­

Pulley Print Program

Design Document 3.2

|  |
| --- |
| Team Dec15-08 Client: Van Gorp Corporation Advisor: Dr. Tien Nguyen Team: Austin Benson, Nick Cervantes, Sean Finn, Evan Kroeger,  Edwin Martinez Velaquz, Matt Petron |

Table of Contents

1. [Introduction 2](#_Toc432610655)

[Background 2](#_Toc432610656)

[Project Scope 2](#_Toc432610657)

1. [System Level Design 3](#_Toc432610658)

[System Requirements 3](#_Toc432610659)

[Functional Requirements 3](#_Toc432610660)

[Non-Functional Requirements 3](#_Toc432610661)

[Functional Decomposition 4](#_Toc432610662)

[Main Application 4](#_Toc432610663)

[Engineering Standards Module 4](#_Toc432610664)

[Machinist Directions Module 4](#_Toc432610665)

[Bill of Materials Module 4](#_Toc432610666)

[Data Source Integrity Module 5](#_Toc432610667)

[ERP System (RealMacola) CSV Exporter Module 5](#_Toc432610668)

[System Analysis 5](#_Toc432610669)

[System Block Diagram 5](#_Toc432610670)

1. [Detailed Description 8](#_Toc432610671)

[I/O Specification 8](#_Toc432610672)

[Interface Specification 8](#_Toc432610673)

[Hardware/Software Specifications 8](#_Toc432610674)

[Implementation Challenges 8](#_Toc432610675)

[Excel Documents 9](#_Toc432610676)

[Testing Procedures 10](#_Toc432610677)

1. [Prototype 10](#_Toc432610678)
2. [Conclusion 11](#_Toc432610679)
3. [Modification Report 12](#_Toc432610680)

# Introduction

## Background

Van Gorp Corporation manufactures standard and customized pulley systems for clients. Common clients for pulleys include FedEx for shipping packages and the mining industry for convey belts. Because these pulleys are not trivial to construct, Van Gorp requires a custom consultation to order special pulleys. Upon receiving a request to build a custom pulley, Van Gorp uses a software program to calculate the dimensions of over twenty-five components of the custom pulley. The standards used to properly calculate the component dimensions are currently hardcoded into this program. This program was written in GW Basic more than twenty years ago and the original author has since left the company. Because of this, the program has become cumbersome to maintain and is unable to be modified by Van Gorp’s employees. The current program also uses a Command Line Interface (CLI) which is not user friendly. In addition, the current program does not run natively on 32 or 64-bit computers, leaving an emulated environment as the only option for Van Gorp to run this program.

The primary goal of this project is replace Van Gorp’s existing solution with one which is maintainable, extensible, and is compatible with Van Gorp’s current IT environment. In particular, completely updating the code base to a more modern programming language will give Van Gorp the ability to use this program for the foreseeable future. This will also allow the company to run the program on their 64-bit computers without the need for special software. Van Gorp has also requested the software have a graphical user interface to replace the current cumbersome CLI.  The current CLI asks for user input to questions one at a time whereas a forms-based application will allow those questions to be condensed reducing the amount of time it takes to enter in the necessary information to calculate the dimension of the pulley.

## Project Scope

The focus for Phase I of the project will be to ensure that we have created a user friendly interface for the system, are able to convert Van Gorp’s Pulley Standards into a more usable format and the system, are able to parse through the Excel documents, and can produce the correct dimensions for drum class pulleys.  Because Van Gorp produces many types of pulleys, we will focus on implementing the program to output the measurements for the more common and simpler pulleys during this phase. In Phase II, we will add in the support of over-writable dimensions, support for all required pulley types, a bootstrapped user interface and customizability for the Pulley Standards used in the system.

After the completion of these first two phases, we plan on surpassing our project’s requirements by improving Van Gorp’s efficiency by automating more steps in the custom pulley creation process. Firstly, we plan on integrating our system’s output with Van Gorp’s Enterprise Resource Planning system. Next, we plan on developing a “Bill of Materials” output for Van Gorp which will estimate the cost of constructing the custom pulley. Lastly, we intended to develop “machinist instructions” which provide directions for constructing the custom pulleu to Van Gorp’s floor operators.

At the completion of this project, a compiled and runnable executable program will be delivered to Van Gorp, along with all code and documentation for this program.  Any materials borrowed from Van Gorp, including but not limited to Van Gorp’s Engineering Standards will be returned to Van Gorp and any copies either hard or electronic, will be securely disposed.

# System Level Design

## System Requirements

### Functional Requirements

#### Primary

##### Refactor the program using a newer programming language

* Current program only executes on machines capable of running 8-bit software which forces any 64-bit machine to use an emulator before it can be run.
* Old code is written in GWBasic, which makes it very difficult to maintain. A newer language is required.

##### Develop a Graphical User Interface which takes less time than the current interface

* Current program is a command line interface and many features of the program are no longer in use, and therefore complicate the interface
* Ideally would like a forms-based interface with dropdown menus, checkboxes, etc.

##### editing of Engineering Standards

* Current program’s engineering standards are hard coded, and many of these standards are outdated.
* Van Gorp would like the standards to be read in from files that already exist in their workflow.

#### Secondary

##### Implementing Automation

* Current process is not streamlined. Requires much manual input from one system to another. The output of the Pulley Print Program should be able to interact with other system.
* Pulley Print Program should be able to automatically size materials.

##### Integration with Current ERP System

* Current ERP system has SQLServer backend. The output of the revised program should directly interface with SQLServer. This step would eliminate the need for manual entry.

### Non-Functional Requirements

##### Performance

* Program executes quickly and easily
* Computation is minimal, so that other running applications are not affected

##### Scalability

* Current codebase is “spaghetti code”
* Clean codebase ensures easier readability and bug tracking
* Can easily implement newer features in the future

##### Reliability

* New program will not crash under any circumstance
* Generated output will be correct
* Recognize invalid input or configurations, according to their standards

##### Maintainability

* Codebase will be easy to read and understand
* Will be well documented and commented
* Engineering standards will be easy to add, remove or change

##### Portability

* Application will be easy to install, use, and run on Van Gorp’s computers
* No confusing batch files or emulated environments like in the current implementation

##### Security

* Ensure the company’s intellectual property remains safe
* Minimize risk when integrating with ERP system

## Functional Decomposition

### Main Application

Our main application will be written in C# .NET and designed using Windows Forms. This application will run on Van Gorp’s newer, 64-bit machines. The application will be a standalone application which can be installed on any of Van Gorp’s systems. This portion of the application contains all of the necessary logic to direct the user to another portion of the program.

### Engineering Standards Module

We will need to have a module to read engineering standards that Van Gorp has established, and are necessary for our application’s calculations. This will also be written in C# .NET, and will parse through the necessary Excel documents and return the data needed.

### Machinist Directions Module

This module is used to provide directions to Van Gorp’s floor operators when constructing the pulley. This includes information on which routers and welders to use when making the parts of the pulley. This portion of the program is also written in C# .NET.

### Bill of Materials Module

This module is used to calculate the cost of producing the pulley in terms of raw materials. This module performs calculations to determine the cost, quantity, and volume of material used in each part of the pulley’s construction. The program then outputs the desired information in a format useful for Van Gorp’s business team.

### Data Source Integrity Module

This module is used to perform data source integrity checks by validating the integrity of each Excel sheet. This module will be written in C# .NET and will validate the Excel documents and their contents.

### ERP System (RealMacola) CSV Exporter Module

Van Gorp also has an ERP system currently in place used to keep track of materials used in the construction of the pulleys. The ERP system accepts tab-separated files as a form of batch import into the CSV program. This module of application formats the data so it can be imported in to the ERP system.

## System Analysis

The primary principles behind our design are usability, extendibility, and maintainability. Above all else we want our program to be usable, meaning that our client can use the program to achieve its primary purpose of calculating pulley dimensions. We intend to do this by using a straight-forward Windows Form which directly displays information to the user as opposed to using a command line interface.

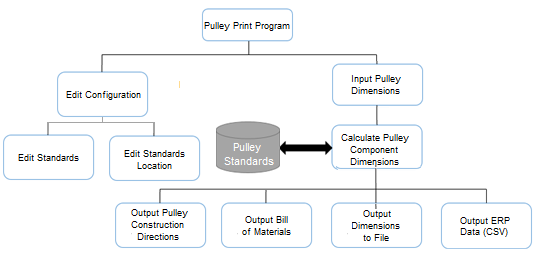
We also want our design to be extensible. One of the major reasons Van Gorp wants to replace their previous version of the Pulley Print Program is because it does not have the capability to easily interface with other pieces of Van Gorp software. By using a C# .NET Windows Forms application, we leverage the power of the C# .NET platform to integrate with other Windows-based products including SQL Server, Excel files, and word documents.

Lastly, we want our program to be maintainable. C# .NET has well-documented test suites which will help in the verification portions of our project. By focusing testing as we develop, we will be able to ensure our software has a minimal number of bugs. By focusing on testing throughout the development cycle, we will ensure of software has few bugs, and will be verified through regression testing. Our use of regression testing will help to ensure any future modifications to the software will not break the features which have already been verified to work.

## System Block Diagram

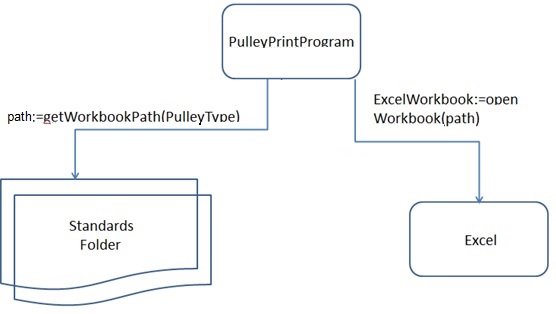
Our Pulley Print Program has the following operational flow. Our program uses a minimalistic user interface with basic functions for editing standards and using the pulley print program to calculate pulley dimensions.

##### Figure 1: Operational Flow Diagram



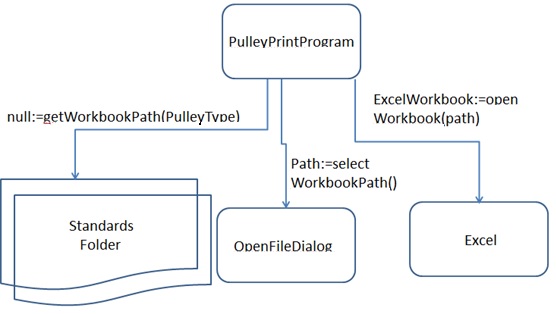
Once the user has entered all of the information necessary to construct a pulley our Pulley Print Program then goes to a pre-defined “standards location” to look for Excel documents containing Van Gorp’s Pulley standards. This “standards location” is set in the user’s settings and can be easily modified. The application flow for editing a standard is shown below:

##### Figure 2: Application Flow – Standards File Found



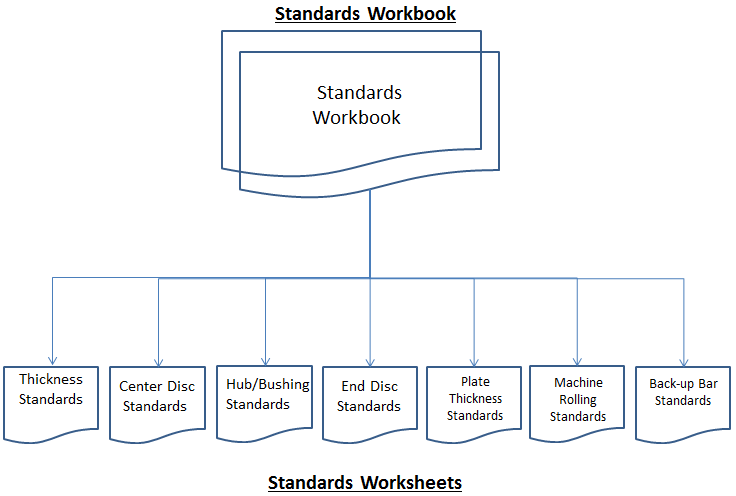
If the standards folder does not exist, or a standards file is not found then the system will prompt the user to select the location of the standards Excel documents. The application flow in this instance is shown below:

##### Figure 3: Application Flow – Standards File Not Found



Van Gorp’s ISO standards have been converted into Excel documents so they can be read in by our program. Each of Van Gorp’s pulley classes have their own standards which have been converted into separate Excel workbooks. Each workbook contains worksheets which have specific pulley standards such as the thickness of the pulley rim, or the number of center discs used in the construction of the pulley. Our program parses through the Excel document to calculate the correct dimensions for constructing the custom pulley. The structure of the standards is shown in the diagram below.

##### Figure 4: Standards Workbook Diagram



# Detailed Description

## I/O Specification

All input and output will be visible and accessible to the user via a desktop application that will be developed using Windows Forms.  Since this application deals a lot with different questions and answers, there will be a series of text boxes, drop down menus, and checkboxes used to gain information from the user about the pulley they wish to create.  To help make this more user friendly, portions of the form will be generated as the user enters information into the system, allowing us to show only the fields that will be required for any calculations.  After all of the fields have been filled out for a specific pulley, the user will submit their values and the application will output to the screen that will have all of the information needed to send the pulley to the manufacturing floor.  The user will also be able to send this information to a PDF document for use elsewhere in the company.

## Interface Specification

After the system receives user input for a set of fields, a set of engineering standards, which will be stored in Excel documents, will be accessed to generate the appropriate output for the system. These engineering standards will act as look up tables to determine different measurements for the pulleys. When the system is being set up on a user’s machine, the location of these engineering standards Excel documents will need to be set.  Once this information is provided and saved, these documents will be accessed as read only by the application and data will be pulled directly from them.

## Hardware/Software Specifications

The original Pulley Print Program was written on an 8-bit Apple II, thus it could not run natively on modern 32 and 64-bit computers. The current and temporary solution is to run the old program on a DOS-Box emulator. This solution does not allow for most Van Gorp employees to run the program on their computer. Our design specifies that the program will work on the IT infrastructure currently present at Van Gorp. Because our new program is a Windows Forms Application written in C# .NET, it will be extremely portable and able to run any Van Gorp workstation. Because of our platform choice, our program will be allowed to run on modern 32-bit and 64-bit machines.

## Implementation Challenges

The biggest component of this program comes from the engineering standards set by Van Gorp Corp. These are very sensitive documents that are proprietary to the company, and must remain secure. This poses a problem because they are a crucial part in the computations done by the program, so our program must be able to access these formulas. Furthermore, they must be easily modifiable if the standards were ever to change. These standards were kept in Word documents which are difficult if not impossible to parse using automated scripts. In addition, , which makes it hard for a program to read in.  We explored various alternatives that we could use to store these standards and decided to store the information in Excel Documents.  Information about these Excel Documents are detailed in the following section.

## Excel Documents

Before deciding to use Excel Documents, we explored a variety of ways to store Van Gorp’s engineering standards for the purposes of our program. The engineering standards were stored in word documents which included descriptions, title, and tables with the engineering standard values. After carefully considering all options for storing  A SQL Database and Excel Documents were the final two options.  We ultimately chose Excel over SQL for five main reasons:

1. **Cost** – every single Van Gorp system is equipped with Microsoft office, whereas the cost for an instance of Microsoft SQL Server 2014 costs around $3,717 per processor core to purchase an in-house server. Because of Van Gorp’s security requirements, leasing hosted server space is not an option due to the need to store secure documents on site. This would be an excess cost for a business the size of Van Gorp to incur to support a program which is used anywhere between 5 and 25 times per month. Excel allowed us to meet our requirements for no additional cost.
2. **Ability to Port Standards from Word** – In order to convert standards. Using SQL Server we would have had to manual input each standard, or copy the standards to a comma-separated-value file for batch import into the SQL Server. The process of manually inputting all engineering standards would be very time-consuming for our team and distract from the core engineering work of developing our core application. In addition, the same process would be required to covert the standards in to CSV is used when copying the standards over to Excel documents.
3. **Extensibility** – Most Van Gorp employees have experience with Microsoft Excel, however very few of them have experience with SQL Server. In order to add new standards rows for new classes of pulleys, Excel documents can be easily modified by adding rows for new sizes of pulleys. This is as simple as adding a new row to an Excel document. In the case of SQL Server, this would require a Van Gorp employee to access the database and use the SQL Server database editor program each time a new pulley size is added.
4. **User Interface** –Many of Van Gorp’s engineering standards have comments, descriptions, and logic which is written out in plain text in addition to engineering tables. Excel sheets have the ability to have blank headers row with extensive comments and documentation describing the engineering standards. In addition, Excel sheets can be formatted with different fonts, text sizes, bolding, and colors to present the engineering standards in a more readable manner. Database rows and columns do not have this luxury.

Currently, we have created Excel files which contain sheets corresponding to the exact standards specified in Van Gorp’s word documents. The sheets are titled based on name of the engineering standards they represent (Ex: ENG-400.09). These sheets are then used as the data source for the engineering standards and parsed by our program to determine the correct values for each measurement of the program. If there is any need to update these sheets with new engineering standards, the parsing logic remains the same.

To protect the information in these Excel Documents from unauthorized viewing and editing, Van Gorp explicitly said they want to take care of protecting all the Cells and Sheets using the built-in Protected Sheets functions of Excel.  We are going to make sure that the program will still function correctly when the Excel sheets are password protected.

## Testing Procedures

During development of this software, multiple testing methods were used to ensure the program operates as expected:

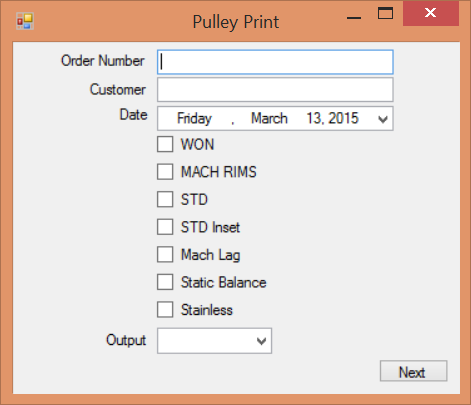
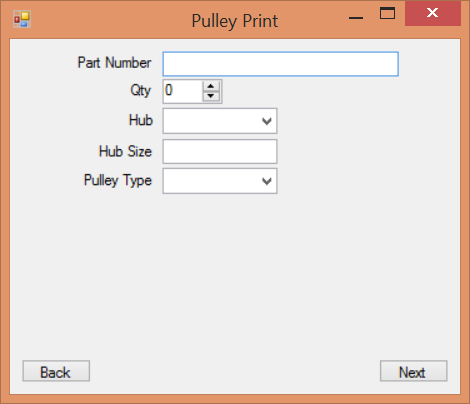
1. **Unit Testing** – In order to verify the correctness of our program, each module is complete with unit testing
2. **Beta Version Testing** – Following Phase I implementation, our team presented a beta version of the software to Van Gorp to determine if we were building an application which met their needs. After this presentation our team gained valuable insight on the need to improve the format of our output and to allow the ability to overwrite values.
3. **Data Source Integrity Checking** – Because the cost of building a pulley is high, keeping mistakes to a minimum is of the utmost priority. In order to prevent errors in the data source documents, our Pulley Print Program has an “integrity check” function which runs every time the Excel sheets are modified. This program loops through each value in the Excel sheet to ensure it is a number and falls within the expected range. This program only runs if the Excel worksheet has been updated.
4. **Log file Output** – As a “paper-trail” for each pulley output generated by the pulley print program, we generate a log file which keeps track of exactly which cells contained which value used in calculating the output. In addition, each formula used and the operands for each equation is also output in this log file.

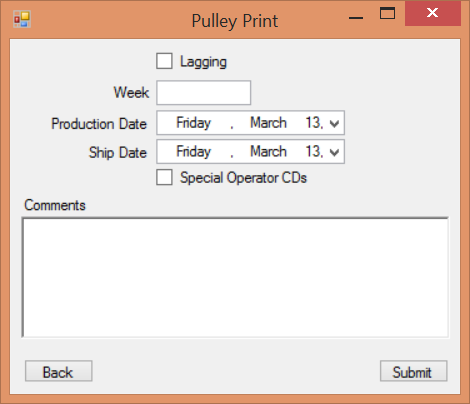
Overall, our testing methods combined with extensive documentation of code (through the use of StyleCop, a code-health program) has allowed us to develop a program with the minimal amount of errors.

# Prototype

To give Van Gorp a better understanding of the type of application that we are going to produce for them, a prototype was created based off of the current system they are using.  The main idea was to get an idea of how the user would interact with the system to get the information that they need.  Without knowing a whole lot about what information the users will need to provide and what information that the users will expect to have returned, we created a simple sample form.  All of the information fields are likely to be altered in the final version of the application.

##### Figure 5: Prototype Screens

 ****

****

# Conclusion

Overall, our design is structured to meet Van Gorp’s requirements, and exceed their expectations by improving the efficiency of the Pulley Print Program in their workplace. We leverage the power of a Windows Forms and C# .NET to make our application user-friendly, extensible, and maintainable. We use a data source which meets Van Gorp’s needs and provides them with the most cost effect solution. In addition to our rigorous testing procedures, our use of rapid feedback has allowed us to develop the right product for Van Gorp. Our design document will remain a fluid changing document used to keep record of our decisions and allow for the extension of our design.

# Modification Report

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change |
| 1.0 | 02/21/15 | AB, NC, SF, EK, MP | Initial Document |
| 1.1 | 02/23/15 | MP | Included Modification Table |
| 2.0 | 04/23/15 | AB, SF, MP | Updated Scope, Block Diagram |
| 2.1 | 09/12/15 | EK | Modified block diagram per Dr. Amarucai’s advice |
| 2.2 | 09/16/15 | MP | Updated Background and project scope |
| 2.3 | 09/18/15 | NC | Expanded upon Excel Documents section |
| 3.0 | 09/21/15 | AB, SF | Updated to reflect all planned features and newest design |
| 3.1 | 09/24/15 | EMV | Added section on outputting to Macola ERP |
| 3.2 | 10/01/15 | AB | Updated Testing Procedures section |