­

Pulley Print Program

Project Plan 3.6

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

Table of Contents

1. [Background 2](#_Toc417994565)
2. [Problem Statement 2](#_Toc417994566)
3. [System Requirements 3](#_Toc417994567)

[Functional Requirements 3](#_Toc417994568)

[Non-Functional Requirements 4](#_Toc417994569)

[Proposed Solution 4](#_Toc417994570)

[Evaluation of Alternatives 6](#_Toc417994571)

[Different Language Selection 6](#_Toc417994572)

[Working On Secondary Requirements Earlier 6](#_Toc417994573)

[Validation and Acceptance Test 6](#_Toc417994574)

1. [Interface/System Description 7](#_Toc417994575)

[Current System State 7](#_Toc417994576)

[Technical Approach 7](#_Toc417994577)

[Process Details and Deliverables 8](#_Toc417994578)

[Test Plan 9](#_Toc417994579)

1. [Work Breakdown Structure 9](#_Toc417994580)

[Project Schedule 9](#_Toc417994581)

[Risk/Feasibility Assessment 10](#_Toc417994582)

[Cost Considerations 11](#_Toc417994583)

[Work Breakdown Structure 11](#_Toc417994584)

1. [Conclusion 12](#_Toc417994585)
2. [Modification Report 12](#_Toc417994586)

# Background

Van Gorp Corporation is an Iowa based manufacturer of quality products that are used in conveyor components throughout the world. Manufactured items include conveyor pulleys, shafting, rubber lagging, bearings, rollers, take-up frames, and other related material handling products. These conveyor pulleys can be found in many different areas including mining, construction, and the shipping industries.

As a part of their services, Van Gorp Corporation provides its customers the option to purchase and order standard pulley systems as well as custom systems. These pulley systems have been designed in house and have been ISO certified for quality. As a result of their disciplined focus, Van Gorp Corporation continues to have a role as one of the largest conveyor pulley manufacturer in North America.

# Problem Statement

A large part of Van Gorp Corporation’s business is creating custom conveyor pulleys. To ensure that the pulleys they create meet certain criteria, they have developed a set of equations and standards that are used when designing their pulleys. These standards and equations are used by a few of the employees who deal with creating the purchase order to be manufactured on the floor.

In their current system, Van Gorp Corporation receives a purchase order from their customers via multiple different sources, including email, phone, or through their website. From there a small portion of the staff has the ability to take these purchase orders and create work orders to be manufactured. If the part that has been ordered is a standard part, or something that has been ordered before, the dimensions can simply be pulled from a database and filled into the work order. However, custom or new products must go through additional steps to ensure that the highest quality product is produced.

After receiving a new part for production, to ensure that the product will meet the standards that have been set by Van Gorp Corporation all of the details about what type of pulley is being requested must be entered into a very outdated command line interface (CLI) system that uses the standardized equations and tables that have been assembled. The program then prints out all the dimensions needed what machines to use, and any other information that may be needed on the manufacturing floor to create the pulley. The information produced from this program is then entered into another software program that is used to create and store all of the work orders that are sent out to the floor for production.

This old system used to generate all of the different calculations needed for the pulleys was created over two decades ago and uses the programming language GW-BASIC. Since this programming language is no longer widely used, it has been very difficult for the staff at Van

Gorp to keep the program updated, and as a result, some of the standards are outdated. Also, the program was originally designed to be run on 8 or 16-bit machines. Due to this, it cannot be run on the newer machines being used in the company and must be run through either an emulator, or on an older and slower machine.

The main goal for this senior design project is to update the system to a more current programming language, as well as make it more user friendly for the employees to use. Along the way, Van Gorp Corporation wishes to update any outdated information that has changed since the initial development of the program. After this goal has been met, other ways to improve the application will be explored. Some of these potential features include adding functionality to update any information needed in the engineering standards tables or equations easily from within the application, adding support to directly port the information into the program that generates the work orders, and the ability to add new options to the list of pulley configuration options.

# 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 GW-Basic, which makes it very difficult to maintain. A newer language is required.

##### Develop a Graphical User 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 GUI with dropdown menus, checkboxes, etc.

##### Allow editing of Engineering Standards

* Current program’s engineering standards are hard coded, and many of these standards are outdated.
* There is no way to modify the current engineering standards without changing the code.
* Van Gorp would like for all standards to be modifiable and extendible.

#### Secondary:

##### Direct program output to a PDF which will be used by Van Gorp’s floor operators

* The current process involves a Van Gorp employee using the Pulley Print program to calculate the dimensions of a pulley. The employee then manually enters the output of the program into a form used by a machine operator to construct the pulley.
* The new pulley print program should be able to output directly to a pdf which will then be used by a Van Gorp machine operator to construct the pulley.

##### Automatic Bill of Materials Output Summary

* Van Gorp needs to output a Bill of Materials section which lists all of the components used in the pulley construction and their scrap value.
* The new system will need to be able to output a file containing Bill of Materials information automatically without requiring the user to manually create the file.

##### Integration with Van Gorp’s Macola Enterprise Resource Planning (ERP) System

* Currently, a Van Gorp employee uses an ERP System which keeps track of the parts and components used in the construction of a custom pulley.
* The new system will need to be able to output directly to the ERP System as opposed to requiring an employee to manually enter this program’s output into the ERP System.

## Non-Functional Requirements

##### Performance

* Program executes quickly and easily
* Program will run natively and not require an emulator to run

##### Extendibility

* New Program will be easily extendible by adding additional program features
* System will be able to output in multiple formats (decimals and fractions)

##### Reliability

* New program will not have bugs resulting in faulty output
* Generated output will be correct for all types of pulleys
* Program will recognize invalid pulley configurations and alert the user

##### Maintainability

* Codebase will be easy to read and understand
* All code and design decisions will be well documented
* Engineering standards will be easy to add, remove or change

##### Portability

* Application will be easy to use on all computers within Van Gorp
* Application will be one single executable as opposed to a series of batch files

##### Security

* Engineering standards should not be compromised by the Pulley Print Program
* Pulley Print Program will work within the Van Gorp firewall

## Proposed Solution

Our current proposed solution is as follows. First, we will take the existing GW-BASIC codebase, and begin to do a direct conversion of it in C#. This will most likely be a lengthy process, since the code base is large, undocumented and old. Once this step is finished, we will begin to remove many of the outdated and unused features. Also, while in this step, we will work on updating the engineering standards that are outdated. Once this is complete, the next step is creating the GUI for the application. One of the primary functional requirements is ease use, so we will spend lots of time analyzing the most important features, and placing them in optimal locations in the interface. Furthermore, we will make sure the workflow of the interface is streamline and intuitive. We want the application to come natural to its users, without any need for an initial learning curve. The last step of this process is to change the engineering standards, so that they are not hard coded, but read in from their existing files. Before this can be done, Van Gorp needs to consolidate and standardize the way these standards are stored. Currently, they are written inside of word documents with no exact layout or formatting. Ideally they should be stored in a file that we can parse through with more ease, such as an Excel document.

After we have finished all of this, we will begin the implementation of the secondary functional requirements. These include automation and integration with their ERP system.

#### Strengths

* C# is a widely-used highly maintainable language. Van Gorp uses windows machines, and C# integrates deeply with Windows. Furthermore, Windows Forms will help us create an easy-to-use graphical user interface. Finally, C# also easily connects with SQL Server, which will simplify our integration with their ERP system.
* The ordering of the steps in our solution has many benefits. Changing the engineering standards to be read in from a file versus hard coded is our last step so Van Gorp has time to organize all of their engineering standards. Furthermore, saving the GUI design for later gives us time to get to know the current program. Once we become familiar with the CLI and what features are more used than others, we will be able to design a better interface.

#### Disadvantages

* C# as our language choice limits the platforms we can develop on, as well as what platform it can be executed on. As aforementioned, Van Gorp uses Windows machines. However, if this were to change, our program would not easily, if at all, be able to run.

#### Tradeoffs

* We are not starting the secondary requirements until all of the primary requirements are complete. Although we think this is the best approach, this may make the implementation of them more difficult when the time comes.

## Evaluation of Alternatives

### Different Language Selection

* We chose C# because of its deep integration with Windows, powerful interface design using Windows Forms, and easy execution of SQL. However, C# limits the platforms that our program can be executed on. There are other language choices that are more cross platform, but may not integrate as well with their current work environment, therefore C# is preferable

### Working On Secondary Requirements Earlier

* We are choosing to not begin development on the secondary requirements until we have successfully completed all of the primary requirements. This is in case the primary requirements take longer to complete than we initially plan. We would rather have a complete product with only the primary functions, rather than an incomplete product with parts of both.
* However, this may make the development of the secondary features slightly more difficult. We will most likely have to go back and change code to implement these secondary features. Whereas if we began developing these features from the beginning, the code would already be written to support these features.

## Validation and Acceptance Test

#### Bi-weekly Video Conferencing with Van Gorp

* In order track our team’s progress, we intend to have bi-weekly video conference meetings with a representative from Van Gorp Corporation. This will help to ensure we are producing the right product.

#### Presentation of C# Language choice to Van Gorp

* We plan to present our choice of C# to Van Gorp in order to validate our language choice will work for them in the future.

#### Initial User Interface Prototype

* Once we have all necessary materials transferred from Van Gorp, we plan to develop user interface prototypes to determine if our proposed user interface will work for Van Gorp.

#### Delivery of Product after Initial Refactoring

* Instead of delivering the entire product at once, we intend to deliver the product in stage, following each major milestone. The first of which is the delivery of the newly refactored program.

#### Delivery of Product after each additional feature

* Following each new feature we add to the program, we will deliver the program to the customer to ensure it meets the customer’s expectations.

# Interface/System Description

## Current System State

Currently, the main application to be updated is not user friendly, has outdated code, and cannot be easily updated. Based on information provided by Van Gorp, the backend GW-BASIC code is very jumbled and unorganized. The application uses a Command Line Interface (CLI) that asks the user a series of questions that vary from the part number of the product to the kind of material they will be using. As materials and part numbers have changed over the years, these questions have become outdated. Also, certain equations that Van Gorp uses to calculate pulley dimensions have been updated and improved over the years. All of this information is hard-coded into the current application, making it difficult for Van Gorp to modify. A backup copy of these equations, as well as other engineering tables used to specify standards, is stored in in Microsoft Word Documents. This format allows users to easily read the information, but presents a problem when trying to automatically integrate this information into an updated application.

By the completion of this project, the GW-Basic code will be updated to a modern language. Van Gorp has asked that alongside this update, an easier UI should be created to allow the user to be able to select information quickly. Also, Van Gorp has specified that the application must be maintainable in the future. Van Gorp would like the ability to update current parts in the application as well as the flexibility to change other specifications as needed.

Aside from this application, Van Gorp also uses a piece of software called Exact Mancola for much of their daily activities. This software stores a majority of company information including different purchase and work orders, parts available, inventory, payroll information, and much more. All of the information used in this system is backed using a SQL database. Van Gorp has expressed interest in trying to connect the software project to this Exact Mancola software. By doing this, they could import new parts automatically in the Exact Mancola software instead of having to manually enter the parts.

## Technical Approach

In order to make Van Gorp’s current software more usable, the code will first be updated to a modern language. This modern language will need to meet the requirements specified by Van Gorp, and will require their approval. After the modern language has been approved, the application’s user interface will be changed from a command line user interface to a graphical user interface. The GUI will allow better user interaction with the system. The current CLI asks one question at a time. If the user needs to change an answer to a question, they have to restart the whole program. A GUI will allow multiple questions to be laid out on one or two screens. Also the user will be able to change and edit answers to their questions easily with this new approach.

In order to improve maintainability of the software, an administrative tab will be created to allow Van Gorp to edit various pieces of the software. This will include but is not limited to changing parts in the system, part numbers, and materials used. Additional editing functionality will be determined after the current software has been thoroughly inspected and Van Gorp has determined the amount of flexibility needed. This also includes developing a way for Van Gorp to easily add and modify engineering standards.

Further extensions include modifying Van Gorp’s current engineering tables. These tables are stored in word documents and are not easily accessible by software. These would be converted into an easily accessible file or database format pending Van Gorp’s approval. The final extension that will be considered is connecting this software into their current Exact Mancola software. This software, as stated above, stores a vast majority of company data. It uses a SQL database backend to store all the information. Currently Van Gorp’s software prints out digits on the screen and these numbers are then plugged into their Exact Mancola software. By having the new software directly access the SQL database, these values could automatically be populated into the Exact Mancola software.

## Process Details and Deliverables

This project will be broken up into two Phases. Phase I will consist of updating code, cleanup of code, and creation of the GUI. During Phase I, the current code will be updated to a modern language. As stated earlier, the modern language has not been determined but will be after evaluation of the current state of the software. Based on Van Gorp’s information, the current code is also unorganized. While updating, the code will also be organized by grouping together portions based on their functionality. Also, detailed documentation will be created so that future developers will easily be able to maintain this code.

Phase II will consist of the extensions motioned earlier. The first extension will be the creation of the administrator portion of the application. This will allow Van Gorp to easily modify certain parts of the software. The second extension will consist of moving Van Gorp’s current Microsoft Word Documents over to a more accessible file format. This extension will require research into the most effect way of doing this that can benefit Van Gorp as well as this software project. The final extension of this Phase will consist of integrating into Van Gorp’s Exact Mancola software. It is currently unknown if this is possible. This software stores its information in a SQL database, but research will be performed to see if that database can be accessed and added to without any negative side effects to both pieces of software.

Upon the completion of Phase II and an executable file will be delivered to Van Gorp along with all source code and any documentation either created during development or borrowed from Van Gorp. The executable file can be loaded onto each machine that requires the new software and run locally on the machines.

## Test Plan

Throughout Phase I, black-box and white-box testing will be performed to ensure the backend code is correct. After the GUI is set up, testing will be done to ensure that not only the UI works correctly, but that it is the most effective layout for the users. At the completion of Phase I, Van Gorp will be given a Beta version of the software. This will give Van Gorp the opportunity to make sure the software works exactly the way they need it too. Also, it will help uncover any bugs that were not caught while testing.

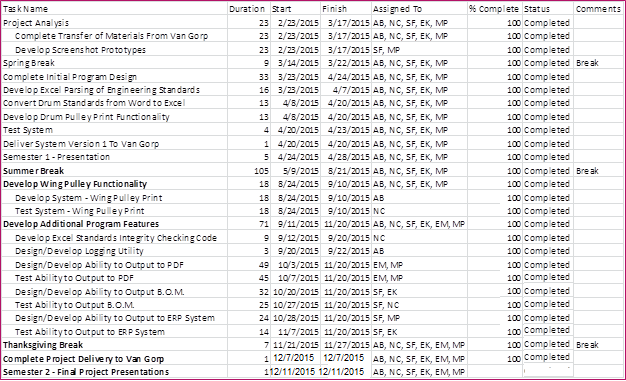
At the beginning of Phase II, all bugs and any requested changes will be made to the software and then tested. The software will again be tested exactly like it is in Phase I. For any parts that require Van Gorp’s servers, on site testing days will be coordinated.

# Work Breakdown Structure

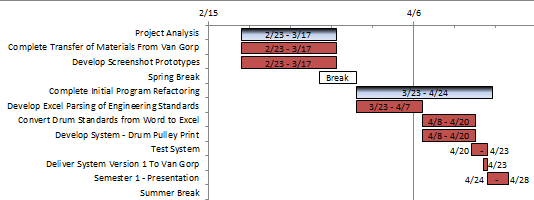
## Project Schedule

We plan to use the following proposed project schedule, though we anticipate changes throughout the course of the project:

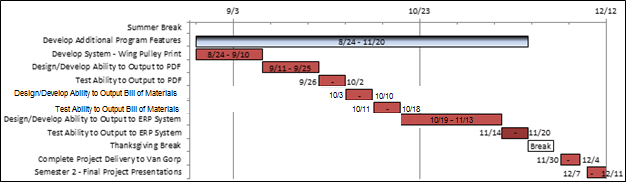
##### Figure 1: Proposed Project Schedule



##### Figure 2: Semester 1 Project Schedule



##### Figure 3: Semester 2 Project Schedule



## Risk/Feasibility Assessment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Probability of occurrence | Criticality | Combined Risk | Mitigation Strategy |
| Running our program using wrong Engineering Standards | Low | Highest | Medium | We will develop an Excel integrity check strategy that ensures the engineering standards are within expected bounds before we calculate the dimensions of a pulley. This code will be run every time the excel document is modified. |
| Our program uses wrong calculations for the dimensions of a pulley component | Low | High | Medium | In addition to the Excel integrity checking, we also plan on using three testing strategies. Firstly, we plan on having 75% unit test coverage minimum for all of our code. Secondly, we plan on having three integration tests for both Wing and Drum Pulleys. Lastly, we plan on using extensive logging to determine where a problem occurred if one exists. |
| Developing a system which does not meet Van Gorp’s requirements. | Medium | Medium | Medium | Throughout our project, we plan on using multiple validation techniques to work with Van Gorp corporation to ensure we are delivering a system which meets and exceeds Van Gorp’s expectations and requirements. |
| Development of the proposed solution falls behind schedule. | Medium | Medium | Medium | We plan to follow an agile method of development by which the system is incrementally built. This will help us evaluate whether or not we are on schedule at all stages of the project. In addition, we plan on front-loading our development schedule. |
| Spaghetti code structure inhibits team’s ability to understand the current system’s architecture. | High | Low | Low | We plan to work with Van Gorp to understand the underlying structure of the system and to understand how the code is broken down into modules. We also plan on developing our own solution as opposed to refactoring the old system. |
| Van Gorp’s Engineering Standards are compromised. | Lowest | High | Low | We plan on hosting Van Gorp’s Engineering standards within Van Gorp’s protected firewall. |

## Cost Considerations

Because our system does not require the purchase of hardware, or new software systems there are no external costs. The three main costs for this project involve development costs in terms of development time spent by the team, and time spent communicating with Van Gorp Corporation through visits to Pella, IA and video conferencing time.

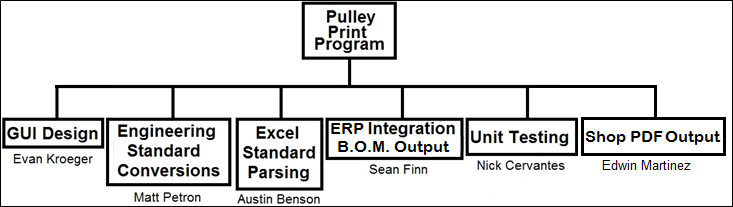
## Work Breakdown Structure

While each team member will take part in all portions of the project, the following roles have been assigned.

* **Team Leader** - Austin Benson
* **Team Communication Lead** - Nick Cervantes
* **Team Webmaster** - Evan Kroeger
* **Key Concept Holder** - Sean Finn
* **Key Concept Holder** - Matt Petron
* **Key Concept Holder** – Edwin Martinez

The work breakdown structure for each part of the system is as follows:

Figure 4: Work Breakdown structure



This breakdown structure will help to ensure our project is modular in design and each team member has a particular portion of the system for which they are in charge of developing.

# Conclusion

This project has very clear goals, and our team has the talent and expertise to produce a quality system for Van Gorp. We are aware of the associated risks, have evaluated possible solutions, and worked to clarify ambiguity in the project plan. Although this project plan provides a roadmap for our team, we understand we will likely need to adapt throughout the project. Only through anticipation of, and responding to, change, we will be successful in reaching our project’s goals.

# Modification Report

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change |
| 1.0 | 2/21/15 | AB, NC, SF, EK, MP | Initial Document |
| 1.1 | 2/23/15 | MP | Included Modification Table |
| 2.0 | 4/3/15 | AB | Updated Project Schedule |
| 2.1 | 4/10/15 | NC | Updated Functional and Non-Functional Requirements |
| 2.2 | 4/11/15 | MP | Updated Work Breakdown Structure |
| 2.3 | 4/13/15 | SF | Fixed Label Formatting |
| 2.4 | 4/17/15 | EK | Added Figure Labels |
| 2.5 | 4/26/15 | AB | Updated Project Schedule |
| 3.0 | 9/14/15 | AB, MP | Project Schedule for Second Semester |
| 3.1 | 9/16/15 | SF | Updated Requirements Section |
| 3.2 | 9/18/15 | MP | Updated Risks Section |
| 3.3 | 9/27/15 | EM | Work Breakdown Structure Update |
| 3.4 | 10/10/15 | NC, EK | Update Project Plan Gantt Chart |
| 3.5 | 10/27/15 | AB, SF | Update Project Schedule |
| 3.6 | 12/9/15 | AB | Final Project Schedule |