

**Project: Dealership Management System**

**CSE 5325 – Fall 2019**

**Project Management**

**Module: COCOMO**

**Deliverable: COCOMO Estimate Report**

**Version: [1.0]**

**Date: [11-07-2019]**

# TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>2</b>
<b>2. ESTIMATING FACTORS .....</b>	<b>3</b>
2.1 Source of Lines of Code.....	3
2.2 Scale Drivers .....	4
2.3 Cost Drivers.....	5
<b>3 PROJECT FINAL TIMELINE AND COST STRUCTURE.....</b>	<b>8</b>
<b>4. CONCLUSION AND RECOMMENDATIONS.....</b>	<b>12</b>
<b>APPENDICES.....</b>	<b>14</b>

# 1. Introduction

---

## Top Level Summary:

- The web application which needs to be developed will be known as dealership management system. This web application would be used by customers as well as employees. The objective of the application is to give a platform (customer specific) to provide information about the company and inventory of the products available and to give a platform (employee specific) to register as new users and also enabling them to view more confidential data such as sales records and customer information. It would also contain feature to allow employees to keep personal notes and an option to search for relevant data. This application would ensure that our client's customer acquisition cost is reduced and provides an effective platform to promote their company.
- A rough estimation of the schedule and cost of development of the project was provided through Microsoft Project Planner tool and here we calculate the same through COCOMO tool.

## Purpose of this document:

- To learn how COCOMO II tool lets you make trade-offs and experiment with "what-if" analyses to arrive at a satisfactory project plan using certain drivers: Cost Driver and Scale Drivers.
- In this document, we estimate the total development cost and the total duration of our project, *dealership management system* through COCOMO II tool. We also provide the various estimating factors (SLOC, Cost Drivers, Scale Drivers) and justification of using a particular value for each of them. We also compare the two ways we made use of, to assign the schedule and calculate the costs attached with each phase and the final cost.
- We realize that there is a difference in the cost (and duration) estimation done by us through project planner and the estimation done by COCOMO. The new schedule tells us that this project requires more duration (around 7.3 months) as compared to our schedule.
- To realize that Microsoft Project planner's estimate gives the total cost of the project (without any profits attached) as **\$312690** while COCOMO calculates the total cost of the project (without any profits attached) as **\$383720**, providing a difference as **\$71,030** **\$575580** is the final cost produced (inclusive of the profits) by COCOMO. While, **\$469035** is the final cost produced (inclusive of the profits) by project planner, providing a difference of **\$106545**.
- We provide the probable reasons (SLOC, Cost drivers and Scale Drivers) for such (above mentioned) differences in the figures in this document and the recommendations following those reasons.
- Recommendations Include Using Agile Approach, Comprehensive testing strategy and allowing the processes to work in parallel and add more developers to the team.

## 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	5000 SLOC
<p>Using <b>View page source</b> for a dealership website (Auto nation) as a reference, I made following observations:</p> <ol style="list-style-type: none"><li>1. Considering that the Welcome page would consist of the Search/View Inventory tab and login tab and notes – This would amount to approx. 5000 SLOC</li><li>2. Registration page would nearly amount to around 1500 SLOC.</li><li>3. Customer Feedback page would amount to approx. 500 SLOC.</li><li>4. CRUD Customer and CRUD Inventory pages (together) would give us around 2000 SLOC.</li><li>5. Sales Record page shall give around 3000 SLOC.</li></ol> <p>TOTAL = Approx. 12,000 SLOC. This is just the SLOC for the front-end coding. There would be more SLOC for back end coding as well as for Service layer as well.</p> <p>But Since, the COCOMO software module which we are using for cost estimation provides a maximum limit of 5000 SLOC, I have taken SLOC as 5000.</p>	

## 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:

<b>PREC   Precedentedness</b>	High: Generally familiar
<b>Justification:</b> Considering that the development team is skilled in Web development and related technologies (As mentioned in the <b>assumptions section</b> ) like HTML, JavaScript and CSS, precededentedness can be set as <b>High</b> i.e. generally familiar, but since this is an entirely new project, there is a possibility that the developers would be implementing some technical concepts for the first time for example, also mentioned in the requirements given by our client that : Web application should hosted on the company's own web server present in their data center, for which the development team might have to find a way to work around which they might not have done in their previous projects.	
<b>FLEX   Development Flexibility</b>	Nominal: Some Relaxation
<b>Justification:</b> It reflects the need for conformance with pre-established and external interface requirements, and a possible premium on early completion. It answers the questions "Can you relax some of the requirements?" I have kept this as "Some Relaxation", assuming the client is firm with their requirements but also considering some space for possibility of changes in future and being on safer side.	
<b>RESL   Architecture/ Risk Resolution</b>	High: Generally (75%)
<b>Justification:</b> Little level of uncertainty in key architecture drivers whether it is the hardware or the technology to be used. Mostly all the associated risk items have already been identified. Tool support is available for developing and verifying architectural specifications. But little knowledge of tool support available for resolving risk items and little or no knowledge of how schedule and budget are compatible with risk management plan.	
<b>TEAM   Team Cohesion</b>	Very High: Highly cooperative
<b>Justification:</b> Experience of working in the team can be evaluated as very high considering that they have worked together in similar projects before. Teambuilding to achieve shared vision and commitments is also considerable. Cannot be evaluated as Extra High considering other points like consistency in the objectives and cultures of the each of the member and ability and willingness of each of the member to accommodate other member's objective, which otherwise would be an ideal case.	
<b>PMAT   Process Maturity</b>	Very High: Managed
<b>Justification:</b> Processes are standardized as well as the performance of processes is controlled using statistical and other quantitative techniques and is quantitatively predictable.	

## 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:

<b>ACAP   Analyst Capability Cost Driver</b>	Very High: 90 <sup>th</sup> Percentile	
<b>Justification:</b> It measures the capability to measure the analyst to know how well the analyst can understand and lay down the requirements of the project. Since our team holds developers who make the best team as already mentioned in the assumptions part of first assignment, I have put it as Very High.		1. It has the capability to measure the analyst so as to know how well the analyst can understand and lay down the requirements of the project. Since our team holds developers who will participate in the requirements phase and lay down the requirements thoroughly, I kept the analyst capability as Nominal.

<b>APEX   Applications Experience Cost Driver</b>	High: 3 years	
<b>Justification:</b> It measures the experience of the team has on this type of application. Since a dealership company (which has been expanding greatly over the past 10 years) is hiring this software development company to release their webpage, It wouldn't be wrong to guess that the dealership would place their trust in somebody who has an experience of at-least 3 years working with similar applications.		

<b>PCAP   Programmer Capability Cost Driver</b>	Very High: 90 <sup>th</sup> Percentile	
---	--	--

**Justification:** It measures the capability of the programmers for the project. Since our team members have significant work experience, As already mentioned in the assumptions sections.

<b>PLEX   Platform Experience Cost Driver</b>	Nominal: 1 year
<b>Justification:</b> As mentioned above, the team may have been working on HTML, CSS, JavaScript for at-least 3 years, <b>but</b> may have little knowledge of working on authentication technologies or resource redundancy or how to handle 500 simultaneous requests at the same time.	

<b>LTEX   Language and Tool Experience</b>	High: 3 years
<b>Justification:</b> Measures the experience, the team has with language and tools. With 3 years of working on similar technologies, we can expect 3 years of language and tool experience of working on advanced technologies like Angular JS and Bootstrap.	

<b>PCON   Personnel Continuity Cost Driver</b>	Nominal – 12% per year
<b>Justification:</b> It measures project's annual personnel turnover with respect to employees. Since some have already been working with the company for some time now, and some might leave the company, I have put the value for PCON cost driver as Nominal.	

<b>TOOL   Use of Software tools Cost Driver</b>	High: Strong, mature, life-cycle tools, moderately integrated
<b>Justification:</b> It measures the number based on the complexity of the tools that the team will use for the development of the project. Since there <b>might</b> be some opportunity to use slightly complicated tools (either in developing or testing), I am being safe by marking this cost driver as High.	

<b>SITE   Multisite Development Cost Driver</b>	Very High: Same building or complex. Occasional video conferencing
<b>Justification:</b> Assuming that the software development company is only 3 years old, it's safe to assume that the company doesn't have branches outside the city and all the team members are contained inside a same complex/ building and communicate with each other through occasional video conferencing, which is a general case.	

<b>SCED   Development Schedule Cost Driver</b>	Nominal: 100% of nominal schedule
<b>Justification:</b> It is a number that is measure of the compressed schedule from that of the default schedule. It's the schedule constraint imposed on the team. Since, we have the constraint of completing this application in 3 months, I have marked this cost driver as Nominal.	

<b>TIME   Execution Time Constraint Cost Driver</b>	Nominal: <=50% use of available execution time
<b>Justification:</b> The rating is expressed in terms of the percentage of available execution time expected to be used by the system or subsystem consuming the execution time resource. It is a measure depending upon the CPU time that the software will use. It is a	

web application and will not have too many processes that will consume CPU power and hence nominal.
---

<b>STOR   Main Storage Constraint Cost Driver</b>	Nominal: <=50% use of available storage
<b>Justification:</b> It is the measure depending upon the main memory that the software will use. It will consume memory <b>primarily</b> because of the GUI, and that's why I have marked it as Nominal.	

<b>PVOL   Platform Volatility Cost Driver</b>	Low: Major change every 12 months; minor change every month
<b>Justification:</b> It is a measure based on how often the platform changes due to updates or any other reason. Since the platform that we are working with will not change much, hence low.	

<b>RELY   Required Reliability Cost Driver</b>	Nominal: Moderate, easily recoverable losses
<b>Justification:</b> It is a measure based upon the consequences of the software failure. Software failure won't pose as a risk to human life or lead to huge financial losses. Hence, marked it as Nominal.	

<b>DATA   Database Size Cost Driver</b>	High: $100 \leq (\text{Database bytes} / \text{SLOC}) < 1000$
<b>Justification:</b> DATA is a measure of data required to test the functionality of the software. Since there should be enough test cases to ensure proper working of the code, I have marked it as high.	

<b>CPLX   Product Complexity Cost Driver</b>	High: Complex Code, Numerical Analysis, Physical I/O
<b>Justification:</b> It is a measure based on the complexity of the software being developed. Since our system will have nested codes, multiple files, attractive GUI and standard math routines, I have kept product complexity as High.	

<b>RUSE   Required Reusability Cost Driver</b>	Nominal: Across Project
<b>Justification:</b> It is a measure which is calculated based on the software components that are to be reused. The team would be re-using the components that can only be used for other web applications with the specific concepts used in this project, I have put this cost driver as Nominal.	

<b>DOCU   Documentation match to life cycle needs</b>	Nominal: Right Sized to life cycle needs
<b>Justification:</b> It is a measure which is largely dependent upon the documentation that we are creating in the product development. There can be similar upcoming projects in the future on which the team might work on and fair chances of the upcoming projects not being similar to this one too, So I have put it as Nominal.	



## Project Final Timeline and Cost Structure

---

- **Previous Cost, Work and Duration:**

PHASE	DURATION	WORK (Person-Months)	COST (PER PERSON MONTH)
Requirements	12 days	576 hours	\$33,600
Design	13 days	624 hours	\$36,400
Implementation	20 days	808 hours	\$60,870
Testing	16 days	720 hours	\$42,000
Deployment	6 days	288 hours	\$16,800
Total	67 days	3016 hours	\$189,670

**Please Note:** Work (Person Months) is calculated as: Number of developers \* Number of days they work on a task \* Number of hours they work in a day.

### HUMAN RESOURCES COST FOR REFERENCE (From Assignment 1)

Resources	Salary/hour	Total salary for total project duration
Manager	\$100	$100 * 520 = \$52,000$
Developer 1	\$50	$50 * 520 = \$26,000$
Developer 2	\$50	$50 * 520 = \$26,000$
Developer 3	\$50	$50 * 520 = \$26,000$
Developer 4	\$50	$50 * 520 = \$26,000$
Developer 5	\$50	$50 * 520 = \$26,000$
Total		\$182,000

### New Schedule (Duration):

**Please Note: In COCOMO:** I have broken down my DESIGN phase from Microsoft project planner into 2 segments – Product design and Detailed Design. And I have integrated my TESTING and Deployment phases together in Integration and Test segment of COCOMO.

TOTALS FOR ENTIRE PROJECT	DURATION (Mo)
Requirements	1.1 months
Development (Product Design + Detailed Design + Code and Unit Testing + Integration and Testing)	6.3 months
Total	7.3 months

- **COCOMO estimated costs**

**Human Resources Cost**

TOTALS FOR ENTIRE PROJECT	COST in K\$ (PER PERSON MONTH)
Requirements	13.9
Development (Product Design + Detailed Design + Code and Unit Testing + Integration and Testing)	246.8
Total	260.7

- **Non-Human Resources:**

Laptops – HP Zbook studio X360	6*2400=\$14400
Database and web server provided by client	Free
McAfee Security Software	6*20=\$120
Microsoft Office	6*50=\$300
Selenium testing tool	Free
Visual Studio	6*1200=\$7200
IBM Server (for client's data center)	\$10,000
Operating system for server - CentOS	Free
Total	\$32020

- **Other Costs:** \$91,000 (Insurance, Utilities, building costs)

*Profit:*

Human resources cost = \$260,700 (According to COCOMO estimation)

Non-Human resources cost = \$(32,020 + 91,000) (from above)

Total Cost = Human resources cost + Non-human resources cost = \$383720

- *Maintaining a Profit Margin of 50%:*

$$\$383720 * 0.5 = \textbf{\$191,860 Profit}$$

*Total cost:*

- *Final Total Cost of the Project including the Profit =*

$$\$191860 + \$383720 = \textbf{\$575,580 Final Cost of the Project including Profit}$$

- **Microsoft project planner's cost**

Human Resources Cost = \$189670

Non-human resources Cost = 32020+91000 = \$123020

Total (human + Non-human) = \$312690

- *Maintaining a Profit Margin of 50%:*

$$\$312690 * 0.5 = \textbf{\$156345 Profit}$$

- *Final Total Cost of the Project including the Profit =*

$$\$156345 + \$312690 = \textbf{\$469035 Final cost of the Project including Profit}$$

**Comparing the two (Project Planner Estimation and COCOMO Estimation):**

*Difference in the total costs (without adding profits):*  $\$383720 - \$312690 =$   
**\$71,030**

*Difference in the final total cost including the profit:*  $\$575580 - \$469035 =$   
**\$106545**

## 4. Conclusion and Recommendations

### CONCLUSIONS

1. A quick Summary of the **DIFFERENCES** in the figures (cost):

Microsoft Project planner's estimate gives the total cost of the project (without any profits attached) as **\$312690** while COCOMO calculates the total cost of the project (without any profits attached) as **\$383720**, providing a difference as **\$71,030**. **\$575580** is the final cost produced (inclusive of the profits) by COCOMO. While, **\$469035** is the final cost produced (inclusive of the profits) by project planner, providing a difference of **\$106545**.

2. Through COCOMO report, we observe that our project duration is also exceeding 3 months.
3. Probable Reasons for such deviations in the figures:
  - One of the most obvious reasons can be limitation of the COCOMO tool in putting SLOC more than 5000. 5000 SLOC is the maximum limit. Although our project approximates the total SLOC as more than that, around 12,000 for just front-end coding.
  - COCOMO considers several other factors which affects the duration and cost attached with the development of a project. Few of these factors can be briefly summarized as:
    - a. Skills of the team members, familiarity of using the tools and languages to be used, team cohesion, availability of risk resolution tools etc.
    - b. Experience of the team on a given platform, estimated complexity of the code differing from the actual complexity of the code etc.

## RECOMMENDATION

1. In the COCOMO Estimate, the project would be finished in 7.3 months, whereas we are required to finish the project in approximately 3 months. Therefore, we need to:
  - **Develop a comprehensive testing strategy** –We are using a lot of different types of testing for our project: system testing, user testing, integrated system testing, integrated user acceptance testing. In any project testing is supposed to take the maximum amount of time and effort. The goal of developing comprehensive testing strategy is to ensure that there are no gaps or overlaps.
  - **Eliminate False Dependencies** - Every task on a project plan must have a predecessor and successor. The first task must complete before the second can begin. In reality, much work is conducted in parallel and can be expressed as finish-to-finish or lagged finish-to-start tasks.
  - **Agile Software Development Approach**. It means cooperation of stakeholders on all stages of software development from beginning to end. The developers can show the product to the customer at each stage to gain reviews and improvise the product based on that. In such a manner, the risk of spending more in reworking (Unlike in Waterfall) is reduced. **Therefore, Continuing in Waterfall method wouldn't provide us much benefits, if we are considering reducing time and cost of the project as a prior thing.**
  - We can also add a greater number of developers to this project and make the team a bit larger.

## Appendices

### TOTALS FOR ENTIRE PROJECT – EFFORT, DURATION, COST AND PRODUCTIVITY

SystemStar - dms (Component1)

File View Reports Components Tools Preferences Monte Carlo Help

Estimate: dms ID: Model: COCOMO II 201

Component: Component1 ID: Increment: 1

ACT ARC CBR CDF CDR CMP CST DET EBR EFF EQS GCS GMI GST IDT ISM MSZ NAM PDF RSK SCH SIZ SSM STF

Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K\$)	Productivity	Equivalent Size
Requirements	RQ:	0.4	1.1	13.9		Total Size: 5,000
Development	PD+DD+CT+IT:	5.9	6.3	246.8	848.5	
Total	RQ+PD+DD+CT+IT:	6.3	7.3	260.7	793.0	

Costs for Component: Component1

Cost per Person-Month

Requirements	\$ 33600	<input type="checkbox"/> Inherit RQ	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Product Design	\$ 18200	<input type="checkbox"/> Inherit PD	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Detailed Design	\$ 18200	<input type="checkbox"/> Inherit DD	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Code & Unit Test	\$ 60870	<input type="checkbox"/> Inherit CT	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Integration & Test	\$ 58800	<input type="checkbox"/> Inherit IT	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Maintenance	\$ 0	<input type="checkbox"/> Inherit MN	<input type="checkbox"/> Use Rates Tab & Labor Distribution

### SCALE DRIVERS AND COST DRIVERS

SystemStar - dms (Component1)

File View Reports Components Tools Preferences Monte Carlo Help

Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K\$)	Productivity	Equivalent Size
Requirements	RQ:	0.4	1.1	13.9		Total Size: 5,000
Development	PD+DD+CT+IT:	5.9	6.3	246.8	848.5	
Total	RQ+PD+DD+CT+IT:	6.3	7.3	260.7	793.0	

COCOMO II Cost Drivers for Component: Component1

<b>Personnel</b>	<b>Platform</b>	<b>Product</b>
ACAP... Very High	TIME... Nominal	RELY... Nominal
APEX... High	STOR... Nominal	DATA... High
PCAP... Very High	PVOL... Low	CPLX... High
PLEX... Nominal		RUSE... Nominal
LTEX... High		DOCU... Nominal
PCON... Nominal		
<b>Project</b>	<b>Size Summary</b>	<b>User Defined</b>
TOOL... High	Size: 5000	USR1... Undefined
SITE... Very High	Method: SLOC	USR2... Undefined
SCED... Nominal		USR3... Undefined
		USR4... Undefined

Drivers & Size / Model / REVL / Reuse / Function Points / Increments / Breakage / Costs / Rates / Maint. / Filter / Descr. /

dms: 6.3 PM, 7.3 Months | Component1: 6.3 PM | EAF: 0.3881 | Level: 1

## FEW OF THE COCOMO REPORTS:

dms - Equations Report

Print Export... Headers << Back Next >>

**dms - Equations Report**

SystemStar 3.0 Demo November 7, 2019 15:45:43 Page: 1

Estimate Name:	dms	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

COCOMO Estimating Equations			
Effort	= 2.9400 * EAF * (KSLOC)	1.0201 EAF = 0.3881	= Effort in Person-Months
Schedule	= 3.6700 * (Effort)	0.3020	= Duration in Months
Maintenance Effort	= 2.9400 * EAF * (KSLOC)	1.0201	= Effort (per year) in Person-Months
152 hours per Person-Month			

dms - Effort Report

Print Export... Headers << Back Next >>

**dms - Effort Report**

SystemStar 3.0 Demo November 7, 2019 15:48:22 Page: 1

Estimate Name:	dms	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

Effort per Component (Person-Months)						
Component Name	RQ	PD	DD	CT	IT	Total RQ to IT
Component1	0.4	1.0	1.6	2.1	1.2	6.3

Effort Summary						
Component Totals	0.4	1.0	1.6	2.1	1.2	6.3
Grand Total	0.4	1.0	1.6	2.1	1.2	6.3

dms - Cost & Breakage Report

Print Export... Headers << Back Next >>

**dms - Cost & Breakage Report**

SystemStar 3.0 Demo November 7, 2019 15:50:02 Page: 1

Estimate Name:	dms	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

Increment 1 of 1							
Names of Leaf Components	Developed Size	RQ Cost	PD Cost	DD Cost	CT Cost	IT Cost	Total Cost (K\$)
Component1	5,000	13.9	18.2	28.4	129.1	71.0	260.7
Incr 1 Total	5,000	13.9	18.2	28.4	129.1	71.0	260.7
Grand Total	5,000	13.9	18.2	28.4	129.1	71.0	260.7



dms - Effort & Breakage Report

Print Export... ☒ Headers << Back Next >>

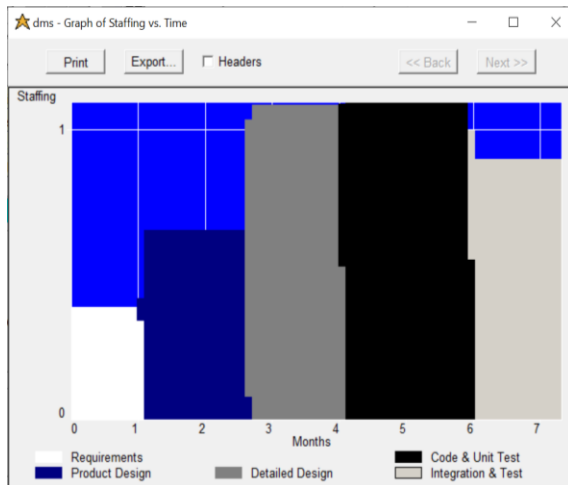
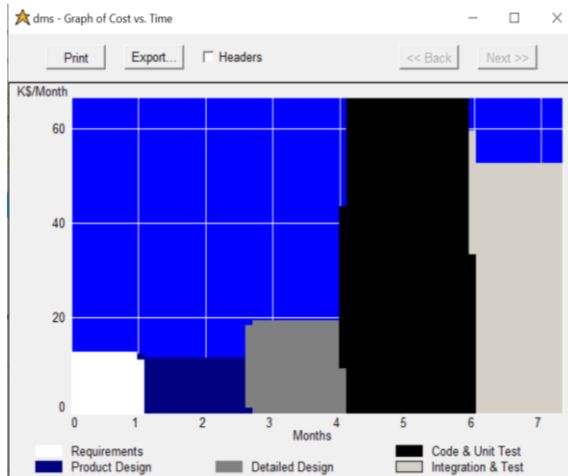
**dms - Effort & Breakage Report**

SystemStar 3.0 Demo November 7, 2019 15:51:24 Page: 1

Estimate Name:	dms	Estimate ID:	
Model Name:	COCOMO® II 2000	Model ID:	2000
Process Model:	COCOMO® II Model	Phases:	Waterfall

Increment 1 of 1

Names of Leaf Components	Developed Size	RQ Effort	PD Effort	DD Effort	CT Effort	IT Effort	Total Effort
Component1	5,000	0.4	1.0	1.6	2.1	1.2	6.3
Incr 1 Total	5,000	0.4	1.0	1.6	2.1	1.2	6.3
Grand Total	5,000	0.4	1.0	1.6	2.1	1.2	6.3



## **REFERENCES:**

<http://www.selectbs.com/process-maturity/what-is-the-capability-maturity-model>  
[https://sunset.usc.edu/research/COCOMOII/expert\\_cocomo/drivers.html](https://sunset.usc.edu/research/COCOMOII/expert_cocomo/drivers.html)  
<https://www.autonationfordarlington.com/>  
<https://tensix.com/2016/02/reducing-project-cost/>  
<https://www.linkedin.com/pulse/20141210012506-13706055-4-ways-to-shorten-a-project-schedule>  
  
<https://www.elinext.com/blog/7-ways-how-reduce-software-development-costs/>  
<https://www.elinext.com/blog/7-ways-how-reduce-software-development-costs/>