Department of Computer Science and Engineering  
The University of Texas at Arlington

System Test Plan

Team: Ink3D

Project: 3-D Printer Fabrication System

Team Members:   
*Daniel Lain*

*Timothy Edmonson*

*Shawn Simonson*

*Jesse Bowles*

Table of Contents

[Introduction 10](#_Toc382756448)

[1.1 Purpose 10](#_Toc382756449)

[1.2 Scope 10](#_Toc382756450)

[2 References 10](#_Toc382756451)

[2.1 System Requirements Specification 10](#_Toc382756452)

[2.1.1 Customer Requirements 11](#_Toc382756453)

[2.1.2 Packaging Requirements 12](#_Toc382756454)

[2.1.3 Performance Requirements 13](#_Toc382756455)

[2.1.4 Safety Requirements 13](#_Toc382756456)

[2.1.5 Maintenance and Support Requirements 14](#_Toc382756457)

[2.1.6 Other Requirements 14](#_Toc382756458)

[2.2 Architecture Design Specification 15](#_Toc382756459)

[2.2.1 Interface Layer 16](#_Toc382756460)

[2.2.2 Pre Processing Layer 16](#_Toc382756461)

[2.2.3 Processing Layer 16](#_Toc382756462)

[2.2.4 Post Processing Layer 16](#_Toc382756463)

[2.2.5 Printer Control Layer 16](#_Toc382756464)

[2.2.6 Printer Feedback Layer 16](#_Toc382756465)

[2.2.7 Communications Layer 17](#_Toc382756466)

[2.2.8 Dataflow Definitions 17](#_Toc382756467)

[2.2.9 Layer Requirements Traceability 18](#_Toc382756468)

[2.3 Detail Design Specification 19](#_Toc382756469)

[2.3.1 Prologue 19](#_Toc382756470)

[2.3.2 Data Flow Definitions 21](#_Toc382756471)

[2.3.3 Module Requirements Traceability 22](#_Toc382756472)

[3 Test Items 25](#_Toc382756473)

[3.1 Product Version 25](#_Toc382756474)

[3.1.1 Stage one 25](#_Toc382756475)

[3.1.2 Stage Two 25](#_Toc382756476)

[3.1.3 Stage Three 25](#_Toc382756477)

[3.2 Design Decomposition 27](#_Toc382756478)

[3.3 Unit Tests 28](#_Toc382756479)

[3.3.1 Import GUI 28](#_Toc382756480)

[3.3.2 Material GUI Tests 28](#_Toc382756481)

[3.3.3 Printer Configuration GUI 28](#_Toc382756482)

[3.3.4 Extruder Configuration GUI 28](#_Toc382756483)

[3.3.5 Print Configuration GUI 29](#_Toc382756484)

[3.3.6 Print Job GUI 29](#_Toc382756485)

[3.3.7 Status GUI 29](#_Toc382756486)

[3.3.8 Import Controller 30](#_Toc382756487)

[3.3.9 Material Controller 30](#_Toc382756488)

[3.3.10 Printer Configuration Controller 31](#_Toc382756489)

[3.3.11 Print Configuration Controller 31](#_Toc382756490)

[3.3.12 Extruder Controller 32](#_Toc382756491)

[3.3.13 Print Job Controller 32](#_Toc382756492)

[3.3.14 Status Controller 33](#_Toc382756493)

[3.3.15 Command Structure 33](#_Toc382756494)

[3.3.16 Subsection Module 34](#_Toc382756495)

[3.3.17 File Translation Module 34](#_Toc382756496)

[3.3.18 Slicing Engine Wrapper 34](#_Toc382756497)

[3.3.19 Parser Module 35](#_Toc382756498)

[3.3.20 Unification Module 35](#_Toc382756499)

[3.3.21 Printer State Controller Module 35](#_Toc382756500)

[3.3.22 Dispatch Module 36](#_Toc382756501)

[3.3.23 Serialization Module 36](#_Toc382756502)

[3.3.24 De-Serialization Module 36](#_Toc382756503)

[3.3.25 TxRx Module 36](#_Toc382756504)

[3.4 Component Tests 37](#_Toc382756505)

[3.4.1 GUI Subsystem 37](#_Toc382756506)

[3.4.2 Controller Subsystem 37](#_Toc382756507)

[3.4.3 Database Subsystem 38](#_Toc382756508)

[3.4.4 Normalization Subsystem 38](#_Toc382756509)

[3.4.5 G-Code Preparation Subsystem 39](#_Toc382756510)

[3.4.6 Communication Subsystem 39](#_Toc382756511)

[3.5 Integration Tests 40](#_Toc382756512)

[3.6 Validation Tests 41](#_Toc382756513)

[4 Risks 42](#_Toc382756514)

[4.1 Hardware Unreliable 42](#_Toc382756515)

[4.1.1 Description 42](#_Toc382756516)

[4.1.2 Impact 42](#_Toc382756517)

[4.1.3 Severity 42](#_Toc382756518)

[4.1.4 Strategy 42](#_Toc382756519)

[4.2 Third Party Software Unreliable 42](#_Toc382756520)

[4.2.1 Description 42](#_Toc382756521)

[4.2.2 Impact 42](#_Toc382756522)

[4.2.3 Severity 42](#_Toc382756523)

[4.2.4 Strategy 42](#_Toc382756524)

[4.3 Untested Input Combinations 42](#_Toc382756525)

[4.3.1 Description 42](#_Toc382756526)

[4.3.2 Impact 42](#_Toc382756527)

[4.3.3 Severity 42](#_Toc382756528)

[4.3.4 Strategy 43](#_Toc382756529)

[4.4 Defect Fixes May Cause Other New Defects 43](#_Toc382756530)

[4.4.1 Description 43](#_Toc382756531)

[4.4.2 Impact 43](#_Toc382756532)

[4.4.3 Severity 43](#_Toc382756533)

[4.4.4 Strategy 43](#_Toc382756534)

[5 Features to be Tested 44](#_Toc382756535)

[5.1 STL File Input 44](#_Toc382756536)

[5.1.1 Description 44](#_Toc382756537)

[5.1.2 Priority 44](#_Toc382756538)

[5.2 Graphical User Interface 44](#_Toc382756539)

[5.2.1 Description 44](#_Toc382756540)

[5.2.2 Priority 44](#_Toc382756541)

[5.3 Generate Machine Instructions 44](#_Toc382756542)

[5.3.1 Description 44](#_Toc382756543)

[5.3.2 Priority 44](#_Toc382756544)

[5.4 Issue Machine Instructions 44](#_Toc382756545)

[5.4.1 Description 44](#_Toc382756546)

[5.4.2 Priority 44](#_Toc382756547)

[5.5 Monitor Temperature 44](#_Toc382756548)

[5.5.1 Description 44](#_Toc382756549)

[5.5.2 Priority 44](#_Toc382756550)

[5.6 Monitor Position 45](#_Toc382756551)

[5.6.1 Description 45](#_Toc382756552)

[5.6.2 Priority 45](#_Toc382756553)

[5.7 Adhere to Material Constraints 45](#_Toc382756554)

[5.7.1 Description 45](#_Toc382756555)

[5.7.2 Priority 45](#_Toc382756556)

[5.8 Identify Materials 45](#_Toc382756557)

[5.8.1 Description 45](#_Toc382756558)

[5.8.2 Priority 45](#_Toc382756559)

[5.9 Identify Shapes 45](#_Toc382756560)

[5.9.1 Description 45](#_Toc382756561)

[5.9.2 Priority 45](#_Toc382756562)

[5.10 Determine Shape of Support Material Structure 45](#_Toc382756563)

[5.10.1 Description 45](#_Toc382756564)

[5.10.2 Priority 45](#_Toc382756565)

[5.11 Create Printing Path 45](#_Toc382756566)

[5.11.1 Description 45](#_Toc382756567)

[5.11.2 Priority 45](#_Toc382756568)

[5.12 Database Interface 46](#_Toc382756569)

[5.12.1 Description 46](#_Toc382756570)

[5.12.2 Priority 46](#_Toc382756571)

[5.13 Store & Load Material Records 46](#_Toc382756572)

[5.13.1 Description 46](#_Toc382756573)

[5.13.2 Priority 46](#_Toc382756574)

[5.14 Slice Geometry into Thickness Levels 46](#_Toc382756575)

[5.14.1 Description 46](#_Toc382756576)

[5.14.2 Priority 46](#_Toc382756577)

[5.15 Monitor Flow Sensors 46](#_Toc382756578)

[5.15.1 Description 46](#_Toc382756579)

[5.15.2 Priority 46](#_Toc382756580)

[5.16 Allow for UV Head Polymerization 46](#_Toc382756581)

[5.16.1 Description 46](#_Toc382756582)

[5.16.2 Priority 46](#_Toc382756583)

[5.17 Fill Density 46](#_Toc382756584)

[5.17.1 Description 46](#_Toc382756585)

[5.17.2 Priority 47](#_Toc382756586)

[5.18 Software Installer 47](#_Toc382756587)

[5.18.1 Description 47](#_Toc382756588)

[5.18.2 Priority 47](#_Toc382756589)

[5.19 Host Software to Printer Connection 47](#_Toc382756590)

[5.19.1 Description 47](#_Toc382756591)

[5.19.2 Priority 47](#_Toc382756592)

[5.20 Startup Time 47](#_Toc382756593)

[5.20.1 Description 47](#_Toc382756594)

[5.20.2 Priority 47](#_Toc382756595)

[5.21 STL Import Time 47](#_Toc382756596)

[5.21.1 Description 47](#_Toc382756597)

[5.21.2 Priority 47](#_Toc382756598)

[5.22 Object Processing Time 47](#_Toc382756599)

[5.22.1 Description 47](#_Toc382756600)

[5.22.2 Priority 47](#_Toc382756601)

[5.23 GUI Responsiveness 47](#_Toc382756602)

[5.23.1 Description 47](#_Toc382756603)

[5.23.2 Priority 47](#_Toc382756604)

[5.24 Real Time Sensor Monitoring 48](#_Toc382756605)

[5.24.1 Description 48](#_Toc382756606)

[5.24.2 Priority 48](#_Toc382756607)

[5.25 Temperature Cutoff Threshold 48](#_Toc382756608)

[5.25.1 Description 48](#_Toc382756609)

[5.25.2 Priority 48](#_Toc382756610)

[5.26 Printing Area Restrictions 48](#_Toc382756611)

[5.26.1 Description 48](#_Toc382756612)

[5.26.2 Priority 48](#_Toc382756613)

[5.27 Material Database 48](#_Toc382756614)

[5.27.1 Description 48](#_Toc382756615)

[5.27.2 Priority 48](#_Toc382756616)

[6 Features Not to be Tested 49](#_Toc382756617)

[6.1 Monitor Door Switch 49](#_Toc382756618)

[6.1.1 Description 49](#_Toc382756619)

[6.1.2 Priority 49](#_Toc382756620)

[6.2 Graphical Object Models 49](#_Toc382756621)

[6.2.1 Description 49](#_Toc382756622)

[6.2.2 Priority 49](#_Toc382756623)

[6.3 Abstract Hardware Interface 49](#_Toc382756624)

[6.3.1 Description 49](#_Toc382756625)

[6.3.2 Priority 49](#_Toc382756626)

[6.4 Modular and Scalable Design 49](#_Toc382756627)

[6.4.1 Description 49](#_Toc382756628)

[6.4.2 Priority 49](#_Toc382756629)

[7 Testing Approach 50](#_Toc382756630)

[7.1 Overall Strategy 50](#_Toc382756631)

[7.2 Configurations 50](#_Toc382756632)

[7.3 Regression 50](#_Toc382756633)

[7.4 Metrics 50](#_Toc382756634)

[7.5 Special Requirements 51](#_Toc382756635)

[8 Item Pass/Fail Criteria 52](#_Toc382756636)

[8.1 Unit Testing Level 52](#_Toc382756637)

[8.1.1 Pass 52](#_Toc382756638)

[8.1.2 Fail 52](#_Toc382756639)

[8.2 Component Testing Level 52](#_Toc382756640)

[8.2.1 Pass 52](#_Toc382756641)

[8.2.2 Fail 52](#_Toc382756642)

[8.3 Integration Testing Level 52](#_Toc382756643)

[8.3.1 Pass 52](#_Toc382756644)

[8.3.2 Fail 52](#_Toc382756645)

[9 Test Deliverables 53](#_Toc382756646)

[9.1 JUnit tests 53](#_Toc382756647)

[9.2 Test output reports 53](#_Toc382756648)

[9.3 Coverage Report 53](#_Toc382756649)

[9.4 Test Results 53](#_Toc382756650)

[10 Test Schedule 54](#_Toc382756651)

[11 Approvals 55](#_Toc382756652)

List of Tables

Table # Title Page #

[Table 2‑1: Customer Requirements 12](#_Toc382756406)

[Table 2‑2 : Packaging Requirements 12](#_Toc382756407)

[Table 2‑3 : Performance Requirements 13](#_Toc382756408)

[Table 2‑4 : Safety Requirements 13](#_Toc382756409)

[Table 2‑5 : Maintenance and Support Requirements 14](#_Toc382756410)

[Table 2‑6 : Other Requirements 15](#_Toc382756411)

[Table 2‑7: Architecture Dataflow Definitions 18](#_Toc382756412)

[Table 2‑8 : Layer Requirements Traceability 19](#_Toc382756413)

[Table 2‑9 : Data Flow Definitions 21](#_Toc382756414)

[Table 2‑10 : UI Layer Requirements Traceability 22](#_Toc382756415)

[Table 2‑11 : Processing Layers Requirements Traceability 23](#_Toc382756416)

[Table 2‑12: Printer State, Comm., and Printer Feedback Requirements Traceability 24](#_Toc382756417)

[Table 3‑1 : Import GUI Tests 28](#_Toc382756418)

[Table 3‑2 : Material GUI Tests 28](#_Toc382756419)

[Table 3‑3 : Printer Configuration GUI Tests 28](#_Toc382756420)

[Table 3‑4 : Extruder Configuration GUI Tests 29](#_Toc382756421)

[Table 3‑5 : Print GUI Configuration 29](#_Toc382756422)

[Table 3‑6 : Print Job Configuration GUI Tests 29](#_Toc382756423)

[Table 3‑7 : Status GUI Tests 29](#_Toc382756424)

[Table 3‑8 : Import Controller Tests 30](#_Toc382756425)

[Table 3‑9 : Material Controller Tests 30](#_Toc382756426)

[Table 3‑10 : Printer Configuration Controller Tests 31](#_Toc382756427)

[Table 3‑11 : Print Configuration Controller 32](#_Toc382756428)

[Table 3‑12 : Extruder Controller Tests 32](#_Toc382756429)

[Table 3‑13 : Print Job Controller Tests 33](#_Toc382756430)

[Table 3‑14 : Status Controller 33](#_Toc382756431)

[Table 3‑15 : Command Structure Tests 34](#_Toc382756432)

[Table 3‑16: Subsection Module Tests 34](#_Toc382756433)

[Table 3‑17 : File Translation Module Tests 34](#_Toc382756434)

[Table 3‑18 : Slicing Engine Wrapper Tests 34](#_Toc382756435)

[Table 3‑19 : Parser Module Tests 35](#_Toc382756436)

[Table 3‑20 : Unification Module Tests 35](#_Toc382756437)

[Table 3‑21 : GUI Subsystem Tests 37](#_Toc382756438)

[Table 3‑22 : Controller Subsystem Tests 37](#_Toc382756439)

[Table 3‑23 : Database Subsystem Tests 38](#_Toc382756440)

[Table 3‑24 : Normalization Subsystem Tests 38](#_Toc382756441)

[Table 3‑25 : G-Code Preparation Subsystem Tests 39](#_Toc382756442)

[Table 3‑26 : G-Code Preparation Subsystem Tests 39](#_Toc382756443)

[Table 3‑27 : Integration Tests 40](#_Toc382756444)

[Table 3‑28 : Validation Tests 41](#_Toc382756445)

[Table 10‑1 : Test Schedule 54](#_Toc382756446)

[Table 11‑1 : Approvals 55](#_Toc382756447)

List of Figures

Figure # Title Page #

[Figure 2‑1: Architecture Dateflow Drawing 15](#_Toc382756402)

[Figure 2‑2: Detail Design Drawing 20](#_Toc382756403)

[Figure 3‑1: Design Decomposition 27](#_Toc382756404)

# Introduction

## Purpose

The 3-D Printer Fabrication System will provide an interface for converting standard stereo lithography or STL files into realized items. The system will use a simple graphical user interface to select the files and materials to be used in the production of the 3D model. When a file is loaded, the system will translate the STL file in to a series of layers based on the granularity of the materials to be used in the final build. The system will then use the layers to produce a series of paths for the print head to traverse to deposit the correct material to the specified location. From this series of steps, an instruction set will be produced for the printer to execute each path for every layer and material. Finally, the instructions will be issued to the printer and it will execute the commands producing the designed object that was described by the original STL files.

## Scope

The scope of the 3-D Printer Fabrication System is to develop software that will produce suitable machine code for a 3-D printer head that is capable of depositing multiple materials within a single print run. The system will present the user with an interface that will allow them to specify which STL files are to be loaded and specify the material properties of the respective STL files. The system will then use this information to process the geometry such that a suitable set of G-Codes can be issued to the device. The system will also provide a method for streaming the information to the printer control hardware via a serial interface. The system is intended to be used by 3-D printer operators, CNC operators, Dr. Shiakolas, and other experienced operators in the research field. The system is not intended for the consumer market.

# References

Throughout this System Test Plan we will refer to other documents created in the design process of the 3-D Printer Fabrication System. For completeness key features of these documents will be represented here for testing considerations. The documents referenced will be the SRS, ADS, and DDS.

## System Requirements Specification

The System Requirements Specification shows the analysis and gathering of customer, packaging, performance, performance, maintenance, and other requirements. The requirements for each set are detailed below. To improve clarity we will only include requirements that are going to be implemented in this project.

### Customer Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | Requirement Name | Description | Priority |
| 3.1 | STL File Input | The system shall provide a way for the user to select an STL file and then input that STL file into the system for processing | 1 – Critical |
| 3.2 | Graphical User Interface | The system shall provide a graphical user interface from which the user can import 3D models and initiate print operations. The GUI must be both intuitive and responsive. | 3 - Moderate |
| 3.3 | Generate Machine Instructions | The system shall generate instructions needed by the printing hardware in order to print a given 3D object. | 1 – Critical |
| 3.4 | Issue Machine Instructions | The system shall issue generated machine instructions from the software component to the printing hardware component. | 1 – Critical |
| 3.5 | Monitor Temperature | The system shall monitor input from heat sensors attached to the printing hardware. The temperature of each extruder’s nozzle must be monitored at all times to ensure that material is extruded at the proper temperature. | 1 – Critical |
| 3.6 | Monitor Position | The system shall monitor the position of the printing head at all times during operation. The system must be aware of the position of the printing head in order to adhere to a predefined printing path. | 1 – Critical |
| 3.7 | Adhere to Material Constraints | The system shall adhere to the material constraints that limit the movement speed, extrusion rate, and nozzle temperature. Different materials have different properties that the system must account for in order to produce a properly printed object. | 1 - Critical |
| 3.8 | Identify Materials | The system shall provide a method for the user to select the material for each discrete part that is being used for printing the 3D object. | 1 - Critical |
| 3.9 | Identify Shapes | The system shall identify the shape of the object being printed by dividing it into smaller shapes for each individual material used. | 1 - Critical |
| 3.10 | Determine Shape of Support Material Structure | The system shall determine the shape that the support material needs to be for stabilizing the 3D object as it is being printed. Without the support, the object could collapse during printing. | 1 - Critical |
| 3.11 | Create Printing Path | The system shall determine a route that the printing head must follow as it prints. | 1 - Critical |
| 3.12 | Database Interface | The system shall have an interface that allows the user to view what material is already stored in the database and enter new information for material not already stored. | 1 - Critical |
| 3.13 | Store & Load Material Records | The system shall be able to load the material records stored in the materials database in order to control the temperature, movement speed, and flow speed of the nozzle at the correct setting. | 1 - Critical |
| 3.14 | Slice Geometry into Thickness Levels | The system shall be able to process geometry in such a way as to generate sub-models of appropriate and customizable thickness such that the 3D printer can print each layer of the given thickness. | 1 - Critical |
| 3.15 | Monitor Flow Sensors | The system shall monitor nozzle flow sensors and be able to maintain and adjust accordingly if the sensor begins to read out of bounds. | 1 - Critical |
| 3.17 | Allow for UV Head Polymerization | The head shall be able to use UV light to cure or dry the extruded material. The system shall accommodate the use of UV to be turned on and off such that the material can be cured. | 1 - Critical |

Table 2‑1: Customer Requirements

### Packaging Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | Requirement Name | Description | Priority |
| 4.1 | Software Installer | The host software shall be delivered as an executable installer via USB flash memory and Compact Disc. | 1 - Critical |
| 4.2 | Host Software to Printer Connection | The host software shall be connected to the printing hardware using a DE-9, DB-25, or Universal Serial Bus cable. | 1 - Critical |
| 4.3 | User Manual | The system shall be delivered with a user manual. The user manual will include detailed instructions on how to operate the host software and how to properly connect the host software to the printer. | 3 - Moderate |

Table 2‑2 : Packaging Requirements

### Performance Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | Requirement Name | Description | Priority |
| 5.1 | Startup Time | The host software shall start in one minute or less. | 4 - Low |
| 5.2 | STL Import Time | The host software shall import STL files in one minute or less. | 4 - Low |
| 5.3 | Object Processing Time | The host software shall perform object processing and machine instruction generation in five minute or less. | 4 - Low |
| 5.4 | GUI Responsiveness | The graphical components of the user interface shall be responsive to user interaction. | 3 - Moderate |
| 5.5 | Real Time Sensor Monitoring | The system shall monitor data from sensors in real time during operation. The sensor data must be monitored in real time to ensure proper printer functionality as well as enforce safety systems. | 2 - High |

Table 2‑3 : Performance Requirements

### Safety Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | Requirement Name | Description | Priority |
| 6.1 | Temperature Cutoff Threshold | The system shall include a temperature cutoff threshold for the printer head. If the temperature of the printer head reaches the cutoff temperature, the system will abort the operation and shut off the heating device. | 1 - Critical |
| 6.2 | Printing Area Restrictions | The system shall only extrude material within a configured area. Material extruded by the printer will be at a high temperature and may cause harm to the printer’s surroundings; therefore it is important to ensure that the material is only extruded in a specified safe area. | 1 - Critical |

Table 2‑4 : Safety Requirements

### Maintenance and Support Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | Requirement Name | Description | Priority |
| 7.1 | Host Software Manual | A manual that details the operation of the host software shall be provided. Here, “host software” is that software which is run on the workstation that generates machine instructions for the printing hardware. The manual must detail common troubleshooting issues as well as provide basic usage instructions. | 1 - Critical |
| 7.2 | Source Code Documentation | The source code developed by the software team shall be well documented with comments explaining the functionality of all modules and any non-obvious code. This documentation is intended to support any future development on the system. | 2 - High |
| 7.3 | Source Code Availability | The source code developed by the software team shall be freely available to developers and the public. The source code will be hosted on a public repository. | 2 - High |

Table 2‑5 : Maintenance and Support Requirements

### Other Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | Requirement Name | Description | Priority |
| 8.1 | Material Database | The system shall have a database that holds information about how the material is printed. For each material, the database must hold the diameter of the material filament, the temperature the filament must be extruded at, the extrusion speed, and weather a secondary head option such as UV curing light is required. | 1 - Critical |
| 8.2 | Abstract Hardware Interface | The system shall allow for multiple different printers and multiple different heads to be used with minimal software change. | 1 - Critical |

|  |  |  |  |
| --- | --- | --- | --- |
| 8.3 | Modular and Scalable Design | The software shall be developed using proven design principles to ensure that it can be scaled and maintained by future development teams. This design will allow the future requirements such as the Door Sensor to be implemented without difficulty. | 2 - High |

Table 2‑6 : Other Requirements

## Architecture Design Specification

This section details the architecture of the 3-D Printer Fabrication System. The system consists of seven layers that have predominantly a one way flow of information. Our architecture was designed with modularity, configurability, extendibility, and portability as the guiding principles.



Figure 2‑1: Architecture Dateflow Drawing

### Interface Layer

The Interface Layer’s purpose is to collect user information and serve it to the processing layers. There are three subsystems providing data to lower levels. The GUI subsystem provides the user the ability to interact with the system and provides the ability to import files, create material profiles, set general configuration, and print specific configuration. The database subsystem provides a persistence framework for the other subsystems in this layer for storage and retrieval of configuration and material information. The controller subsystem decouples the user interface from the program logic. The controller provides the GUI with the information to present and takes in the user actions allowing the initiation of a print run, then packing the collected configuration, object files, and material data in to a print request object for the preprocessing layer.

### Pre Processing Layer

The purpose of the preprocessing layer is to provide an abstract interface between the interface and the processing layer. Its primary duty is to translate and repackage the print request object in to the format that the processing layer needs. This will allow for modular construction. The primary feature is conversion from STL to AMF. This is just one form of object definition to another. By using the interface abstraction the processing layer and interface could change independently with just a new concrete interface if the required file formats changed. The preprocessing layer packages the configuration, object definition, and material data into the correct format for the processing layer.

### Processing Layer

The purpose of the processing layer is to take the formatted package and translate it into G-codes. The slicing engine is the only subsystem in this layer. The slicing engine needs to be able to be replaced based on the growing needs of future development. The processing layer will therefore have an abstract interface and Slic3r will be the initial concrete engine that will be prepared. There are numerous engines available, so the conceptual design needs to allow for other engines to be implemented with minimal changes. This layer design will allow configuration data to select the correct slicing engine based on the package received from the preprocessor. The processing layer will pass a package of G-codes and configuration.

### Post Processing Layer

The purpose of the post processing layer is to prepare the G-Codes for the printer. The post processing layer has one subsystem that provides an abstract interface between the processing layer and the physical layer. This will allow the processor and physical implementations to change with menial code change. The configuration and G-codes will be read in the G-Code preparation subsystem and it will do final G-Code modification or addition for the current printer configuration. The post processing layer then packages the G-Code and the configuration then sends it to the physical layer.

### Printer Control Layer

The purpose of the printer control layer is to control the command flow to the printer and translate the commands to a byte stream to be sent to the printer. There is one subsystem in this layer: the print state controller subsystem. The print state controller receives the state of the printer from the printer feedback layer, the G-Code, and the configuration from the post processing layer as well as pause/resume request from the interface layer. The print state controller uses the state information to control the flow of G-Codes to the packet preparation subsystem. The print state controller then sends the G-Code stream to the communications layer to be serialized for transmission to the printer itself

### Printer Feedback Layer

The purpose of the printer feedback is to provide printer statistics to the physical and interface layers. The state monitoring subsystem continuously polls the printer for state information. This information is packaged for the interface layer to display the print status. The state monitoring subsystem also sends interrupts to the printer control layer with error state conditions so that the printer state controller can correct the state and prevent bad prints.

### Communications Layer

The purpose of the communications layer is to provide a serial connection to the printer for the printer control layer and printer feedback layer. The communications subsystem will provide serialization and deserialization interfaces such that the printer feedback and printer control layers will send and receive standard object. The communications layer will also be the single point of contact for all information flowing to or from the printer.

### Dataflow Definitions

The table below describes the data flows between layers and subsystems in Figure 2-1.

|  |  |  |
| --- | --- | --- |
| Layer | Data Element | Description |
| Outside Inputs | OI1. User Actions | User clicks and data entry. These will be in button clicks for printing, saving printer or material configuration, navigation, and initiating print runs. Data entry forms to load specific information for the printer such as dimensions of print bed, number of heads. Material configuration such as material flow rate, thickness of filament, temperature bounds, motor speeds. |
| OI2. Binary File | This represents the object file loaded in memory. This file could be an AMF, STL, or other standard object file. |
| OI3. Database response | The database will return SQL data sets based on the SQL requests received from the database subsystem. |
| 3D1. State Information | This is a stream of state information from the printer to be translated and checked against the configuration. |
| User Interface | UI1. Actions | The GUI will activate the controller and pass the actions such as button presses and the supporting data for processing. |
| UI2. GUI Data Updates | The controller updates the GUI objects with current data and status in the form of repaint requests and data objects. |
| UI3. Data Objects | The database will return the requested data objects from the controller. These can be material, print configuration, or printer configuration objects. |
| UI4. Database request | The controller will request data objects from the database. These requests will be in the form of load or save objects such as load printer configuration or save updated material object. |
| UI5. Selection Statements | Database statements to load or store specific material or configuration data from or to the server. |
| UI6. The Packet | The packaged object definitions, material, and configuration. This is all the information the lower layers will need to process and print the object. |
| UI7. Pause/Resume Request | This is a signal to the state controller to pause printing. |
| Preprocessing | PR1. Normalized Object and Configuration | The package object with the correct file format for the processor, material, and configuration. |
| Processing | PO1. Object package | This carries the sliced object G-Code and printer configuration. |
| Post Processing | PP1. Specific G-Code | This carries the printer specific G-Codes and configuration. |
| Printer Control | PL1. G-Code | This is a stream of G-Codes to be sent to the printer. |
| Communications | CL1. Bits on a wire | This is the G-Code stream packetized and transmitted to the printer. |
| CL2. Printer State | This is the unmodified printer state information to be interpreted and forwarded to other subsystems |
| Printer Feedback | PF1. Printer State | This is an object that contains specific printer state conditions to be acted on. |
| PF2. State Information | This is a package of printer state information to be displayed on the printer status subsystem. |

Table 2‑7: Architecture Dataflow Definitions

### Layer Requirements Traceability

The following table details the mapping of critical requirements to specific layers. Ensuring all requirements are covered by at least one layer and providing a map of what functionality needs to be tested in each layer.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number | Title | User Interface Layer | Preprocessing Layer | Processing Layer | Post Processing Layer | Printer Control Layer | Comm. Layer | Printer | Printer Feedback Layer |
| 3.1 | STL File Input | X |  |  |  |  |  |  |  |
| 3.2 | Graphical User Interface (GUI) | X |  |  |  |  |  |  |  |
| 3.3 | Generate Machine Instructions |  |  | X | X | X |  |  |  |
| 3.4 | Issue Machine Instructions |  |  |  |  |  | X |  |  |
| 3.5 | Monitor Temperature | X |  |  |  | X | X | X | X |
| 3.6 | Monitor Position |  |  |  |  | X | X | X | X |
| 3.7 | Adhere to Material Constraints | X |  | X | X | X |  | X | X |
| 3.8 | Identify Materials | X | X | X | X |  |  |  |  |
| 3.9 | Identify Shapes |  |  | X |  |  |  |  |  |
| 3.1 | Determine Shape of Material Support Structure |  |  | X |  |  |  |  |  |
| 3.11 | Create Printing Path |  |  | X | X |  |  |  |  |
| 3.12 | Database Interface | X |  |  |  |  |  |  |  |
| 3.13 | Load & Store Materials | X |  |  |  |  |  |  |  |
| 3.14 | Slice Geometry into Thickness Levels |  |  | X |  |  |  |  |  |
| 3.15 | Monitor Flow Sensors |  |  |  |  | X |  | X | X |
| 3.17 | Allow for UV Head Polymerization | X |  | X | X | X |  | X |  |
| 8.1 | Material Database | X |  |  |  |  |  |  |  |
| 8.2 | Abstract Hardware Interface | X |  |  |  | X |  | X | X |
| 8.3 | Modular and Scalable Design | X | X | X | X | X |  | X | X |

Table 2‑8 : Layer Requirements Traceability

## Detail Design Specification

### Prologue

This section provides information from the Detail Design Specification of the 3-D Printer Fabrication System. The DDS takes the abstract decomposition of the ADS and decomposes the subsystems into discrete modules. The DDS also provides detailed module descriptions and pseudo code for the functions being described.



Figure 2‑2: Detail Design Drawing

### Data Flow Definitions

The table below describes the data flows between subsystems and modules in Figure 2-2.

|  |  |  |
| --- | --- | --- |
| Layer | Data Flow ID | Data |
| Outside Inputs | OI1 | Object file of the STL file to be printed |
| OI2 | Printer configuration data entry values |
| OI3 | Material configuration data entry values |
| OI4 | Print configuration data entry values |
| OI5 | Print selections and button press |
| OI6 | Pause, resume, and stop button Presses |
| OI7 | Disk reads of XML and Directory structure |
| OI8 | Printer state information |
| OI9 | OS Driver information |
| OI10 | Extruder configuration data entry values |
| User Interface Layer | UI1 | Import file name |
| UI2 | Success state of import |
| UI3 | Loaded printer configuration |
| UI4 | Save printer configuration request |
| UI5 | Save material configuration request |
| UI6 | Loaded material configuration |
| UI7 | Loaded print configuration |
| UI8 | Save print configuration request |
| UI9 | Loaded configuration data |
| UI10 | Run print job |
| UI11 | Pause, resume, and stop button Presses |
| UI12 | Printer state information |
| UI13 | File Configuration object |
| UI14 | Printer configuration object |
| UI15 | Material configuration object |
| UI16 | Print Configuration object |
| UI17 | All requested configuration objects |
| UI18 | Object save/load requests |
| UI19 | Object save/load results |
| UI20 | XML File writes to disk |
| UI21 | Save print job configuration |
| UI22 | Print package object |
| UI23 | Print, Pause, Resume request |
| UI24 | Loaded Extruder Configuration Data |
| UI25 | Save Extruder Configuration Request |
| UI26 | Extruder Configuration Object |
| Preprocessing Layer | PR1 | Print package object |
| PR2 | Print package object |
| Processing Layer | PO1 | Print package object |
| Post-Processing Layer | PP1 | Print package object |
| PP2 | Print package object |
| Printer Control Layer | PL1 | G-Code |
| Communications Layer | CL1 | Serialized G-Codes |
| CL2 | Serialized printer state |
| CL3 | Command stream |
| Printer Feedback Layer | PF1 | Printer State Object |
| PF2 | Printer State Object |

Table 2‑9 : Data Flow Definitions

### Module Requirements Traceability

The tables below map the requirements defined in the SRS to the modules described in the DDS.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number | Title | Interface Layer | Import GUI | Printer Configuration GUI | Material GUI | Print Configuration GUI | Print Job GUI | Status GUI | Extruder Configuration GUI | Import Controller | Printer Configuration Controller | Material Controller | Print Configuration Controller | Print Job Controller | Extruder Configuration Controller | Status Controller | Persistence Framework | Command Structure |
| 3.1 | STL File Input | X | X |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| 3.2 | Graphical User Interface (GUI) | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| 3.3 | Generate Machine Instructions |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.4 | Issue Machine Instructions |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.5 | Monitor Temperature | X |  |  |  |  |  | X |  |  |  |  |  |  |  | X |  |  |
| 3.6 | Monitor Position |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.7 | Adhere to Material Constraints | X |  |  | X | X | X |  | X |  | X | X | X |  |  |  |  |  |
| 3.8 | Identify Materials | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.9 | Identify Shapes |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1 | Determine Shape of Material Support Structure |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.11 | Create Printing Path |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.12 | Database Interface | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |
| 3.13 | Load & Store Materials | X |  |  | X |  |  |  |  |  |  | X |  |  |  |  | X | X |
| 3.14 | Slice Geometry into Thickness Levels |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.15 | Monitor Flow Sensors |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.17 | Allow for UV Head Polymerization | X |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  |
| 8.1 | Material Database | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |
| 8.2 | Abstract Hardware Interface | X |  | X |  |  | X |  |  |  | X |  |  | X |  |  |  |  |
| 8.3 | Modular and Scalable Design | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

Table 2‑10 : UI Layer Requirements Traceability

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Title | Preprocessing Layer Object Subsection | Preprocessing Layer Object File Translation | Processing Layer Slicing Engine Wrapper | Post Processing Layer Parser | Post Processing Layer Unification |
| 3.1 | STL File Input |  |  |  |  |  |
| 3.2 | Graphical User Interface (GUI) |  |  |  |  |  |
| 3.3 | Generate Machine Instructions |  |  | X | X | X |
| 3.4 | Issue Machine Instructions |  |  |  |  |  |
| 3.5 | Monitor Temperature |  |  |  |  |  |
| 3.6 | Monitor Position |  |  |  |  |  |
| 3.7 | Adhere to Material Constraints |  |  | X | X | X |
| 3.8 | Identify Materials | X | X | X | X | X |
| 3.9 | Identify Shapes |  |  | X |  |  |
| 3.1 | Determine Shape of Material Support Structure |  |  | X |  |  |
| 3.11 | Create Printing Path |  |  | X | X | X |
| 3.12 | Database Interface |  |  |  |  |  |
| 3.13 | Load & Store Materials |  |  |  |  |  |
| 3.14 | Slice Geometry into Thickness Levels |  |  | X |  |  |
| 3.15 | Monitor Flow Sensors |  |  |  |  |  |
| 3.17 | Allow for UV Head Polymerization |  |  | X | X | X |
| 8.1 | Material Database |  |  |  |  |  |
| 8.2 | Abstract Hardware Interface |  |  |  |  |  |
| 8.3 | Modular and Scalable Design | X | X | X | X | X |

Table 2‑11 : Processing Layers Requirements Traceability

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number | Title | Printer Control Layer Printer State Controller | Communications Serialization | Communications RxTx | Communications Deserialization | Printer | Printer Feedback Layer  Dispatch |
| 3.1 | STL File Input |  |  |  |  |  |  |
| 3.2 | Graphical User Interface (GUI) |  |  |  |  |  |  |
| 3.3 | Generate Machine Instructions | X |  |  |  |  |  |
| 3.4 | Issue Machine Instructions |  | X | X | X |  |  |
| 3.5 | Monitor Temperature | X | X | X | X | X | X |
| 3.6 | Monitor Position | X | X | X | X | X | X |
| 3.7 | Adhere to Material Constraints | X | X | X | X | X | X |
| 3.8 | Identify Materials |  |  |  |  |  |  |
| 3.9 | Identify Shapes |  |  |  |  |  |  |
| 3.1 | Determine Shape of Material Support Structure |  |  |  |  |  |  |
| 3.11 | Create Printing Path |  |  |  |  |  |  |
| 3.12 | Database Interface |  |  |  |  |  |  |
| 3.13 | Load & Store Materials |  |  |  |  |  |  |
| 3.14 | Slice Geometry into Thickness Levels |  |  |  |  |  |  |
| 3.15 | Monitor Flow Sensors | X |  | X | X | X | X |
| 3.17 | Allow for UV Head Polymerization | X |  |  |  | X |  |
| 8.1 | Material Database |  |  |  |  |  |  |
| 8.2 | Abstract Hardware Interface | X |  |  |  | X | X |
| 8.3 | Modular and Scalable Design | X | X | X | X | X | X |

Table 2‑12: Printer State, Comm., and Printer Feedback Requirements Traceability

# Test Items

## Product Version

We will be following a staged development approach during the implementation stage of this project. This means we plan to implement a different set of functionality during each stage. The goal is to provide a completed tested portion of the application during each stage. Below are the stages to be developed with the subsystems and modules to be developed in the stages. Stage one will begin upon completion of this Test Plan document.

### Stage one

* User Interface Layer
  + Database subsystem
    - Persistence Framework
    - Command Structure
* Preprocessing Layer
  + Normalization Subsystem
    - Object Subsection Module
    - Object Translation Module
* Processing Layer
  + Slicing Engine
    - Slicing Engine Wrapper
* Post Processing Layer
  + G-Code Preparation
    - Parser Module
    - Unification Module
* Communications Layer
  + Communications Subsystem
    - Serialization Module
    - TxRx Module
    - Deserialization Module

### Stage Two

* User Interface Layer
  + GUI Subsystem
    - Print Job GUI Module
  + Controller Subsystem
    - Print Job Controller
* Printer Control Layer
  + Printer State Controller
    - Printer State Controller

### Stage Three

* User Interface Layer
  + GUI Subsystem
    - Import GUI Module
    - Printer Configuration GUI Module
    - Material GUI Module
    - Print Configuration GUI Module
    - Extruder GUI Module
    - Status GUI Module
  + Controller Subsystem
    - Import Controller
    - Printer Configuration Controller
    - Material Controller
    - Print Configuration Controller
    - Extruder Controller
* Printer Feedback Layer
  + State Monitoring
    - Dispatch Module

## Design Decomposition



Figure 3‑1: Design Decomposition

## Unit Tests

### Import GUI

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| IG1 | * Button Press * String File Name | * Returns message of success to user. * Returns message of failure to user. * Returns message of failure to user. | * Choose a valid STL file and click Import. * Choose an invalid STL file and click Import. * Choose an invalid file type and click “Import”. |
| IG2 | * Button Press * String File Name | * Returns message of success to user. | * Choose an STL file and click “Delete”. |

Table 3‑1 : Import GUI Tests

### Material GUI Tests

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| GM1 | * Button Press * String File Name | * Updates display | * Manually click “New”. |
| GM2 | * Button Press * String File Name * Material Data | * Returns message of success to user. * Returns message of failure to user. | * Insert data of valid type and click “Save”. * Insert data of invalid type and click “Save”. |
| GM3 | * Button Press * String File Name | * Returns message of success to user. | * Choose a material file and click “Delete”. |

Table 3‑2 : Material GUI Tests

### Printer Configuration GUI

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| CO1 | * Button Press * String File Name | * Updates display | * Manually click “New”. |
| CO2 | * Button Press * String File Name | * Returns message of success to user. * Returns message of failure to user. | * Insert data of valid type and click “Save”. * Insert data of invalid type and click “Save”. |
| CO3 | * Button Press * String File Name | * Returns message of success to user. | * Choose a printer config file and click “Delete”. |

Table 3‑3 : Printer Configuration GUI Tests

### Extruder Configuration GUI

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| GE1 | * Button Press * String File Name | * Updates display | * Manually click “New”. |
| GE2 | * Button Press * String File Name | * Returns message of success to user. * Returns message of failure to user. | * Insert data of valid type and click “Save”. * Insert data of invalid type and click “Save”. |
| GE3 | * Button Press * String File Name | * Returns message of success to user. | * Choose an extruder config file and click “Delete”. |

Table 3‑4 : Extruder Configuration GUI Tests

### Print Configuration GUI

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| PC1 | * Button Press * String File Name | * Updates display | * Manually click “New”. |
| PC2 | * Button Press * String File Name | * Returns message of success to user. * Returns message of failure to user. | * Insert data of valid type and click “Save”. * Insert data of invalid type and click “Save”. |
| PC3 | * Button Press * String File Name | * Returns message of success to user. | * Choose a print config file and click “Delete”. |

Table 3‑5 : Print GUI Configuration

### Print Job GUI

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| GP1 | * Button Press * String File Name | * Updates display | * Choose a print job configuration. |
| GP2 | * Button Press | * Updates display | * Click “New Subsection”. |
| GP3 | * Button Press * String File Name * Print Job Config Data | * Returns message of success to user. * Returns message of failure to user. | * Insert data of valid type and click “Save”. * Insert data of invalid type and click “Save”. |
| GP3 | * Button Press * String File Name * Print Job Config Data | * Returns message of success to user. | * Choose a print job file and click “Delete”. |

Table 3‑6 : Print Job Configuration GUI Tests

### Status GUI

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| GS1 | * Button Press | * Updates display * Updates display | * Click “pause” button * Click “resume” button |
| GS2 | * Button Press | * Updates display | * Click “Cancel” button. |

Table 3‑7 : Status GUI Tests

### Import Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| IC1 | * File Name of STL | * Boolean Success state = success * Boolean Success state = failure * Boolean Success state = failure | * Pass a valid STL file to import() * Pass an invalid STL file to import() * Pass an invalid file type to import() |
| IC2 | * File Name of STL | * Boolean Success State = success * Boolean Success State = failure | * Pass a valid STL file to delete() * Pass an invalid file to delete() |

Table 3‑8 : Import Controller Tests

### Material Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| MC1 | * String File Name * Material Config Object | * Boolean Success State = success * Boolean Success State = failure | * Pass a Material config object with values within boundary to save() * Pass a Material config object with values exceeding boundary to save() |
| MC2 | * String File Name | * Material Config Object * Error | * Pass valid string name to load() * Pass invalid string name to load() |
| MC3 | * String File Name | * Boolean Success State = success * Boolean Success State = failure | * Pass valid string name to delete() * Pass invalid string name to delete() |
| MC4 | * None | * Array List<String> of file names | * Call loadAvailableMaterials() |

Table 3‑9 : Material Controller Tests

### Printer Configuration Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| PR1 | * String File Name * Printer Config Object | * Boolean Success State = success * Boolean Success State = failure | * Pass a Printer config object with values within boundary to save() * Pass a Printer config object with values exceeding boundary to save() |
| PR2 | * String File Name | * Printer Config Object * Error | * Pass valid string name to load() * Pass invalid string name to load() |
| PR3 | * String File Name | * Boolean Success State = success * Boolean Success State = failure | * Pass valid string name to delete() * Pass invalid string name to delete() |
| PR4 | * None | * Array List<String> of file names | * Call the function to loadAvailablePrinters() |
| PR5 | * None | * Array List<String> of port descriptors | * Call the function to enumeratePorts() |

Table 3‑10 : Printer Configuration Controller Tests

### Print Configuration Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| PT1 | * String File Name * Print Config Object | * Boolean Success State = success * Boolean Success State = failure | * Pass a Print Config object with values within boundary tosave() * Pass a Printer Config object with values exceeding boundary to save() |
| PT2 | * String File Name | * Print Config Object * Error | * Pass valid string name to load() * Pass invalid string name to load() |
| PT3 | * String File Name | * Boolean Success State = success * Boolean Success State = failure | * Pass valid string name to delete() * Pass invalid string name to delete() |
| PT4 | * None | * Array List<String> of file names | * Call loadAvailablePrintConfigs() |

Table 3‑11 : Print Configuration Controller

### Extruder Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| EC1 | * String File Name * Extruder Config Object | * Boolean Success State = success * Boolean Success State = failure | * Pass a Extruder Config object with values within boundary to save() * Pass a Extruder Config object with values exceeding boundary to save() |
| EC2 | * String File Name | * Extruder Config Object * Error | * Pass valid string name to load() * Pass invalid string name to load() |
| EC3 | * String File Name | * Boolean Success State = success * Boolean Success State = failure | * Pass valid string name to delete() * Pass invalid string name to delete() |
| EC4 | * None | * Array List<String> of file names | * Call loadAvailableExtruders() |

Table 3‑12 : Extruder Controller Tests

### Print Job Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| PJ1 | * String File Name * Print Job Config Object | * Boolean Success State = success * Boolean Success State = failure | * Pass a Print Job Config object with values within boundary to save() * Pass a Print Job Config object with values exceeding boundary to save() |
| PJ2 | * String File Name | * Print Job Config Object * Error | * Pass valid string name to load() * Pass invalid string name load() |
| PJ3 | * String File Name | * Boolean Success State = success * Boolean Success State = failure | * Pass valid string name to delete() * Pass invalid string name to delete() |
| PJ4 | * None | * Array List<String> of file names | * Call loadAvailablePrintJobs() |
| PJ5 | * Print Job Config Object | * Open Status GUI * Error | * Pass a correct Print Job Config Object with values within boundary to startPrint() * Pass a Print Job Config Object with values outside boundary to startPrint() |

Table 3‑13 : Print Job Controller Tests

### Status Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| SC1 | * Float or double * Vec3 | * Returns variables * Ignore input. | * Pass valid input to update() * Pass invalid input to update() |
| SC2 | * None | * BooleanStatus = success * BooleanStatus = success | * Call pause() * Call resume |
| SC3 | * None | * BooleanStatus = success | * Call cancel() |

Table 3‑14 : Status Controller

### Command Structure

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| CS1 | * Function Call | * Multiple Array Lists of File Names | Execute get Configurations and ensure all files are listed for each type of object in storage |
| CS2 | * Function Call * String | * Fully formed Printer Configuration Object | Call get Printer Config function with a file name. |
| CS3 | * Function Call * String | * Fully formed Extruder Configuration Object | Call get Extruder Config function with a file name. |
| CS4 | * Function Call * String | * Fully formed Material Configuration Object | Call get Material Config function with a file name. |
| CS5 | * Function Call * String | * Fully formed Print Job Configuration Object | Call get Print Job Config function with a file name. |
| CS6 | * Function Call * String | * Fully formed Print Configuration Object | Call get Print Config function with a file name. |
| CS7 | * Function Call * Printer Configuration Object | * Boolean True | Call save Printer Configuration function with a file name. |
| CS8 | * Function Call * Extruder Configuration Object | * Boolean True | Call save Extruder Configuration function with a file name. |
| CS9 | * Function Call * Material Configuration Object | * Boolean True | Call save Material Configuration function with a file name. |
| CS10 | * Function Call * Print Job Configuration Object | * Boolean True | Call save Print Job Configuration function with a file name. |
| CS11 | * Function Call * String | * Boolean True | Call delete Configuration function with a file name. |

Table 3‑15 : Command Structure Tests

### Subsection Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| SM1 | * Start Z * End Z * Parent STL for each material * Per subsection. | * STL files for each material in each subsection all bound by their specified Start Z and End Z. | Manually build a PrintJobConfiguration object with the described input and pass it to the Subsection Module. Observe the output STL files for correctness. |

Table 3‑16: Subsection Module Tests

### File Translation Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| FT1 | * Subsection STL files for each subsection * Material Configurations for each material in each subsection | * AMF file for each subsection that describes the combination of all subsection STL files as a combination of volumes, each volume mapped to its correct material. | Manually build a PrintJobConfiguration object with the described input and pass it to the File Translation Module. Observe the output AMF file for correctness. |

Table 3‑17 : File Translation Module Tests

### Slicing Engine Wrapper

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| SE1 | * All data items described in the Detailed Design Specification section 6.1.1.4 | * G-Code files for each subsection. | Manually build a PrintJobConfiguration object with the described input and pass it to the Slicing Engine Wrapper Module. Observe the output G-Code files for correctness. |

Table 3‑18 : Slicing Engine Wrapper Tests

### Parser Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| PM1 | * The printer G-Code flavor * Printer custom start G-Code * Printer custom end G-Code, and G-Code files for each subsection. | * G-Code files for each subsection with only G-Code commands specific to the specified printer G-Code flavor, the printer customer start G-Code on the bottom most layer, and the printer custom end G-Code at the end of the top most layer. | * Manually build a PrintJobConfiguration object with the described input and pass it to the Parser Module. Observe the output G-Code files for correctness. |

Table 3‑19 : Parser Module Tests

### Unification Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| UM1 | * G-Code files for each subsection. | * Finalized G-Code file with all subsection G-Code in the order from the bottom most subsection to the top most subsection. | Manually build a PrintJobConfiguration object with the described input and pass it to the Unification Module. Observe the output G-Code file for correctness. |

Table 3‑20 : Unification Module Tests

### Printer State Controller Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| PS1 | * G-Codes * User Interface Status (Cancel) | * The Printer State Controller shall halt the sending of G-Codes to the Serialization Module. | Manually build PrintJobConfiguration object with test G-Codes. Trigger cancel event to ensure that no further G-Codes are sent to Serialization Module. There is no return state. Therefore, the print job cannot be resumed. |
| PS2 | * G-Codes * Printer Feedback Status (Temperature and Position) | * The Printer State Controller shall consume temperature and position information from Printer Feedback Status and insert halt command into the G-Code buffer to stop the print when values are out of range. | Manually build PrintJobConfiguration object with test G-Codes. Manually build Printer State Object with out-of-bounds operating parameters. Ensure that proper halt G-Codes are inserted into the output G-Code buffer. |
| PS3 | * G-Codes and * User Interface Status (Pause/Resume) | * The Printer State Controller shall halt the sending of G-Codes to the Serialization Module until resume action is received. | Manually build PrintJobConfiguration object with test G-Codes. Trigger pause event to ensure that no further G-Codes are sent to Serialization Module. Trigger resume event to ensure that G-Codes resume transmission from prior state. |

Table 3‑7 : Printer State Controller Module Tests

### Dispatch Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| DM1 | * De-Serialized printer feedback information | * Populated PrinterFeedback object | Build de-serialized printer feedback object which represents the de-serialized printer feedback data. Verify that the PrinterFeedback Object is built correctly by the Dispatch Module. |

Table 3‑8 : Dispatch module Tests

### Serialization Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| SE1 | * Finalized G-Codes | * Serialized stream of G-Codes to be sent to the printer | Manually build simple G-Code set to be tested. Inspect and verify that Serialization Module serializes G-Codes without serialization errors. |

Table 3-9 : Serialization Module Tests

### De-Serialization Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| DS1 | * Serialized byte-stream of printer feedback information | * De-Serialized printer feedback object | Build test program that establishes connection to the printer and issues simple G-Codes. Connect test PC to printer and monitor feedback stream from printer. Inspect and verify that the de-serialized printer feedback object is built correctly. Inspection and verification can be accomplished with a Java debugger hooked into the test program. |

Table 3‑10 : De-Serialization Module Tests

### TxRx Module

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| TR1 | * PrintJobConfiguration object | * Successfully establish connection to printer firmware via virtual serial communications using the parameters specified within the PrintJobConfiguration object. | Manually build a PrintJobConfiguration object with connection information. The test PrintJobConfiguration object will be passed to a small test program that uses its information to establish connection to the printer firmware. |

|  |  |  |  |
| --- | --- | --- | --- |
| TR2 | * Serialized G-Code stream | * Successfully poll and lock printer G-Code buffer. Write serialized G-Code stream to printer firmware. | Using the test program that establishes a connection to the printer, we will poll the printer, lock the G-Code buffer and verify that the test G-Codes have been sent. This can be verified visually as the printer executes the commands and via the printer feedback stream. |
| TR3 | * Byte-stream of printer feedback information from printer | * Successfully populate printer feedback buffer and raise a “buffer ready” flag signifying that printer feedback data is ready. | Using the test program that establishes a connection to the printer, we will continually monitor the input stream for printer feedback. We will verify that the feedback data buffer has been populated, is recent, and raises the ready flag. This can be inspected and verified with a Java debugger hooked into the test program. |

Table 3‑11 : TxRx Module Tests

## Component Tests

### GUI Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| GS1 | * Button Presses * Text Fields | * Updates display with correct results * The menus switch correctly | * Test individual units * Tester navigates the menus by clicking the tabs. |

Table 3‑21 : GUI Subsystem Tests

### Controller Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| GC1 | String File Names  Object Files | * Makes the proper load and save requests with the correct info. | * Check how the subsystem handles correct and incorrect data. |

Table 3‑22 : Controller Subsystem Tests

### Database Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| PF1 | * Function Call | * Multiple Array Lists of File Names | Execute get Configurations and ensure all files are listed for each type of object in storage |
| PF2 | * Function Call * String | * Fully formed Printer Configuration Object | Call get Printer Config function with a file name. |
| PF3 | * Function Call * String | * Fully formed Extruder Configuration Object | Call get Extruder Config function with a file name. |
| PF4 | * Function Call * String | * Fully formed Material Configuration Object | Call get Material Config function with a file name. |
| PF5 | * Function Call * String | * Fully formed Print Job Configuration Object | Call get Print Job Config function with a file name. |
| PF6 | * Function Call * String | * Fully formed Print Configuration Object | Call get Print Config function with a file name. |
| PF7 | * Function Call * Printer Configuration Object | * Boolean True | Call save Printer Configuration function with a file name. |
| PF8 | * Function Call * Extruder Configuration Object | * Boolean True | Call save Extruder Configuration function with a file name. |
| PF9 | * Function Call * Material Configuration Object | * Boolean True | Call save Material Configuration function with a file name. |
| PF10 | * Function Call * Print Job Configuration Object | * Boolean True | Call save Print Job Configuration function with a file name. |
| PF11 | * Function Call * String | * Boolean True | Call delete Configuration function with a file name. |

Table 3‑23 : Database Subsystem Tests

### Normalization Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| NS1 | * Start Z * End Z for each subsection * Parent STL File for each material in each subsection. | * AMF file for each subsection that describes the combination of all subsection STL files as a combination of volumes, each volume mapped to its correct material. | Manually build a PrintJobConfiguration object with the described input and pass it to the Normalization Subsystem. Observe the output AMF file for correctness. |

Table 3‑24 : Normalization Subsystem Tests

### G-Code Preparation Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| GP1 | * The printer G-Code flavor * Printer custom start G-Code * Printer custom end G-Code * G-Code files for each subsection. | * Finalized G-Code file with all subsection G-Code in the order from the bottom most subsection to the top most subsection. | Manually build a PrintJobConfiguration object with the described input and pass it to the G-Code Preparation Subsystem. Observe the output G-Code file for correctness. |

Table 3‑25 : G-Code Preparation Subsystem Tests

### Communication Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| CM1 | * Finalized G-Codes * PrintJobConfiguration object * Serialized byte-stream of printer feedback information | * Connection will be established with the printer. G-Codes will be sent to the printer, and printer feedback information will be received from the printer. | Manually build a PrintJobConfiguration object with the printer connection parameters. Test G-Codes will be sent to the printer. We will verify that the G-Codes are sent correctly by observing the printer during the test print run and monitoring the printer feedback information. |

Table 3‑26 : G-Code Preparation Subsystem Tests

## Integration Tests

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test |
| IT1 | * Full Configuration * STL Files * Print Job Start | * Fully Formed Configuration object (UI Layer) * Subsection AMF Files (Preprocessing Layer) * Raw G-Code Files (Processing Layer) * Compiled G0Code (Post Processing Layer) * Stream of Codes (Printer Control Layer) * Stream of data (Communications Layer) * Printer Object (Printer) * Status Updates (Printer) * Status Control Updates (State Monitoring Layer) | This test will be a series of simple objects taken from STL to final printer object. At each layer divide the processing will be stopped using a debug mode so the objects can be inspected for completeness and correctness. |

Table 3‑27 : Integration Tests

## Validation Tests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Requirement Tested | | Expected Output/Action | | Test | | |
| VT1 | 3.1 STL File Input | | * The user will be able to see the file has been accepted and the file name will be displayed in the GUI. | | Use the GUI to import a STL File. | | |
| VT2 | | 3.2 Graphical User Interface  3.13 Database Interface  3.14 Store and Load Material Records  8.1 Material Database | | * Configuration data will be stored and retrieved | | The user will load the GUI and click on the view/edit database button. The user then will see and be able to edit stored values. Upon changing values the user will return to the main menu. Clicking on the view/edit button again will display the edited values. |
| VT3 | | 3.1 STL File Input  3.3 Generate Machine Instructions  3.4 Issue Machine Instructions  3.6 Monitor Temperature  3.7 Monitor Position  3.8 Identify Material Constraints  3.9 Identify Materials  3.10 Identify Shapes  3.11 Determine Shape of Support Material Structure  3.12 Create Printing Path  3.14 Store and Load Material Records  3.15 Slice Geometry into Thickness Levels  3.16 Monitor Flow Sensors  4.2 Host Software to Printer Connection | | * Printed Model | | The user will load an STL file. Set configuration Data and click print. The system will then print the correct shape and material. |
| VT4 | | 6.1 Temperature cutoff threshold  3.16 Monitor Flow Sensors | | * System stops printing if out of range | | During a print run a fan will be pointed at the head reducing its temperature to below specified material requirements and the printer will stop printing until the temperature is raised to the correct level. |

Table 3‑28 : Validation Tests

# Risks

This section details the risks that are associated with the test plan. For each risk identified the impact on the project, severity of the impact, and the strategy for dealing with the risk is detailed below.

## Hardware Unreliable

### Description

There is a chance that the 3-D printer being developed by the mechanical team will not reliably perform as expected.

### Impact

It may be difficult to identify whether errors in a print job were caused by the software or the hardware.

### Severity

Low

### Strategy

G-Codes that will be output to the printer will be examined to determine if they are correct. Incoming messages from the printer will be caught by separate software and examined.

## Third Party Software Unreliable

### Description

Most of the processing of this system relies on third party software that is assumed to work as expected. If the third party software is giving erroneous output, the tests in this test plan will not return accurate results.

### Impact

During third party software failure, tests in this test plan may not return accurate results.

### Severity

High

### Strategy

During any test where third party software will be performing work for the system, that third party software will be tested separately in addition to being tested within the system.

## Untested Input Combinations

### Description

There is an extremely large amount of different data configurations that can be input into this system and it is not possible to test every single combination of inputs.

### Impact

Some defects specific to certain combinations of input may never be discovered through the testing process.

### Severity

Medium

### Strategy

Boundary conditions will be tested for each input.

## Defect Fixes May Cause Other New Defects

### Description

Defects fixed in one place of the system may cause new unknown defects somewhere else in the system.

### Impact

Extends the time needed for testing.

### Severity

Low

### Strategy

Regression testing will be performed whenever defects are patched.

# Features to be Tested

This section details the features that will be tested in this test plan.

## STL File Input

### Description

The system shall provide a way for the user to select an STL file and then input that STL file into the system for processing.

### Priority

1 – Critical

## Graphical User Interface

### Description

The system shall provide a graphical user interface from which the user can import 3D models and initiate print operations. The GUI must be both intuitive and responsive.

### Priority

3 – Moderate

## Generate Machine Instructions

### Description

The system shall generate instructions needed by the printing hardware in order to print a given 3D object.

### Priority

1 – Critical

## Issue Machine Instructions

### Description

The system shall issue generated machine instructions from the software component to the printing hardware component.

### Priority

1 – Critical

## Monitor Temperature

### Description

The system shall monitor input from heat sensors attached to the printing hardware. The temperature of each extruder’s nozzle must be monitored at all times to ensure that material is extruded at the proper temperature.

### Priority

1 – Critical

## Monitor Position

### Description

The system shall monitor the position of the printing head at all times during operation. The system must be aware of the position of the printing head in order to adhere to a predefined printing path.

### Priority

1 – Critical

## Adhere to Material Constraints

### Description

The system shall adhere to the material constraints that limit the movement speed, extrusion rate, and nozzle temperature. Different materials have different properties that the system must account for in order to produce a properly printed object.

### Priority

1 - Critical

## Identify Materials

### Description

The system shall provide a method for the user to select the material for each discrete part that is being used for printing the 3D object.

### Priority

1 - Critical

## Identify Shapes

### Description

The system shall identify the shape of the object being printed by dividing it into smaller shapes for each individual material used.

### Priority

1 - Critical

## Determine Shape of Support Material Structure

### Description

The system shall determine the shape that the support material needs to be for stabilizing the 3D object as it is being printed. Without the support, the object could collapse during printing.

### Priority

1 - Critical

## Create Printing Path

### Description

The system shall determine a route that the printing head must follow as it prints.

### Priority

1 - Critical

## Database Interface

### Description

The system shall have an interface that allows the user to view what material is already stored in the database and enter new information for material not already stored.

### Priority

1 - Critical

## Store & Load Material Records

### Description

The system shall be able to load the material records stored in the materials database in order to control the temperature, movement speed, and flow speed of the nozzle at the correct setting.

### Priority

1 – Critical

## Slice Geometry into Thickness Levels

### Description

The system shall be able to process geometry in such a way as to generate sub-models of appropriate and customizable thickness such that the 3D printer can print each layer of the given thickness.

### Priority

1 – Critical

## Monitor Flow Sensors

### Description

The system shall monitor nozzle flow sensors and be able to maintain and adjust accordingly if the sensor begins to read out of bounds.

### Priority

1 – Critical

## Allow for UV Head Polymerization

### Description

The head shall be able to use UV light to cure or dry the extruded material. The system shall accommodate the use of UV to be turned on and off such that the material can be cured.

### Priority

1 – Critical

## Fill Density

### Description

The system shall allow for models to be processed in such a way that the geometry can be simplified or made more complex according to user specifications from any arbitrary model geometry. Such that the density of the path generated will provide a porous structure instead of a solid structure. This will allow the user to prototype a design without wasting large amounts of material or time.

### Priority

5 – Future

## Software Installer

### Description

The host software shall be delivered as an executable installer via USB flash memory and Compact Disc.

### Priority

1 – Critical

## Host Software to Printer Connection

### Description

The host software shall be connected to the printing hardware using a DE-9, DB-25, or Universal Serial Bus cable.

### Priority

1 – Critical

## Startup Time

### Description

The host software shall start in one minute or less.

### Priority

4 – Low

## STL Import Time

### Description

The host software shall import STL files in one minute or less.

### Priority

4 – Low

## Object Processing Time

### Description

The host software shall perform object processing and machine instruction generation in five minute or less.

### Priority

4 – Low

## GUI Responsiveness

### Description

The graphical components of the user interface shall be responsive to user interaction.

### Priority

3 – Moderate

## Real Time Sensor Monitoring

### Description

The system shall monitor data from sensors in real time during operation. The sensor data must be monitored in real time to ensure proper printer functionality as well as enforce safety systems.

### Priority

2 – High

## Temperature Cutoff Threshold

### Description

The system shall include a temperature cutoff threshold for the printer head. If the temperature of the printer head reaches the cutoff temperature, the system will abort the operation and shut off the heating device.

### Priority

1 – Critical

## Printing Area Restrictions

### Description

The system shall only extrude material within a configured area. Material extruded by the printer will be at a high temperature and may cause harm to the printer’s surroundings; therefore it is important to ensure that the material is only extruded in a specified safe area.

### Priority

1 – Critical

## Material Database

### Description

The system shall have a database that holds information about how the material is printed. For each material, the database must hold the diameter of the material filament, the temperature the filament must be extruded at, the extrusion speed, and weather a secondary head option such as UV curing light is required.

### Priority

1 – Critical

# Features Not to be Tested

This section details the features that will not be tested with this test plan. These are features that are verified by design or features that will be implemented in future iterations of the system.

## Monitor Door Switch

### Description

The system shall be able to use a sensor to monitor the 3D printer’s access door. Appropriate action should be taken when the door is opened during printing and ensure the door is closed prior to starting to print.

### Priority

5 – Future

## Graphical Object Models

### Description

The system shall display a graphical model of the objects represented by imported STL files. The user will be able to drag the graphical model around on a virtual printer bed in order to specify the location on the printer bed where the object will be printed.

### Priority

5 – Future

## Abstract Hardware Interface

### Description

The system shall allow for multiple different printers and multiple different heads to be used with minimal software change.

### Priority

1 – Critical

## Modular and Scalable Design

### Description

The software shall be developed using proven design principles to ensure that it can be scaled and maintained by future development teams. This design will allow the future requirements such as the Door Sensor to be implemented without difficulty.

### Priority

1 – Critical

# Testing Approach

## Overall Strategy

The overall strategy for this test plan is to start by testing the smallest testable units (modules), then test components (subsystems), then test layers. This approach is chosen because it will ensure that each part of a subsystem or layer is verified before testing the subsystem or layer as a whole. By verifying the parts before the whole, the source of test failures can be more easily traced. As described in the Test Items section of this document, the functionality of all modules, subsystems, and layers will be tested. Integration testing will also be performed to ensure that cross-layer interfaces are functioning properly.

Modules will be tested using JUnit tests as much as possible. The JUnit tests will use generated or mock input and test for an excepted output for each module. Any modules that cannot be directly tested using JUnit, or modules that require human observation for testing, will be tested using manually generated input with known expected output.

Components will be tested to ensure that all of their modules function properly as a subsystem. Component testing will be performed by providing input to a subsystem, then verifying that the subsystem creates to correct output or performs the correct action.

Integration testing will be performed to ensure that cross-layer interfaces are functioning properly. This testing will be a series of simple objects taken from STL to final printer object. At each layer the processing will be stopped using a debug mode so the objects can be inspected for completeness and correctness.

## Configurations

The system as a whole will be tested with the 3-D printer, currently being developed by the Mechanical team, connected through USB to a personal computer with the Windows XP SP 3 or higher operating system. No other hardware configurations will be used to test the system because there is no other known 3-D printer capable of testing all the features of this system. However, any 3-D printer using the RepRap firmware can be used to test the basic pathing and printing functionality of the system in case the Mechanical team cannot deliver a 3-D printer capable of performing the tests. A printer with the RepRap firmware can be provided by the project sponsor, Dr. Shiakolas, or the Mechanical team if necessary.

## Regression

Regression testing will be performed if any fixes or patches are applied to the system. Regression testing will involve the re-testing of all items that have already been tested up until the time the fix or patch is applied.

## Metrics

The metrics used to determine overall success of the project are described below.

|  |  |  |
| --- | --- | --- |
| Metric | Pass Criteria | Fail Criteria |
| Percentage of critical priority features verified | 100% | Less than %100 |
| Percentage of high priority features verified | 100% | Less than 100% |
| Percentage of moderate priority features verified | 75% - 100% | Less than 75% |
| Percentage of low priority features verified | 50% - 100% | Less than 50% |
| Branch and Line Coverage | 80%-100% | Less than 80% |

## Special Requirements

In order to properly test the system, access is needed to the 3-D printer under development by the Mechanical team. The 3-D printer must be capable of executing G-Code commands properly and providing position and temperature feedback.

# Item Pass/Fail Criteria

This section describes the criteria used to determine if a test passes or fails. The criteria is specified for each level of testing.

## Unit Testing Level

### Pass

Given valid input, the tested module returns the expected output. Given invalid input, the module throws an exception.

### Fail

Given valid input, the tested module fails to return the expected output. Given invalid input, the module fails to throw an exception.

## Component Testing Level

### Pass

Given valid input, the tested subsystem returns the expected output. Given invalid input, the module throws an exception.

### Fail

Given valid input, the tested subsystem fails to return the expected output. Given invalid input, the module fails to throw an exception.

## Integration Testing Level

### Pass

Given valid input, the tested layer returns the expected output. Given invalid input, the module throws an exception.

### Fail

Given valid input, the tested layer fails to return the expected output. Given invalid input, the module fails to throw an exception.

# Test Deliverables

This section will detail the materials and information to be delivered with the final product.

## JUnit tests

All modules will have at least one JUnit test to cover all lines and branches. This means for every control statement each possibility will be tested by a unit test. This deliverable includes:

1. Test Cases
2. Test Execution Code

## Test output reports

All unit tests will be run using a test runner such as NetBeans test executer. The runner’s output will be compiled in to a report to demonstrate that the tests were run and produced correct test results. This deliverable will consist of a text file detailing the regression suite output.

## Coverage Report

All of the unit tests will be run with a code coverage tool such as emma or jacoco. The output reports detailing code coverage will be included in the final report of the project. This will be in the form of printed HTML pages or printed images displaying the code and branch statistics.

## Test Results

All test done outside of JUnit will be detailed in a tabular format detailing the following information:

1. Test ID
2. Tester Name
3. Inputs
4. Expected Output
5. Actual Output
6. Pass/Fail
7. Notes

# Test Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Stage | Unit | Start | Finish |
| 1 | Persistence Framework | 3/18/14 | 4/1/14 |
| 1 | Command Structure | 3/18/14 | 4/1/14 |
| 1 | Database subsystem | 3/18/14 | 4/1/14 |
| 1 | Object Subsection Module | 3/18/14 | 4/1/14 |
| 1 | Object Translation Module | 3/18/14 | 4/1/14 |
| 1 | Normalization Subsystem | 3/18/14 | 4/1/14 |
| 1 | Preprocessing Layer | 3/18/14 | 4/1/14 |
| 1 | Slicing Engine Wrapper | 3/18/14 | 4/1/14 |
| 1 | Slicing Engine | 3/18/14 | 4/1/14 |
| 1 | Processing Layer | 3/18/14 | 4/1/14 |
| 1 | Parser Module | 3/18/14 | 4/1/14 |
| 1 | Unification Module | 3/18/14 | 4/1/14 |
| 1 | G-Code Preparation | 3/18/14 | 4/1/14 |
| 1 | Post Processing Layer | 3/18/14 | 4/1/14 |
| 1 | Serialization Module | 3/18/14 | 4/1/14 |
| 1 | TxRx Module | 3/18/14 | 4/1/14 |
| 1 | Deserialization Module | 3/18/14 | 4/1/14 |
| 1 | Communications Subsystem | 3/18/14 | 4/1/14 |
| 1 | Communications Layer | 3/18/14 | 4/1/14 |
| 2 | Print Job GUI Module | 4/1/14 | 4/15/14 |
| 2 | Controller Subsystem | 4/1/14 | 4/15/14 |
| 2 | Print Job Controller | 4/1/14 | 4/15/14 |
| 2 | Printer State Controller (module) | 4/1/14 | 4/15/14 |
| 2 | Printer State Controller (subsystem) | 4/1/14 | 4/15/14 |
| 2 | Printer Control Layer | 4/1/14 | 4/15/14 |
| 3 | Import GUI Module | 4/15/14 | 4/29/14 |
| 3 | Printer Configuration GUI Module | 4/15/14 | 4/29/14 |
| 3 | Material GUI Module | 4/15/14 | 4/29/14 |
| 3 | Print Configuration GUI Module | 4/15/14 | 4/29/14 |
| 3 | Extruder GUI Module | 4/15/14 | 4/29/14 |
| 3 | Status GUI Module | 4/15/14 | 4/29/14 |
| 3 | GUI Subsystem | 4/15/14 | 4/29/14 |
| 3 | Import Controller | 4/15/14 | 4/29/14 |
| 3 | Printer Configuration Controller | 4/15/14 | 4/29/14 |
| 3 | Material Controller | 4/15/14 | 4/29/14 |
| 3 | Print Configuration Controller | 4/15/14 | 4/29/14 |
| 3 | Extruder Controller | 4/15/14 | 4/29/14 |
| 3 | Controller Subsystem | 4/15/14 | 4/29/14 |
| 3 | User Interface Layer | 4/15/14 | 4/29/14 |
| 3 | Dispatch Module | 4/15/14 | 4/29/14 |
| 3 | State Monitoring | 4/15/14 | 4/29/14 |
| 3 | Printer Feedback Layer | 4/15/14 | 4/29/14 |

Table 10‑1 : Test Schedule

# Approvals

This section lists the approvals of all individuals involved with this System Test Plan.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Signature | Date |
| Mike O’Dell | Project Supervisor |  |  |
| Panos Shiakolas | Project Sponsor |  |  |
| Daniel Lain | Team Lead |  |  |
| Jesse Bowles | Team Member |  |  |
| Tim Edmondson | Team Member |  |  |
| Shawn Simonson | Team Member |  |  |

Table 11‑1 : Approvals