|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | | | | |
|  | |  | | | |
| Project: | | ServeMe System (SMS)  CSE 5325 – Spring 2022  Project Management | | | |
| Module: | | COCOMO | | | |
| Deliverable: | | COCOMO Estimate Report | | | |
| Version: | | | 1.0 | Date: | 04/12/2022 |

Prepared by: Srajan Ahuja, 1001858728

TABLE OF CONTENTS

[1. Introduction 2](#_Toc23340379)

[2. Estimating Factors 3](#_Toc23340380)

[2.1 Source of Lines of Code 3](#_Toc23340381)

[2.2 Scale Drivers 3](#_Toc23340382)

[2.3 Cost Drivers 3](#_Toc23340383)

[3 Project Final Timeline and Cost Structure 4](#_Toc23340384)

[4. Conclusion and Recommendations 5](#_Toc23340385)

[Appendices 6](#_Toc23340386)

# 1. Introduction

The purpose of this document is to give an estimate for the cost for our project, ServeMe System (SMS) using the cost estimation software tool which uses the COCOMO Model. Planning a project with their cost estimates is a tricky part in a software development life cycle. With the help of cost estimating tools and cost estimating models, we can get the duration to complete the project, the effort required in person-months, and the total cost.

ServeMe System is a web, android, IOS based application which allows the users of the application to request for services which in turn can be accepted and quoted for prices and then the user can go ahead and choose one of the services provided from the quotes provided.

The document includes an overview of several features such as the number of Source Lines of Code (SLOC) necessary to code the complete program, Scale Drivers, which provides a thorough picture of the project's scale, and Cost Drivers, which provides an estimate of the project's effort and length. The elements listed above assist the PM in planning each phase of the project based on the model chosen for the project (Incremental Model), navigating each step based on the estimate and expense assigned for them, and moving on to the next step.

At the conclusion of the estimation, it will assist the Project Team in gaining a clear understanding of the most efficient way to design the complete project by utilizing resources and efficiently assigning cost to each of them all at the same time. not sacrificing the final product's quality in pursuit of a bigger profit.

Based on the originally planned and estimated project budget/duration and a comparison to the current COCOMO estimation, a recommendation on the most efficient way to plan, including where to make changes to the planning activities to ensure that the final product is of high quality without compromising any factors.

# 2. Estimating Factors

## 2.1 Source of Lines of Code

The following is the number of lines of code delivered as part of this project, A justification for the total amount of LOC is provided.

|  |  |
| --- | --- |
| **SLOC | Source Lines of Code** | Value Chosen: 5000 |
| Justification: For ServeMe system the code will be greater than 5000 SLOC but with the help of code optimization and reusability of components we can lower it down to 5000 SLOC. We will be building components and reuse them which will lower the source line of code by a huge amount. | |

## 2.2 Scale Drivers

The following is the list of scale drivers, the values applicable to this project and a justification for each value chosen:

|  |  |
| --- | --- |
| **PREC | Precendentedness** | Value Chosen: Very High |
| Justification: Precendentedness is set to very high as our team is familiar with the development environment of this project. We have done similar projects previously so it will not be very challenging. | |

|  |  |
| --- | --- |
| **FLEX | Development Flexibility** | Value Chosen: Low |
| Justification: The development conditions are very firm and need approval for every modification to the plan so development flexibility is set to low. | |

|  |  |
| --- | --- |
| **RESL | Architecture / Risk Resolution** | Value Chosen: High |
| Justification: The work and data flow of the project was very well explained while gathering the project requirements by the customer so the Architecture/ Risk Resolution is set to High. | |

|  |  |
| --- | --- |
| **TEAM | Team Cohesion** | Value Chosen: Very High |
| Justification: Team cohesion is set to very high as the team has worked in this environment before and there is a minimal chance of conflict. Team work will be seen as a positive aspect in this project | |

|  |  |
| --- | --- |
| **PMAT | Process Maturity** | Value Chosen: High |
| Justification: This project's process maturity is high, as the team has a greater level of maturity than the nominal norm. The procedure is designed in such a way that the project will go smoothly and without major stumbling blocks. | |

## 2.3 Cost Drivers

The following is the list of cost drivers, the values applicable to this project and a justification for each value chosen:

**PERSONNEL COST DRIVERS:**

|  |  |
| --- | --- |
| **ACAP | Analyst Capability** | Value Chosen: High |
| Justification: This project's analysts are very capable and have experience with similar projects, so they know what they're doing. They are familiar with the environment, as well as the languages and tools. | |

|  |  |
| --- | --- |
| **APEX | Application Experience** | Value Chosen: High |
| Justification: Because our team has been working on these types of projects and apps for the past three years, I'll rate this cost driver as 'high.' The members of the team are more than capable of handling the application's development. | |

|  |  |
| --- | --- |
| **PCAP | Programmer Capability** | Value Chosen: Very High |
| Justification: This is the best squad I've ever had. The programmers are qualified and dependable enough to complete this project without severe complications or obstacles. The team understands the strategy and how to put it into action within the stated time frame. | |

|  |  |
| --- | --- |
| **PLEX | Platform Experience** | Value Chosen: High |
| Justification: The team has three years of experience with the desired platform, which happens to be react. React allows us to create both a website and an Android application using the same components, reducing the number of lines of code required for two separate projects. As a result, the team prefers to use react because they have previous expertise with it. | |

|  |  |
| --- | --- |
| **LTEX | Language and Tool Experience** | Value Chosen: High |
| Justification: The tool used is Visual Studio Code, and the language utilized is JavaScript. The team has been doing this for three years and has a lot of experience with it. | |

|  |  |
| --- | --- |
| **PCON | Personnel Continuity** | Value Chosen: Very High |
| Justification: Our company's annual turnover rate is planned to be extremely high, at only 3% or less per year. There are no plans to cut costs and lay off employees as a firm. Also, the squad is content with where they are right now and has signed a three-year commitment with us. | |

**PROJECT COST DRIVER:**

|  |  |
| --- | --- |
| **TOOL | Use of Software Tools** | Value Chosen: High |
| Justification: This project's tools are durable and dependable. We have a very tight timetable and don't want to cut corners on tools, which is why we chose highly efficient and best-in-class instruments that would also delight the consumer. | |

|  |  |
| --- | --- |
| **SITE | Multisite Development** | Value Chosen: Very High |
| Justification: Because the team works in the same building and complex, they are better able to communicate and collaborate on the project. They have video calls on occasion, therefore this cost driver is set at a very high value. | |

|  |  |
| --- | --- |
| **SCED | Required Development Schedule** | Value Chosen: Very Low |
| Justification: This project's timetable corresponds to the plan stated in project 2, and we will work 75 percent of the time according to the nominal plan as mentioned in the second phase. There will be small adjustments to the plans, but no major alterations are anticipated. | |

**PLATFORM COST DRIVERS:**

|  |  |
| --- | --- |
| **TIME | Execution and Time Constraint** | Value Chosen: Nominal |
| Justification: The CPU time will be minimal, typically less than or equal to 50% of the available execution time. | |

|  |  |
| --- | --- |
| **STOR | Main Storage Constraint** | Value Chosen: Nominal |
| Justification: Because this is a website and an Android application, the storage cost will be less than or equal to 50% of the available storage space. | |

|  |  |
| --- | --- |
| **PVOL | Platform Volatility** | Value Chosen: Low |
| Justification: Every month, there will be a minor database or operating system change, and every year, there will be a big operating system modification to keep up with the latest technology and tools. | |

**PRODUCT COST DRIVERS:**

|  |  |
| --- | --- |
| **RELY | Required Software Reliability** | Value Chosen: High |
| Justification: We will suffer a significant financial loss if the software fails. We can't afford any damage to our project because it's entirely software-based. We will suffer a significant financial loss if the software fails. If software fails, it will result in a broken application and website, costing a lot of money. | |

|  |  |
| --- | --- |
| **DATA | Database size** | Value Chosen: Low |
| Justification: Because our application is a meal ordering application, it does not store a lot of data and uses very little data, hence the database size/SLOC will be less than 10. The value chosen is Low since we will just be storing order details and profile data. | |

|  |  |
| --- | --- |
| **CPLX | Software Product Complexity** | Value Chosen: Nominal |
| Justification: This program is simple to use and does not require any sophisticated statements or code. Only nested code, standard math procedures, and many files, graphics, and stylesheets will be present. Nothing too difficult. | |

|  |  |
| --- | --- |
| **RUSE | Required Reusability** | Value Chosen: Low |
| Justification: As this isn't a large-scale project, we won't be able to reuse our components between projects or platforms. Components of food ordering applications cannot be reused across projects or platforms. We can use very small components in one project, but not in other projects. If we wanted to use it, we'd have to increase our budget to cover the cost of copyright for those components. | |

|  |  |
| --- | --- |
| **DOCU | Document Match to Lifecycle Needs** | Value Chosen: Nominal |
| Justification: The documentation will be brief and to-the-point. We don't think expending so much resource, time, and money on documentation is necessary with such a tight deadline. As a result, documentation will be tailored to Agile requirements. | |

# 3 Project Final Timeline and Cost Structure

In this section we will see our previously calculated cost for the project based on our estimates and the requirements we had gathered. As we have used a cost estimate tool for this project, we will now compare it with the previously calculated cost.

Table

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

After Using COCOMO :

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Cost (K$) | $ 53.7K |
| Non-Human Resources Cost | $292,400 |
| Effort in Person-Months | 5.7 |
| Duration in Months | 4.8 |

Previously estimated cost = $797,970

The new estimated total cost = 346,100

So the parameters will now change to:

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Net Profit Margin: | More than 50% (50-55%) |
| Net Profit: | $451,870.00 |
| Profit Percentage: | 56.6% |

# 4. Conclusion and Recommendations

To conclude, in the previous documentations we had only considered the requirements and the human and non-human costs for the project. But after using COCOMO II we can see there are significant other aspects to the development phase and will be costing more than what we had expected. The duration will also be more than 3 months and we will not be able to deliver the project in 3 months even if we utilize maximum of the resources we have, be it human resources or non-human.

SLOC and Estimating Model were two elements that were not considered for Microsoft Project Plan. The SLOC defines the size of the project's code, whereas the model determines the project's estimating approach. These were considered as part of the COCOMO method's estimate. CPU execution time, storage capacity, amount of the data trained for the database, platform utilized, skill of the platform programmers, and other aspects were not included in our MPP calculation.

All the above aspects were taken into account in the COCOMO tool calculation, resulting in an accurate estimate of the project's length and cost. Some of these elements were on the upper end of the scale, resulting in a larger cost and a longer time frame at the conclusion of the estimation.

My proposal is to proceed with the project work flow based on COCOMO's anticipated cost and time, which is far more accurate than the MPP. Even though the COCOMO technique estimates a larger cost and duration for the project than the MPP method, the COCOMO method is the preferable alternative since it considers all scale and cost variables.

When it comes to the project's duration, almost 5 months appears to be a longer period. As a result, I would advise the board of directors to consider adding more resources to the team in order to ensure that the work is completed faster and the overall duration of the project is reduced. The sooner the entire project is completed, the sooner we will be able to benefit from all of the application's features.

# Appendices

Graphical user interface, application

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated