

Software Requirements Specification for The Nursery Project: The Pot-Pulator

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

October 5, 2022

Contents

1	Reference Material	iv
1.1	Table of Units	iv
1.2	Table of Symbols	iv
1.3	Abbreviations and Acronyms	v
1.4	Mathematical Notation	v
2	Introduction	1
2.1	Purpose of Document	1
2.2	Scope of Requirements	1
2.3	Behavioural Overview	2
2.4	Clients and Stakeholders	2
2.5	Characteristics of Intended Reader	3
2.6	Organization of Document	3
3	Project Constraints	3
3.1	System Context	3
3.2	User Responsibilities & Characteristics	3
3.3	System Constraints	4
4	Functional Requirements	4
4.1	Tray Dispensing Functional Requirements	4
4.2	Pot Dispensing Functional Requirements	5
4.3	Conveyor Requirements	5
4.4	Verification Requirements	5
5	Nonfunctional Requirements	6
5.1	Appearance Requirements	6
5.2	Usability Requirements	6
5.3	Learning Requirments	6
5.4	Accessibility Requirements	6
5.5	Speed Requirements	6
5.6	Safety Critical Requirements	7
5.7	Precision Requirements	7
5.8	Capacity Requirements	7
5.9	Reliability Requirements	7
5.10	Robustness and Fault Tolerance Requirements	7
5.11	Scalability or Extensibility Requirements	7
5.12	Expected Physical Environement Requirements	7
5.13	Maintenance Requirements	8
5.14	Requirements for Interacting with Adjacent Systems	8
5.15	Supportability Requirements	8

5.16	Compliance Requirements	8
6	Functional Requirements Likelihood of Change	8
6.1	Tray Dispensing Functional Requirements	8
6.2	Pot Dispensing Functional Requirements	9
6.3	Conveyor Functional Requirements	9
6.4	Verification Requirements	9

Revision History

Date	Version	Notes
2022-10-05	Juan Moncada, Aaron Billones, Steven Ramundi, Gillian Ford	Initial release

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	unit	SI
m	length	metre
kg	mass	kilogram
s	time	second
°C	temperature	centigrade
J	energy	joule
W	power	watt ($W = J s^{-1}$)

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
A_C	m^2	coil surface area
A_{in}	m^2	surface area over which heat is transferred in

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
ProgName	
T	Theoretical Model

1.4 Mathematical Notation

2 Introduction

The current method of preparing pots and trays to be filled with soil and populated with seeds at Sheridan Nurseries is a process with little to no automation, requiring many manual labour hours. Each year, 250,000 pots need to be placed in trays, each one being placed by hand. Recently, the owners of these farms have been finding it increasingly more difficult to fill these roles with enough workers to run the operation smoothly and meet production demands.

We aim to aid in this process by designing and implementing a machine that is able to fill trays with pots and prepare them for populating with soil and seeds. This will alleviate the reliance on manual labour and improve the overall efficiency of the farm.

This introduction will outline the document's purpose, the scope of requirements, characteristics of the intended reader, and the organization of document.

2.1 Purpose of Document

This Software Requirements Specification (SRS) document is to define the purpose of the Pot-Pulator. It will describe how the software will interact with the hardware, and how it is expected to perform, defining the executable requirements leading to the creation of a final product. This document should lay the important groundwork needed for every team member to understand the in-depth technical details of the product to be developed, in preparation for the design stage.

2.2 Scope of Requirements

The branches of the Pot-Pulator include a conveyer belt, tray allocation, pot dropping and verification. These four systems will be implemented to work together to create an efficient product. It will be reliable, easily configurable, affordable, and will greatly reduce the need for hands-on labour in relation to seed population.

The system implemented will be designed exclusively for Sheridan Nurseries, and for the pots and trays at their locations in a three-dimensional workspace. It is limited to usage on their round, 4-inch diameter pots and their respective trays. Considering the conditions of the greenhouses this machine will be used in, material degradation, water resistance, and pressure will all be negligible.

In order to maintain speed of the pots being placed in trays in a timely manor, one of the goals of the machine is to fill each tray with 10 pots every 30 seconds. This is the time it will take for the trays to be populated and move down the conveyer belt. With this goal in place, the machine will only need to be refilled with pots and trays every 15 minutes, greatly

reducing the amount of hands-on labour from the workers at Sheridan Nurseries.

After the initial completion of the goal associated with the 4-inch pots, larger pots and trays may be considered as a future goal. There will always be the limitation of the requirement of a rim on the top of the pot, for the machine to easily grasp onto when lifting and dropping.

2.3 Behavioural Overview

The Pot-Pulator will have 4 branches working together to accomplish the task. First, the trays and pots will be manually filled into a section of the machine, ideally 30 trays and 300 pots. The first section of the machine will drop a tray, and the conveyer belt will move it to the pot dropping section.

The conveyer belt will stop once it senses the tray is in the correct position. The machine will drop pots until all 10 spots are filled. The conveyer belt will move the tray according to the needed positions for pot dropping, dispensing 2 pots into the allocated positions, moving forward 4 inches at a time until the process is completed.

The tray filled with pots will then move into a verification system, to verify the pots have been placed in the correct positions. The conveyer belt will then move it to an existing conveyer system, which will lead to the collection of completed trays. The only maintenance this machine needs is the refill of pots and trays every 15 minutes, and the collection of completed trays at the end of the final conveyer belt.

2.4 Clients and Stakeholders

This system provides a solution for all nursery applications that are not working on the cutting edge. For those nurseries that have an existing assembly line to fulfil their propagation this is a great solution as other solutions replace the assembly line in its entirety, this solution instead adds to it. Clients include:

- Sheridan Nurseries manager
- Sheridan Nurseries owner

Stakeholders include:

- Dr. Smith (Project Supervisor).
- Juan Moncada, Gillian Ford, Aaron Billiones, and Steven Ramundi, (Project Proposers).
- Nicholas Annable (Supervising TA)

2.5 Characteristics of Intended Reader

This SRS document is written for the individuals directly involved with the development of the Pot-Pulator. This includes every member of the team; all being involved with the development of the project. It is useful as reference to the functionality of in-depth details regarding the project, and will be read, reviewed, and maintained as the project progresses by all team members.

In general, any users of this document should have an undergraduate level understanding of software and hardware system functionality, physics, and mathematics.

2.6 Organization of Document

This SRS template is based on [Smith and Lai \(2005\)](#); [Smith et al. \(2007\)](#). These templates have been modified to reflect rubric requirements, and sections have been reorganized to follow the LazyBots student sample SRS. Sections 3 and 4 describe the project constraints, and show the context and functional decomposition diagrams. Section 5 is dedicated to functional requirements. This section will define functional requirements separated in terms of the systems which make up the Pot-Pulator. Section 6 describes the non-functional requirements in terms of look and feel, usability, performance, operational, maintainability and portability, security, cultural and political, and legal requirements. Section 7 evaluates each functional requirement on its likelihood to change, and specifies how the requirement may change if it is not very unlikely to.

3 Project Constraints

This section identifies the interfaces between the system and its environment, describes the user characteristics, lists the system constraints, and provides assumptions relating to the system/environment.

3.1 System Context

The system will consist of a stack of pots, stack of trays, and the mechanism that will autonomously populate the trays with the pots. The trays will move along a conveyor system that operates at approximately 3 feet and will act as the operating workspace for the device. Once the trays are populated with pots, they are fed into a soil potting machine in preparation for planting.

3.2 User Responsibilities & Characteristics

The user will be responsible for refilling the stacks of trays and pots when the supply is depleted. The user will also be responsible for knowing how the entire system will operate

within each state. For example, if a tray fails the validation step, the user must be capable of removing the defect and resume the operation of the machine.

The characteristics of the user will ensure smooth operation and workplace safety. The user should be able to lift a sufficient weight and have a good understanding of basic functionality of the system.

3.3 System Constraints

[System constraints differ from other type of requirements because they limit the developers' options in the system design and they identify how the eventual system must fit into the world. This is the only place in the SRS where design decisions can be specified. That is, the quality requirement for abstraction is relaxed here. However, system constraints should only be included if they are truly required. —TPLT]

4 Functional Requirements

This section provides the functional requirements of the Pot-Pulator, separated into tray dispensing, pot dispensing, conveyor, and verification requirements.

4.1 Tray Dispensing Functional Requirements

- TDR1: The tray dispenser must start when the Pot-Pulator start is pressed.
- TDR2: The tray dispenser must be able to place a standard tray from the loaded stack onto the conveyor belt.
- TDR3: The tray dispenser must be able to store multiple trays at a time.
- TDR4: The tray dispenser must stop operating once it has run out of trays.
- TDR5: The tray dispenser must be able to recognize when trays need to be reloaded and notify an operator.
- TDR6: The tray dispenser must stop when an error arises (verification failed, pot dispenser malfunction/empty, conveyor malfunction).
- TDR7: The tray dispenser must be able to reset after it has been stopped due to error or needs to reload

4.2 Pot Dispensing Functional Requirements

- PDR1: The pot dispenser must start when the Pot-Pulator start is pressed.
- PDR2: The pot dispenser must be able to place a pot into an empty place in a tray.
- PDR3: The pot dispenser must be able to dispense a 4" diameter pot.
- PDR4: The pot dispenser must be able to store multiple pots at a time.
- PDR5: The pot dispenser must stop operating once it has run out of pots.
- PDR6: The pot dispenser must be able to recognize when pots need to be reloaded and notify an operator.
- PDR7: The pot dispenser must stop when an error arises (verification failed, tray dispenser malfunction/empty, conveyor malfunction).
- PDR8: The pot dispenser must be able to reset after it has been stopped due to an error or needs to reload.

4.3 Conveyor Requirements

- CR1: The conveyor must start when the Pot-Pulator start is pressed.
- CR2: The conveyor must be able to move a tray from the tray dispensing station to the verification station.
- CR3: The conveyor must be able to stop when a tray is being placed on the conveyor, or when a pot is being placed in a tray.
- CR4: The conveyor material must have a high enough friction such that the trays do not slide while in contact with the belt.
- CR5: The conveyor must stop when an error arises (verification failed, pot dispenser malfunction/empty, tray dispenser malfunction/empty).
- CR6: The conveyor must be able to reset after it has been stopped due to an error or pots/trays need to be reloaded.

4.4 Verification Requirements

- VR1: The verification must check that only one tray is being output by the conveyer belt, and the tray is filled with exactly 10 pots, such that $n_{traysout} = 1$ & $n_{potsout} = 10$
- VR2: The verification must notify all other systems if verification failes, such that $n_{traysout} \neq 1$ or $n_{potsout} \neq 10$

5 Nonfunctional Requirements

This section provides the non-functional requirements of the Pot-Pulator

5.1 Appearance Requirements

- NFR1: All electrical equipment and electronics must be well covered and protected. The user must not have access to equipment.
- NFR2: All wiring must be tucked away and not accessible to avoid potential electrical failure.
- NFR3: All moving parts must be covered and protected. Moving parts should be covered to protect both the mechanism and the safety of the operator.

5.2 Usability Requirements

- NFR4: System must have tray and pot refill locations visible and accessible in order to provide visual verification of capacity status.
- NFR5: System output (end of conveyor) must be clear and visible in order to provide visual verification and access to failed outputs.

5.3 Learning Requirements

- NFR6: System must be simple to operate, requiring minimal training to operate effectively ($< 1 \text{ hour}$).

5.4 Accessibility Requirements

- NFR7: System must have both audible and visible signal outputs for each system status.

5.5 Speed Requirements

- NFR8: Conveyor system must not accelerate in a manner that would shift the position of the tray on the conveyor belt. A shift in the position of the tray could result in a misalignment and potential error.
- NFR9: The pot dispenser must dispense pots at a rate of 10 pots every 30 seconds, such that $t_{cycle} = 30 \text{ seconds}$ & $n_{pots}(t) = 300 - t/3$; $t \in \{30, 60, 90, \dots, 900\}$. This will allow the machine to meet the output goal required.
- NFR10: The tray dispenser must dispense trays at a rate of 1 tray every 30 seconds, such that $t_{cycle} = 30 \text{ seconds}$ & $n_{trays}(t) = 30 - t/30$; $t \in \{30, 60, 90, \dots, 900\}$. This will allow the machine to meet the output goal required.

5.6 Safety Critical Requirements

NFR11: System must have emergency cut off. In case of any emergency, this will trip off all power to the Pot=Pulator.

NFR12: System must be able to locate and identify failures within each independent subsystem.

5.7 Precision Requirements

NFR13: Conveyor must centre trays before potting dispenser withing 1 cm of centered position. This allows the tray to enter the pot dispensing system within tolerance for a pot to be dropped in.

NFR14: Pot dispenser should dispense pots within a 0.5 cm radius of centered position.

5.8 Capacity Requirements

NFR15: The pot dispenser must be able to hold a maximum of 300 pots at all times, such that $n_{pots}(t) \leq 300$ This will allow the machine to meet the output goal required.

NFR16: The tray dispenser must be able to hold a maximum of 30 trays at all times, such that $n_{trays}(t) \leq 30$ This will allow the machine to meet the output goal required.

5.9 Reliability Requirements

NFR17: System must be able to operate under constant low and high frequency vibration (small amplitude).

5.10 Robustness and Fault Tolerance Requirements

NFR18: Components must withstand 250,000 cycles. System should call for replacement parts after a full season of operation.

5.11 Scalability or Extensibility Requirements

NFR19: System must fit into the existing assembly line and should take minimal effort to implement (< 1 day).

5.12 Expected Physical Environment Requirements

NFR20: System must withstand operating in a room with high arial polution including dust, dirt, and small amounts of water.

5.13 Maintenance Requirements

NFR21: System must be built for ease of maintenance. High wear parts should be easily accessible.

5.14 Requirements for Interacting with Adjacent Systems

NFR22: System must operate at same speed as adjacent systems.

NFR23: System must operate at the same conveyor height as the current system to maintain continuity between systems.

5.15 Supportability Requirements

NFR24: System documentation must be available to troubleshoot, diagnose and fix common issues and replace high wear parts.

5.16 Compliance Requirements

NFR25: System must follow electronics system safety requirements.

6 Functional Requirements Likelihood of Change

6.1 Tray Dispensing Functional Requirements

TDR1: Very unlikely to change, key implementation of system.

TDR2: Very unlikely to change, key implementation of system.

TDR3: Very unlikely to change, key implementation of system.

TDR4: Very unlikely to change, ensures safety of operator reloading trays.

TDR5: Very unlikely to change, key implementation of system.

TDR6: Very unlikely to change, ensures safety of operator and prevents machine from dispensing trays while conveyor and/or pot dispenser are not operational.

TDR7: Unlikely to change, ensures safety of operator and robustness of Pot-Pulator. May change to be integrated with start command or resume command.

6.2 Pot Dispensing Functional Requirements

PDR1: Very unlikely to change, key implementation of system.

PDR2: Very unlikely to change, key implementation of system.

PDR3: Unlikely to change, pot size is standard across the nursery. New pots may be introduced at a future time.

PDR4: Very unlikely to change, key implementation of system.

PDR5: Very unlikely to change, ensures safety of operator reloading pots.

PDR6: Very unlikely to change, key implementation of system.

PDR7: Very unlikely to change, ensures safety of operator and prevents machine from dispensing pots while conveyor and/or tray dispenser are not operational.

PDR8: Unlikely to change, ensures safety of operator and robustness of Pot-Pulator. May change to be integrated with start command or resume command.

6.3 Conveyor Functional Requirements

CR1: Very unlikely to change, key implementation of system.

CR2: Unlikely to change, ensures accuracy of placement of trays and pots. May change to place trays on conveyor as conveyor is moving.

CR3: Very unlikely to change, key implementation of system.

CR4: Very unlikely to change, ensures safety of operator and prevents machine from dispensing pots while conveyor and/or tray dispenser are not operational.

CR5: Unlikely to change, ensures safety of operator and robustness of Pot-Pulator. May change to be integrated with start command or resume command.

6.4 Verification Requirements

VR1: Very unlikely to change, key implementation of system.

VR2: Very unlikely to change, key implementation of system.

References

- W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.
- W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.

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 this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

(a) Steven:

Steven will acquire skills related to programming microcontrollers and interfacing hardware and software.

(b) Juan:

This project presents an unique challenge when it comes to group management and teamwork, this being one of few interdisciplinary capstone it is the only capstone project(from my knowledge) that include groups from vastly different engineering streams. While ECE and CAS Share many common teachings, engineering physics lacks that familiarity. this however is ,in my oppinion, is a great representation of the future that is to come. When working in industry there are very few instances that you will be surrounded by a those with similar knowledge sets and backgrounds, most of the time the team you will work with will be of diverce experience and backgrounds and understanding how to work as a team is detrimental to good productivity. I beleive throughout this project I will further develop my project management skills, they will be put to the test as each menmebr of the team will have their own expertise and we will be fusing knowledge together to create a solution. Team management and general coding.

Through out this project I also with 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:

(d) Gillian:

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 programming skills is researching and practicing specific programming exercises that are specific to the environment I will be working in. An alternative approach is familiarizing myself with microcontroller documentation and creating solutions from the knowledge I obtain. I will pursue the former approach, as I find myself able to learn more effectively and efficiently with a hands on approach, rather than reading through documentation. Approaches to learning how to interface hardware and software include reviewing previous coursework, and immitating examples I find as practice. I will chose the latter option for the same reason stated above.

(b) Juan:

Project management skill can be developed throughout this project in many different ways. The first will be in taking a heavy rool in the logistical and administrative side of the team, and another is to run the project management through an issue tracking system. like the one provided by git hub. this will be an instrumental tool in organization and task management. this might be tedious as it adds an extra step to all tasks as they will have to be updated and statused regularly, but the pros outways the cons as this will help with the overall management of the team. Of the two solutions both will be attempted however more emphasis will be put toward the task tracking software as it add the added benefit of simulating how large corporations are running their buissness.// In order to improve general coding there are two approaches that can be taken on. the first being that I can take on a soley software portion and be responsible for its fruition, or I can volunteer to take one of these software challenges on with another member in my group who is better versed in coding so that we may work hand in hand. This way if there will be support and guidance throughout and could offer a solution faster. The second option in this case seems more reasonable as there are hard deadline for this project and the system itself is already ambitious.

(c) Aaron:

(d) Gillian: