pot dispenser will dispense pots into trays. Centred position of each opening in trays will be established and measured relative to centre line of pots. Test will be repeated for all pots until pot storage is empty. Average offset measurement will be calculated.

### 4.2.3 **Reliability Testing**

#### 1. RT-01: **Function Under Vibration**

Type: Static, Manual

How test will be performed: Pot-pulator will run continuously for 8 hours, with a tester ensuring pots and trays are available to the machine at all times so operation is never interrupted. Machine will be subject to vibrations resulting from conveyor motor, tray dispenser motors, and pot dispenser motors. Behaviour will be observed to ensure machine is able to function under long-term exposure to small amplitude vibration.

### 4.2.4 **Expected Physical Environment Testing**

#### 1. EPET-01: **Function Under Aerial Pollution**

Type: Static, Manual

How test will be performed: Pot-pulator will run continuously for 8 hours in an environment with an amount of aerial pollution similar to what is expected of the environment at Sheridan Nursery. A tester will be present to ensure that pots and trays are available to the machine at all times so operation is never interrupted. Behaviour will be observed to ensure machine is able to function for a long period of time in an environment with aerial pollution.

#### 4.2.5 Speed Requirements Testing

##### 1. SRT-01: **Acceleration Displacement of Trays**

Type: Dynamic, Manual

Initial State: Multiple trays (2-3) on the conveyor.

How test will be performed: The conveyor speed and acceleration will be modified while trays are on it. The trays should not show perpendicular axis movement along the conveyor while in higher than normal speeds.

##### 2. SRT-02: **Pot Dispensing Rate**

Type: Dynamic, Automatic

Initial State: Subsystem in idle state ready to dispense pots.

Input: Stack of pots.

Output: Pots dispensing.

Test Case Derivation: The preceding system for filling pots with soil operates at a certain rate which should be met by the Pot-purlator.

How test will be performed: Pots will be placed in the stack. When turned on, the pots will dispense at the desired rate (10 pots / 30 sec) and timed using a stopwatch. If the dispenser can meet these requirements then it passes.

##### 3. SRT-02: **Tray Dispensing Rate**

Type: Dynamic, Automatic

Initial State: Subsystem in idle state ready to dispense trays.

Input: Stack of Trays.

Output: Trays dispensing.

Test Case Derivation: The preceding system for filling pots with soil operates at a certain rate which should be met by the Pot-purlator.

How test will be performed: Trays will be placed in the stack. When turned on, the trays will dispense at the desired rate (1 tray / 30 sec) and timed using a stopwatch. If the dispenser can meet these requirements then it passes.

#### 4.2.6 Learning Requirements Testing

##### 1. LRT-01: Operational Simplicity

Type: Static

How test will be performed: An individual will be trained on operating the entire system. After training is complete, the individual must demonstrate that they are able to successfully and safely handle all possible scenarios that may occur during operation.

#### 4.2.7 Accessibility Requirements Testing

##### 1. ART-01: Audio and Visual Indicators

Type: Static, Manual

Inputs: Trigger signal.

Outputs: A corresponding light, sound, or screen display.

How test will be performed: LEDs, speakers, and LCD screens will be placed in the system and sent a signal to activate in a specified way.

### 4.3 Traceability Between Test Cases and Requirements

The following table outlines all of the system tests and how they relate to the relevant requirements. The requirements can be referenced in the SRS document.

Table 2: Corresponding Test IDs and Requirements

Test ID	Supporting Requirements
TDST-01	TDR3, TDR5
TDST-02	TDR4, TDR5
TDST-03	TDR2
TDST-04	TDR2
PDST-01	PDR2
PDST-02	PDR2
PDST-03	PDR3

PDST-04	PDR4
PDST-05	PDR5, PDR6
PCST-01	TDR1
PCST-02 TDR6	TDR5
PCST-03	TDR7
PCST-04	PDR1
PCST-05 PDR7	PDR6
PCST-06	PDR8
PCST-07	CR1
PCST-08	CR5
PCST-09	CR6
VST-01	VR1
VST-02	VR2
SCT-01	NFR12
SCT-02	NFR12
SCT-03	NFR12
SCT-04	NFR12
PT-01	NFR13
PT-02	NFR14
RT-01	NFR17
EPET-01	NFR20
LRT-01	NFR6
ART-01	NFR7
SRT-01	NFR8
SRT-02	NFR9
SRT-03	NFR10



## Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage etc. You should look to identify at least one item for each team member.

(a) Steven:

I will acquire skills related to dynamic testing of software and hardware.

(b) Juan:

I will acquire skills related to testing of hardware ware and software through the learning how to use a linter. This is not something that I have been exposed to before and find it interesting that it is something that exist. I also believe that there will be some development in critical thinking as when testing it is impossible to think of all the possible scenarios the system can go through thus a good set of critical thinking will be used in order to find all of the possible test cases.

Throughout this project I also wish to enhance my general coding skill. Which I have always had an interest in coding and have made attempts to use it whenever possible throughout my university career. I am excited to take on a project with a defined goal and no clear solution to improve my coding ability.

(c) Aaron:

Project management, team management, and time management are crucial to the success of the project. Working towards improving these skills will greatly benefit me and my team members in the future.

(d) Gillian:

Throughout this project, I am looking forward to learning skills from the other members of the team. As an Engineering physics student, the areas I would like to improve the most about are software and writing. I would like to be prepared for the industry and potentially working in a job that requires coding because it's something I enjoy, but I don't think it's something I am experienced enough in yet. In terms of writing and communication, it isn't often where I am required to write long reports anymore, and I like the challenge provided by creating this extent of documentation. I look forward to my writing skills to improve over the course of this capstone project.

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

(a) Steven:

An approach to acquiring dynamic testing skills is practicing simulating testing conditions with our microcontrollers. An alternative approach is familiarizing myself with microcontroller documentation and creating testing conditions from the knowledge I obtain. I will pursue the former approach, as I feel the trial-and-error this will enable will be a better approach to learning how to effectively test specific portions of the project, while having the added bonus of allowing me to learn more about the software and hardware I will be working with.

(b) Juan:

I believe critical thinking will be developed through trial and error as well in the review with peers as they will have different idea that will bring new insight to test cases that i might not have though of before. the linter skill will be developed through trial and error and some over the shoulder learning from those in my group that have dealt with this before.

(c) Aaron:

Working towards improving my team communication is something that I strive to do while participating in this capstone. This can be achieved in several ways. One approach that I will take is to strengthen my bond with the team members. In doing so, I will feel more comfortable in sharing ideas and being constructive when necessary. Another approach is to document what each team member is working on and some of the problems they have run into. This will increase my team awareness and ultimately lead to the success of the project. I chose to pursue this skill because I know sometimes I want to do a lot of work on my own without developing trust towards my team members. This skill will help me when I become an engineer and work with large teams where communication and collaboration is crucial.

(d) Gillian:

I know I can improve my coding skills effectively by working on something closely to another member of the team who is more skilled in that area. I think this will be very helpful because I can work on a portion of software on my own, and if I have questions or get stuck, I will have someone working on something similar who will understand the problem and can help me troubleshoot. I can improve writing and communication skills by reviewing and editing not only my own work, but the writing of my peers as well. After each deliverable is finished, I think it is very useful when we go over all the writing as a team and come up with ideas on how each section of the documentation can possibly be improved.