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 6](#_Toc382037559)

[1.1 Purpose 6](#_Toc382037560)

[1.2 Scope 6](#_Toc382037561)

[2 References 6](#_Toc382037562)

[2.1 System Requirements Specification 6](#_Toc382037563)

[2.1.1 Customer Requirements 7](#_Toc382037564)

[2.1.2 Packaging Requirements 8](#_Toc382037565)

[2.1.3 Performance Requirements 9](#_Toc382037566)

[2.1.4 Safety Requirements 9](#_Toc382037567)

[2.1.5 Maintenance and Support Requirements 10](#_Toc382037568)

[2.1.6 Other Requirements 10](#_Toc382037569)

[2.2 Architecture Design Specification 11](#_Toc382037570)

[2.2.1 Interface Layer 12](#_Toc382037571)

[2.2.2 Pre Processing Layer 12](#_Toc382037572)

[2.2.3 Processing Layer 12](#_Toc382037573)

[2.2.4 Post Processing Layer 12](#_Toc382037574)

[2.2.5 Printer Control Layer 12](#_Toc382037575)

[2.2.6 Printer Feedback Layer 12](#_Toc382037576)

[2.2.7 Communications Layer 13](#_Toc382037577)

[2.2.8 Dataflow Definitions 13](#_Toc382037578)

[2.2.9 Layer Requirements Traceability 14](#_Toc382037579)

[2.3 Detail Design Specification 15](#_Toc382037580)

[2.3.1 Prologue 15](#_Toc382037581)

[2.3.2 Data Flow Definitions 17](#_Toc382037582)

[2.3.3 Module Requirements Traceability 18](#_Toc382037583)

[3 Test Items 21](#_Toc382037584)

[3.1 Product Version 21](#_Toc382037585)

[3.1.1 Stage one 21](#_Toc382037586)

[3.1.2 Stage Two 21](#_Toc382037587)

[3.1.3 Stage Three 21](#_Toc382037588)

[3.2 Design Decomposition 22](#_Toc382037589)

[3.3 Unit Tests 22](#_Toc382037590)

[3.3.1 Command Structure 22](#_Toc382037591)

[3.4 Component Tests 23](#_Toc382037592)

[3.4.1 Database Subsystem 23](#_Toc382037593)

[3.5 Integration Tests 24](#_Toc382037594)

[3.6 Validation Tests 24](#_Toc382037595)

[4 Risks 25](#_Toc382037596)

[5 Features to be Tested 26](#_Toc382037597)

[5.1 STL File Input 26](#_Toc382037598)

[5.2 Graphical User Interface 26](#_Toc382037599)

[6 Features Not to be Tested 27](#_Toc382037600)

[7 Testing Approach 28](#_Toc382037601)

[7.1 Overall Strategy 28](#_Toc382037602)

[7.2 Configurations 28](#_Toc382037603)

[7.3 Regression 28](#_Toc382037604)

[7.4 Metrics 28](#_Toc382037605)

[8 Item Pass/Fail Criteria 29](#_Toc382037606)

[8.1 Unit Testing Level 29](#_Toc382037607)

[8.2 Component Testing Level 29](#_Toc382037608)

[8.3 Integration Testing Level 29](#_Toc382037609)

[9 Test Deliverables 30](#_Toc382037610)

[9.1 JUnit tests 30](#_Toc382037611)

[9.2 Test output reports 30](#_Toc382037612)

[9.3 Coverage Report 30](#_Toc382037613)

[9.4 Test Results 30](#_Toc382037614)

[10 Test Schedule 31](#_Toc382037615)

[11 Approvals 32](#_Toc382037616)

List of Tables

Table # Title Page #

[Table 2‑1: Customer Requirements 8](#_Toc382037617)

[Table 2‑2 : Packaging Requirements 8](#_Toc382037618)

[Table 2‑3 : Performance Requirements 9](#_Toc382037619)

[Table 2‑4 : Safety Requirements 9](#_Toc382037620)

[Table 2‑5 : Maintenance and Support Requirements 10](#_Toc382037621)

[Table 2‑6 : Other Requirements 11](#_Toc382037622)

[Table 2‑7: Architecture Dataflow Definitions 14](#_Toc382037623)

[Table 2‑8 : Layer Requirements Traceability 15](#_Toc382037624)

[Table 2‑9 : Data Flow Definitions 17](#_Toc382037625)

[Table 2‑10 : UI Layer Requirements Traceability 18](#_Toc382037626)

[Table 2‑11 : Processing Layers Requirements Traceability 19](#_Toc382037627)

[Table 2‑12: Printer State, Comm., and Printer Feedback Requirements Traceability 20](#_Toc382037628)

[Table 3‑1 : Command Structure Tests 23](#_Toc382037629)

[Table 3‑2 : Database Subsystem Tests 24](#_Toc382037630)

[Table 10‑1 : Test Schedule 31](#_Toc382037631)

List of Figures

Figure # Title Page #

[Figure 2‑1: Architecture Dateflow Drawing 11](#_Toc382037632)

[Figure 2‑2: Detail Design Drawing 16](#_Toc382037633)

# 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 ‑: 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 ‑ : 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 ‑ : 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 ‑ : 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 ‑ : 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 ‑ : 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 ‑: 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 ‑ : 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 ‑ : 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 ‑ : 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 ‑ : 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 ‑: 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

[Drawing]

## Unit Tests

### Command Structure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| 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 | 1 - Critical |
| CS2 | * Function Call * String | * Fully formed Printer Configuration Object | Call get Printer Config function with a file name. | 1 – Critical |
| CS3 | * Function Call * String | * Fully formed Extruder Configuration Object | Call get Extruder Config function with a file name. | 1 – Critical |
| CS4 | * Function Call * String | * Fully formed Material Configuration Object | Call get Material Config function with a file name. | 1 – Critical |
| CS5 | * Function Call * String | * Fully formed Print Job Configuration Object | Call get Print Job Config function with a file name. | 1 – Critical |
| CS6 | * Function Call * String | * Fully formed Print Configuration Object | Call get Print Config function with a file name. | 1 – Critical |
| CS7 | * Function Call * Printer Configuration Object | * Boolean True | Call save Printer Configuration function with a file name. | 1 – Critical |
| CS8 | * Function Call * Extruder Configuration Object | * Boolean True | Call save Extruder Configuration function with a file name. | 1 – Critical |
| CS9 | * Function Call * Material Configuration Object | * Boolean True | Call save Material Configuration function with a file name. | 1 – Critical |
| CS10 | * Function Call * Print Job Configuration Object | * Boolean True | Call save Print Job Configuration function with a file name. | 1 – Critical |
| CS11 | * Function Call * String | * Boolean True | Call delete Configuration function with a file name. | 1 – Critical |

Table ‑ : Command Structure Tests

### Subsection Module

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| UPr1 | Start Z and End Z for each subsection and a Parent STL File for each material in each 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. | 3 – Moderate |

Table ‑: Subsection Module Tests

### File Translation Module

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| UPr2 | Subsection STL files for each subsection and 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. | 3 – Moderate |

Table ‑ : File Translation Module Tests

### Slicing Engine Wrapper

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| UP1 | 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. | 3 – Moderate |

Table ‑ : Slicing Engine Wrapper Tests

### Parser Module

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| UPo1 | 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. | 3 – Moderate |

Table ‑ : Parser Module Tests

### Unification Module

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| UPo2 | 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. | 4 – Low |

Table ‑ : Unification Module Tests

## Component Tests

### Database Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| 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 | 1 - Critical |
| PF2 | * Function Call * String | * Fully formed Printer Configuration Object | Call get Printer Config function with a file name. | 1 – Critical |
| PF3 | * Function Call * String | * Fully formed Extruder Configuration Object | Call get Extruder Config function with a file name. | 1 – Critical |
| PF4 | * Function Call * String | * Fully formed Material Configuration Object | Call get Material Config function with a file name. | 1 – Critical |
| PF5 | * Function Call * String | * Fully formed Print Job Configuration Object | Call get Print Job Config function with a file name. | 1 – Critical |
| PF6 | * Function Call * String | * Fully formed Print Configuration Object | Call get Print Config function with a file name. | 1 – Critical |
| PF7 | * Function Call * Printer Configuration Object | * Boolean True | Call save Printer Configuration function with a file name. | 1 – Critical |
| PF8 | * Function Call * Extruder Configuration Object | * Boolean True | Call save Extruder Configuration function with a file name. | 1 – Critical |
| PF9 | * Function Call * Material Configuration Object | * Boolean True | Call save Material Configuration function with a file name. | 1 – Critical |
| PF10 | * Function Call * Print Job Configuration Object | * Boolean True | Call save Print Job Configuration function with a file name. | 1 – Critical |
| PF11 | * Function Call * String | * Boolean True | Call delete Configuration function with a file name. | 1 – Critical |

Table ‑ : Database Subsystem Tests

### Normalization Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| UPo2 | Start Z and End Z for each subsection and a 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. | 3 – Moderate |

### Slicing Engine Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| UP1 | 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 Subsystem. Observe the output G-Code files for correctness. | 3 – Moderate |

### G-Code Preparation Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Output/Action | Test | Risk |
| UPo2 | The printer G-Code flavor, printer custom start G-Code, printer custom end G-Code, and 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. | 3 – Moderate |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Hardware | Input | Expected Output/Action | Test | Risk | Dependent Tests |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Integration Tests

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Hardware | Input | Expected Output/Action | Test | Risk | Dependent Tests |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Validation Tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Hardware | Input | Expected Output/Action | Test | Risk |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Risks

[Risks bad test, poor coverage, configurability, ME team, and 3rd party software.]

# Features to be Tested

## STL File Input

**5.1.1 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.

**5.1.2 Source:** Dr. Panos S. Shiakolas (Sponsor)

**5.1.3 Constraints:** The system must support multi-material objects but the STL file format does not support material information. Model must be closed / complete.

**5.1.4 Standards:** STL (STereo Lithography) File Format

**5.1.5 Priority:** 1 – Critical

## Graphical User Interface

**5.2.1 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.

**5.2.2 Source:** Dr. Panos S. Shiakolas (Sponsor)

**5.2.3 Constraints:** None

**5.2.4 Standards:** Java Swing/AWT

**5.2.5 Priority:** 3 - Moderate

# Features Not to be Tested

[Modular scalable design, Abstract hardware interface]

# 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 performed by comparing the output from the sending layer to the input received by receiving layer. If these values match, then the test is deemed successful. Otherwise, the test fails.

## 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 7 or higher operating system. No other hardware configurations will be used to test the system as whole because there is not known to be another 3-D printer capable of testing all the features in 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 if the Mechanical team cannot deliver a 3-D printer capable of performing testing with. 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% |

## 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

## 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 on 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 runners output will be compiled in to a report to demonstrate that the test 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 folloing 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 ‑ : Test Schedule

# Approvals