

Laboratory Assignment # 5
Andreas Landgrebe
Ian Macmillan
Tristan Challener
Matt Hajduk
Ryan Mong

In this laboratory assignment, each group member has given a different task to be able to complete this assignment at the proper time. In order to complete this assignment at the correct time, that group must split up the work and each group member will need to know what to accomplish in order to complete this assignment at the correct time. If we as a group did not split up the work, then this laboratory assignment will not be completed at the correct time.

Andreas and Ryan were given the task to write the test suites and the requirements document. Andreas and Ryan was acting as the administration and testers to complete this laboratory assignment at the correct time. Andreas and Ryan are able to organize the requirements so the group knows what needs to be accomplished at the end. As Andreas and Ryan are acting as testers, we are able to test the full system to make that it can large instances of the next release problem at the shortest amount of time.

Ian was given the task to learn and use the JCommander tool to make sure that all of the command-line arguments are recognized and verified to be able to efficiently solve large instances of the next release problem. Ian is acting as the tool smith. As Ian was acting as the tool smith, he is able to use this new tool in order to parse command line parameters in order to recognize and verify these arguments.

Matt and Tristan acted as co-pilots to ensure that this assignment is complete properly and correctly. They will be working on the working methods.

1. Introduction to the Document

1.1 Purpose of the product

The purpose of this product is to provide the user with decisions about when to release an application with the proper amount of requirements that should be outlined. This product will provide a particular user what requirements should be implemented in the next release of a program or system.

1.2 Scope of the Product

The scope of this product would be to provide the user with the available requirements in order to successfully toe next release planner. By doing this, the user will be able to easily be able to implement the proper amount of requirements before it goes over the certain amount of cost of a product. This is essential in order to successfully provide customer with a product that one would be implemented.

1.3 Acronyms, Abbreviations, Definitions

The only definitions that needs to be set is that the SRS stands for the software requirement specifications.

1.4 References

Some of the references that we had used in order to implement this laboratory assignment successfully was using a new new software environment called JCommander. In order to learn to use this tool successfully we had to use the website <http://jcommander.org/>.

1.5 Outline of the rest of the SRS

- 1.1 Purpose of the product
- 1.2 Scope of the product
- 1.3 Acroyms, Abbreviations, Definitions,
- 1.4 References
- 2 General Description of Product
 - 2.1 Context of Product
 - 2.2 Product Functions
 - 2.3 User Characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and Dependencies
- 3 Specific Requirements
 - 3.1 External Interface Requirements
 - 3.1.1 User Interfaces
 - 3.1.2 Hardware Interfaces
 - 3.1.3 Software Interfaces
 - 3.1.4 Communications Interfaces
 - 3.2 Functional Requirements
 - 3.2.1. Class 1
 - 3.2.2 Class 2
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Quality Requirements

2. General Description of the Product

2.1 Context of Product

The context of the product would be to have the requirement as R, Benefits as B, and Cost as C. This values will be put in as input and the output will be able to show that as the Cost is a factor, it will show the amount of requirements that one is able to implement to have the most amount of benefit without exceeding the cost for the product.

2.2 Product Functions

The functions of this product is to be able to provide the output of the available requirements that one is able to use with achieving the most amount of benefit without exceeding the cost of a product

2.3 User Characteristics

Some of the user characteristics that is essentially to have for this product would be to be able to have the user to have a readable output when one is input the components of a particular product.

2.4 Constraints

Some of the constraints that we had faced during this assignment was that we must be using integer values through the command line. Unfortunately, we are unable to compute any of the other primitive or complex data types, it must be an integer.

2.5 Assumptions and Dependencies

Some of the assumptions that is being made in this product is that the cost of the product is always going to be greater than zero. This may not be likely, but the cost of one of the requirements might be zero. Another assumption that could be made would be the cost of each of the requirements might be the same. Each of the requirements will have a different cost then any of the other requirements. Due to this, managers will need to get an educated decision on which of the requirements should be implemented in the next release planner.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Some of the user interfaces of this product will be available to provide managers access to easier steps to make decisions about what requirements to release in the next release planner. Due to this, it will also provide any users to gain easier access to any of the other of the other products that these manages are trying to implement. The input will be displayed as this:

CSV File Format:

Req # \$Cost \$ Benefit ReqString

Cost Benefit [-C \$total cost] -file

Cost Benefit -C[\$total cost] -r [Req #, Cost #, Benefit, "ReqStr,"

The Output of this will be as follows:

Total Cost: \$ %d

Elapse time: %d

Estimated total cost: \$ %d

Estimated total benefit: \$ %d

Remaining flex budget: \$ %d

Profit \$ %d

List of Includes: %d " "%s"

List of Discarded %d "%s"

3.1.2 Hardware Interfaces

This view will allow the user to view the operational status of each available requirements that one is able to implement after accounting into effect the cost of a particular product to effectively find the most amount of benefit.

3.1.3 Software Interfaces

The software interfaces for this product will be similar to the user interface of how the input and output will be displayed.

3.1.4 Communications Interfaces

The communications interface of the product is how the java main classes that have been written will show how much coverage there is by the test cases that is being written. This interface is essential in order for the product to be successfully implemented.

3.2 Functional Requirements

The functional requirements for this laboratory assignment would be to be able to list the inputs of the requirements, benefit and cost. As output, the product will be able to product as output the elapsed time, and the requirements, benefits and cost of the product.

3.2.1 Class 1 - ReleasePlanner

As part of the inputs for this Laboratory assignment, we have created a main class called ReleasePlanner. In this class, we have created a method called nextRelease that will have 4 parameters of List R, List B, List C, and int w. As input, we are using each list for a different task. For List R, we have assigned that to be the set of requirements. For List B, we have assigned that to be the benefit. For List C, we have assigned that to be the cost. For List C, we have arranged the method to be so that for a new List P per C, we have assigned that to be the list of profits which will have the list of requirements for list C to be subtracted from the list of requirements for list B. After this, we have had the int w to be assignment to do total weight allowed.

3.2.2 Class 2 - parameters

For the second class that we have, we have created a class called parameters. In this class, we have the input for the user interface to be able to display the correct output that has been outlined.

3.3 Performance Requirements

Some of the requirements for the performance include that the product needs to be as quick as possible. This product needs to be able to read in these variable and the elapse time needs to be as little as possible. In order for our product to compete with the rest of the class, it needs to be able to have an elapse time to be as little as possible to show that we have found out a way to have create this product to perform in as little time as possible.

3.4 Design Constraints

Some of the design constraints that are in effect for the product are to have the product to have the most amount of benefit. When we have set the input of the requirements, benefit and the cost, we have set some of the constraints for the product. For example, we have set a constraint that if the cost variable exceeds the benefits for the list of requirements, this will not be displayed in the output for all the inputs given.

3.5 Quality Requirements

The requirements for quality are that our product needs to be able to estimate the output of the inputs that were set for the requirements, benefits and cost of a particular product. These inputs needs to be able to display the correct data for the manager that is planning to the next release planner.