Homework #3
CSE 566: Software Project/ Process/ Quality Management
(2015 spring)

Submitted By:

Madhu Meghana Talasila

(1207740881)

Hospital Management System Description

Hospital Management system (client-server architecture model) where users are allowed to access the system 24*7, 365 days using World Wide Web. Features of Hospital Management System

1. Users can use the system at any time
2. Servers process the transactions in a short time.
3. Multiple users can access the system
4. Transaction processing is normal.

Type of transactions that may happen might be booking an appointment, buying medicines, paying hospital bill and other details regarding health checkups. For this kind of project we need some API's which forms the reused and modified part of code such as Email API, Weather forecast API, Map API which accounts to **1,60,000 lines of code**.

New code which we are estimating is for the development of new website, creating various modules like appointment site, medical store, doctor information and diagnosis information. Some of the information like reports, medicine information will be available in the database statically, but data particular to user is generated dynamically on the reports basing on the data and kind of problem the user is facing with. All these kind of development accounts to **2,30,000 lines of code.** This includes code written in the database end to create database and maintain user transactions. [1]

Software Estimation of Drivers for Hospital Management System in general case scenario

Software Scale Drivers:

Precedentedness (High): I rated this feature to be high as this project requires thorough understanding of product requirements, considerable experience in related software and minimal need for innovative data processing architecture.

Development Flexibility (High): I rated this as high because as there is minimal need for software to be in conformance with pre-established requirements and external interface specifications. There are no immediate deadlines and early completion of project is not required.

Architecture/ Risk Resolution (High): This project needs high risk resolution as it generally identifies all critical risk items and created milestones to resolve them and plans the budget and internal milestones. 80 of top software architects are available to the project and some tools support is available for resolving risks. It can take up to 1 critical risk items.

Team cohesion (High): I rated this feature to be high as our team has considerable consistency of stakeholder objectives, cultures and willingness to accommodate other stake holders. Also our team has basic experience in operating as team and to share vision and commitments.

Process Maturity (Extra High): I rated this feature as extra high as our company is CMM level 5 company and it covers all the key process areas like organization process focus, peer reviews, defect prevention etc.

Software Cost Drivers: Product:

Product Software Reliability (High): I rated this as low as the effect of our software failure is relatively high financial loss.

Data base size (Nominal): I rated it as nominal as our project is having (Development time/ People required) is approximately < 100.

Product Complexity (low): I rated this a low, because the weighted average of computational operations and data management operations is low which characterize the product.

Developed for reusability (Nominal): I rated this as nominal because we are planning to reuse the code across the project and we are planning to maintain a considerable design of software, general documentation and proper testing of components so that when we integrate it won't be an issue.

Documentation match to lifecycle needs (Nominal): I rated it as nominal as our project covers the required size of product documentation to life-cycle needs.

Personnel:

Analyst Capability (High): I rated this as high as our analysts stand in almost 75th percentile and they are capable enough to "analyze and design ability, efficiency and thorough ness and has the ability to operate and communicate" [2].

Programmer Capability (High): I rated this as high because our programmers are good team players and they are highly efficient to deal with complex COTS packages.

Application Experience (High): our team has on average software development experience of 3 years so I rated it as high.

Personnel Continuity (High): In our company we have a personnel turnover of 6% per year. So I rated it as high

Platform Experience (High): I rated this as high as our team has a number of personnel whose application level varies some are very highly experienced, others are nominal and low. So I rated this as high.

Language and Toolset Experience (High): I rated this as high as our team members are having varied experience in design, analysis, configuration management, library management etc.

Platform:

Time Constraint (Nominal): I rated this as nominal as our system is expected to use 50% of the available execution time.

Storage Constraint (Nominal): I rated this as nominal as our system is expected to make use of 50% of main available storage

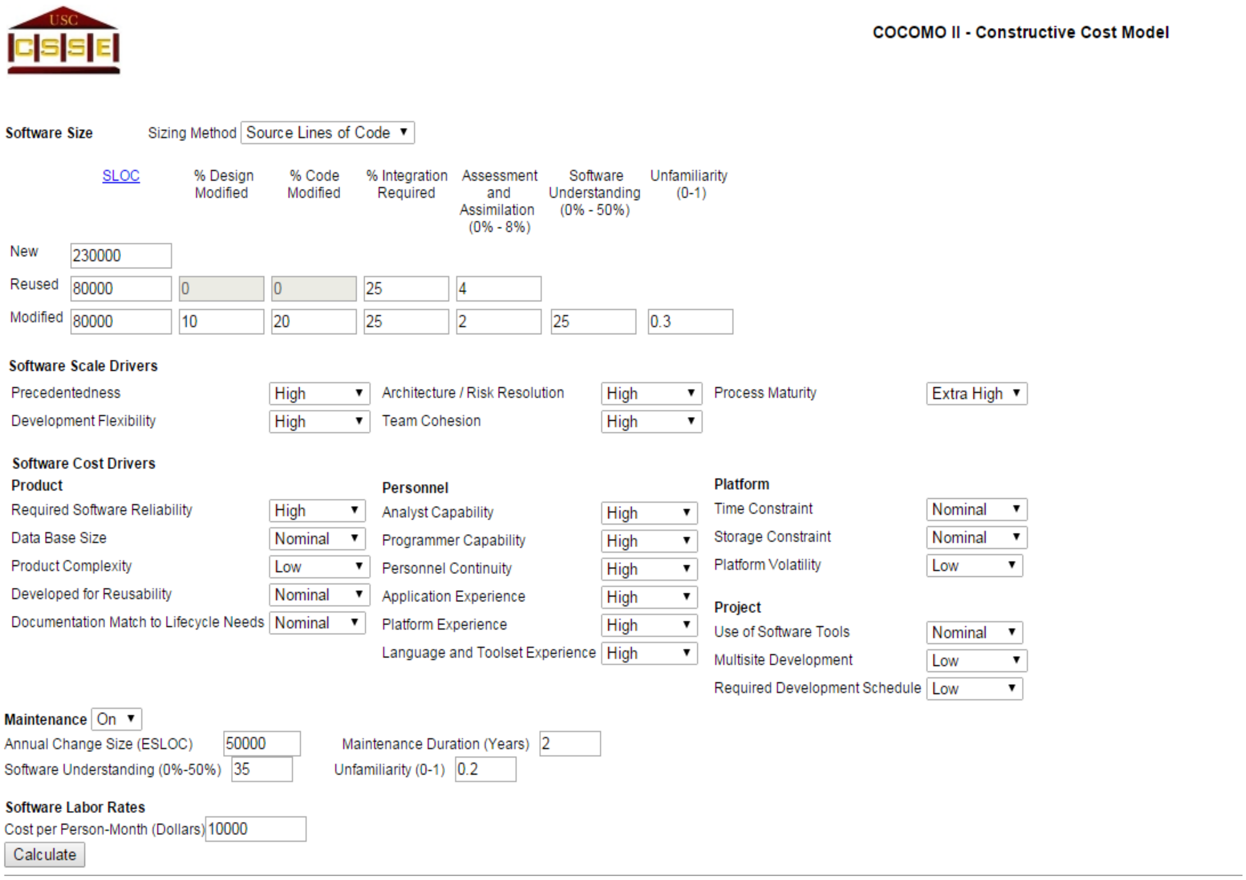
Platform Volatility (low): The platform volatility is low as our system is expected to have a major change once in 12 months and minor change once in one month.

Project:

Use of software tools (Nominal): I rated this as nominal as our team is planned to make use of basic life cycle tools with moderate integration.

Multisite Development (Low): As our team makes use of individual phone or FAX to communicate a max.

Required development Schedule (Low): I rated it as low as our system is developed in such a way that stretch out is 85% of nominal schedule.



COCOMO II - Constructive Cost Model

Software Size Sizing Method: Source Lines of Code ▼

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	230000						
Reused	80000	0	0	25	4		
Modified	80000	10	20	25	2	25	0.3

Software Scale Drivers

Precedentedness	High ▼	Architecture / Risk Resolution	High ▼	Process Maturity	Extra High ▼
Development Flexibility	High ▼	Team Cohesion	High ▼		

Software Cost Drivers

Product		Personnel		Platform	
Required Software Reliability	High ▼	Analyst Capability	High ▼	Time Constraint	Nominal ▼
Data Base Size	Nominal ▼	Programmer Capability	High ▼	Storage Constraint	Nominal ▼
Product Complexity	Low ▼	Personnel Continuity	High ▼	Platform Volatility	Low ▼
Developed for Reusability	Nominal ▼	Application Experience	High ▼	**Project**	
Documentation Match to Lifecycle Needs	Nominal ▼	Platform Experience	High ▼	Use of Software Tools	Nominal ▼
		Language and Toolset Experience	High ▼	Multisite Development	Low ▼
				Required Development Schedule	Low ▼

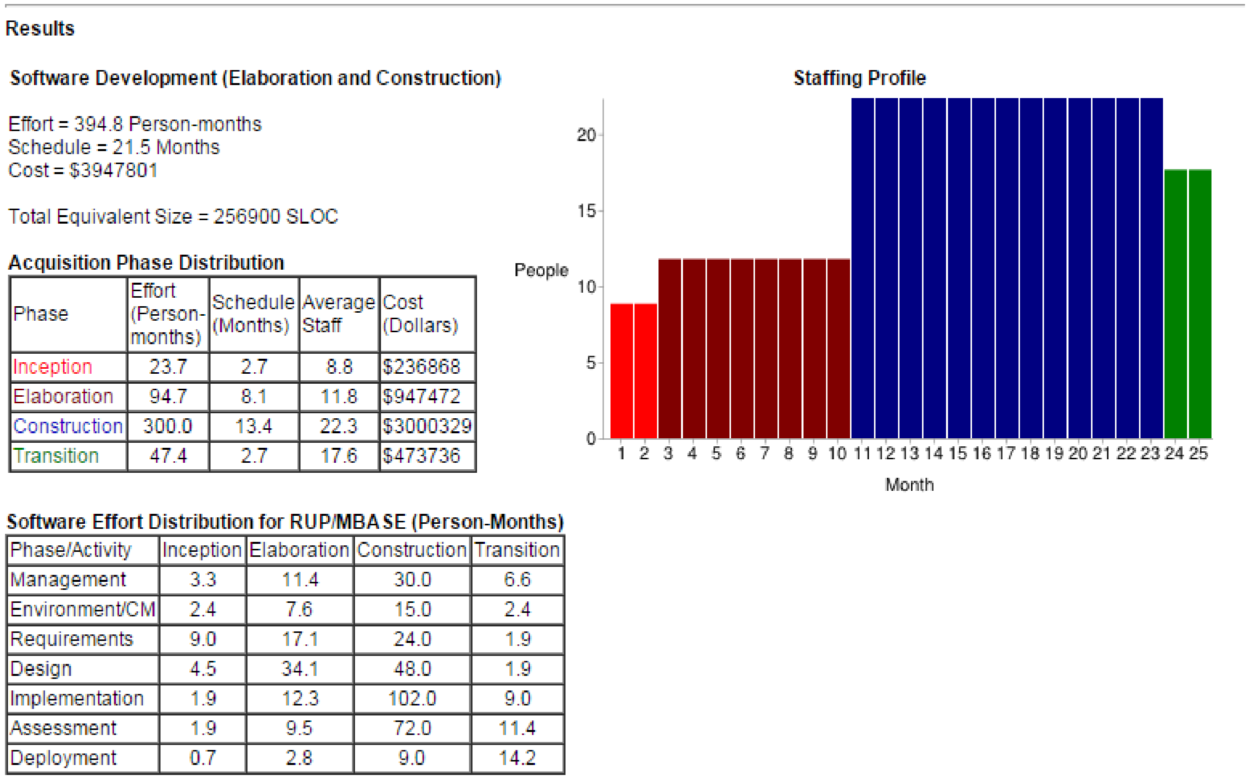
Maintenance On ▼

Annual Change Size (ESLOC)	50000	Maintenance Duration (Years)	2
Software Understanding (0%-50%)	35	Unfamiliarity (0-1)	0.2

Software Labor Rates

Cost per Person-Month (Dollars) 10000

Figure 1 Software estimation in general scenario [3]



Results

Software Development (Elaboration and Construction)

Effort = 394.8 Person-months
Schedule = 21.5 Months
Cost = $3947801

Total Equivalent Size = 256900 SLOC

Acquisition Phase Distribution

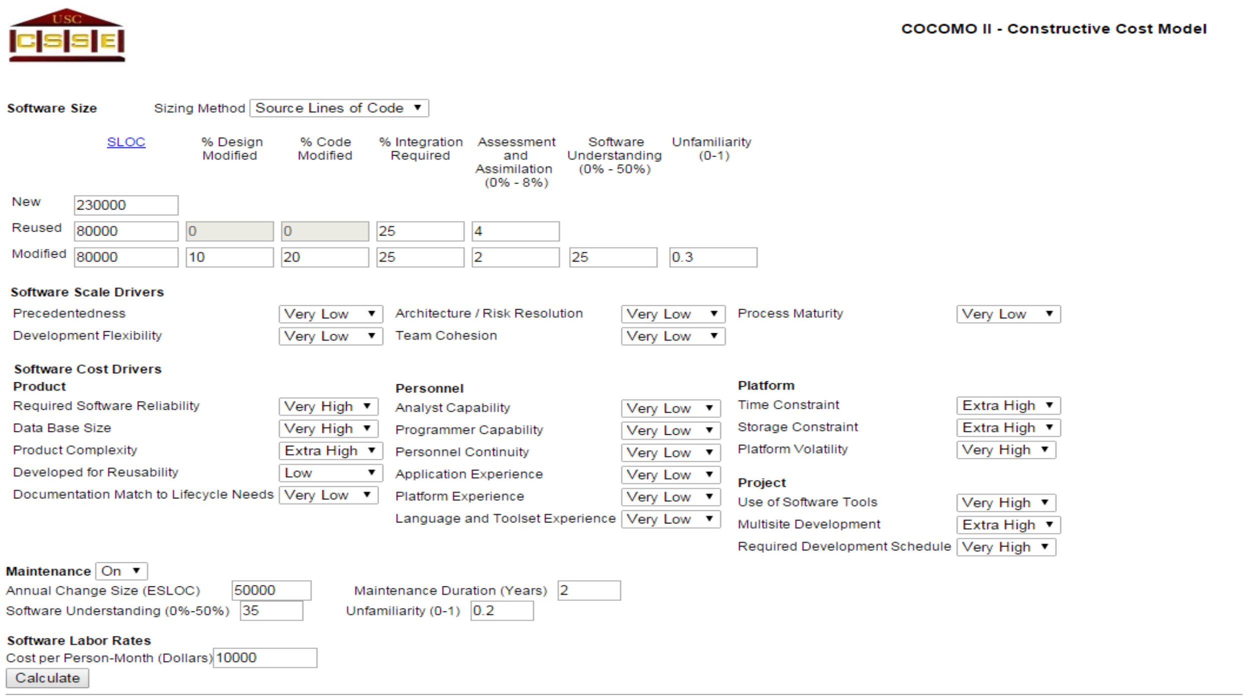
Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	23.7	2.7	8.8	$236868
Elaboration	94.7	8.1	11.8	$947472
Construction	300.0	13.4	22.3	$3000329
Transition	47.4	2.7	17.6	$473736

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	3.3	11.4	30.0	6.6
Environment/CM	2.4	7.6	15.0	2.4
Requirements	9.0	17.1	24.0	1.9
Design	4.5	34.1	48.0	1.9
Implementation	1.9	12.3	102.0	9.0
Assessment	1.9	9.5	72.0	11.4
Deployment	0.7	2.8	9.0	14.2

Figure 2 Result of software estimation in general scenario [3]

Project under worst case scenario



COCOMO II - Constructive Cost Model

Software Size Sizing Method Source Lines of Code

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	230000						
Reused	80000	0	0	25	4		
Modified	80000	10	20	25	2	25	0.3

Software Scale Drivers

Precedentedness	Very Low	Architecture / Risk Resolution	Very Low	Process Maturity	Very Low
Development Flexibility	Very Low	Team Cohesion	Very Low		

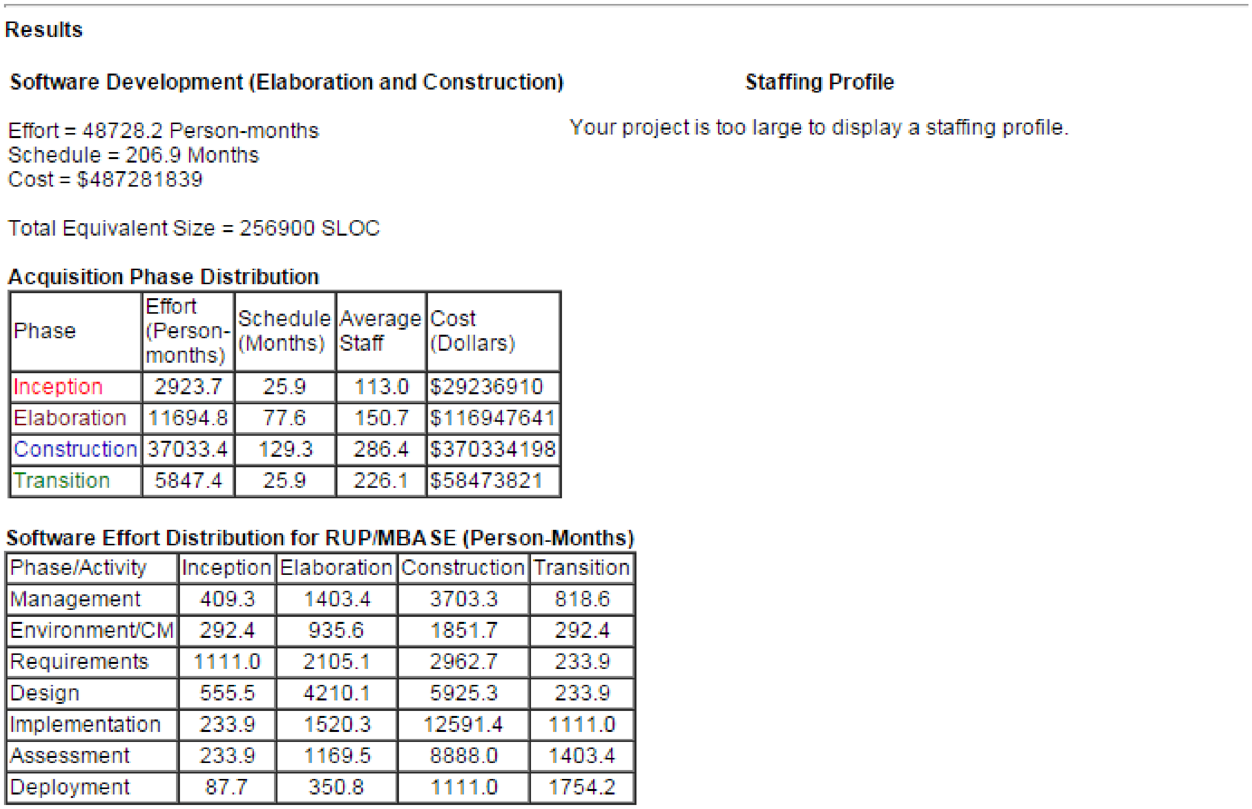
Software Cost Drivers

Product		Personnel		Platform	
Required Software Reliability	Very High	Analyst Capability	Very Low	Time Constraint	Extra High
Data Base Size	Very High	Programmer Capability	Very Low	Storage Constraint	Extra High
Product Complexity	Extra High	Personnel Continuity	Very Low	Platform Volatility	Very High
Developed for Reusability	Low	Application Experience	Very Low	**Project**	
Documentation Match to Lifecycle Needs	Very Low	Platform Experience	Very Low	Use of Software Tools	Very High
		Language and Toolset Experience	Very Low	Multisite Development	Extra High
				Required Development Schedule	Very High

Maintenance On
Annual Change Size (ESLOC) 50000 Maintenance Duration (Years) 2
Software Understanding (0%-50%) 35 Unfamiliarity (0-1) 0.2

Software Labor Rates
Cost per Person-Month (Dollars) 10000

Figure 3 Software estimation in worst case scenario [3]



Results

Software Development (Elaboration and Construction)

Effort = 48728.2 Person-months
Schedule = 206.9 Months
Cost = $487281839

Total Equivalent Size = 256900 SLOC

Staffing Profile

Your project is too large to display a staffing profile.

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	2923.7	25.9	113.0	$29236910
Elaboration	11694.8	77.6	150.7	$116947641
Construction	37033.4	129.3	286.4	$370334198
Transition	5847.4	25.9	226.1	$58473821

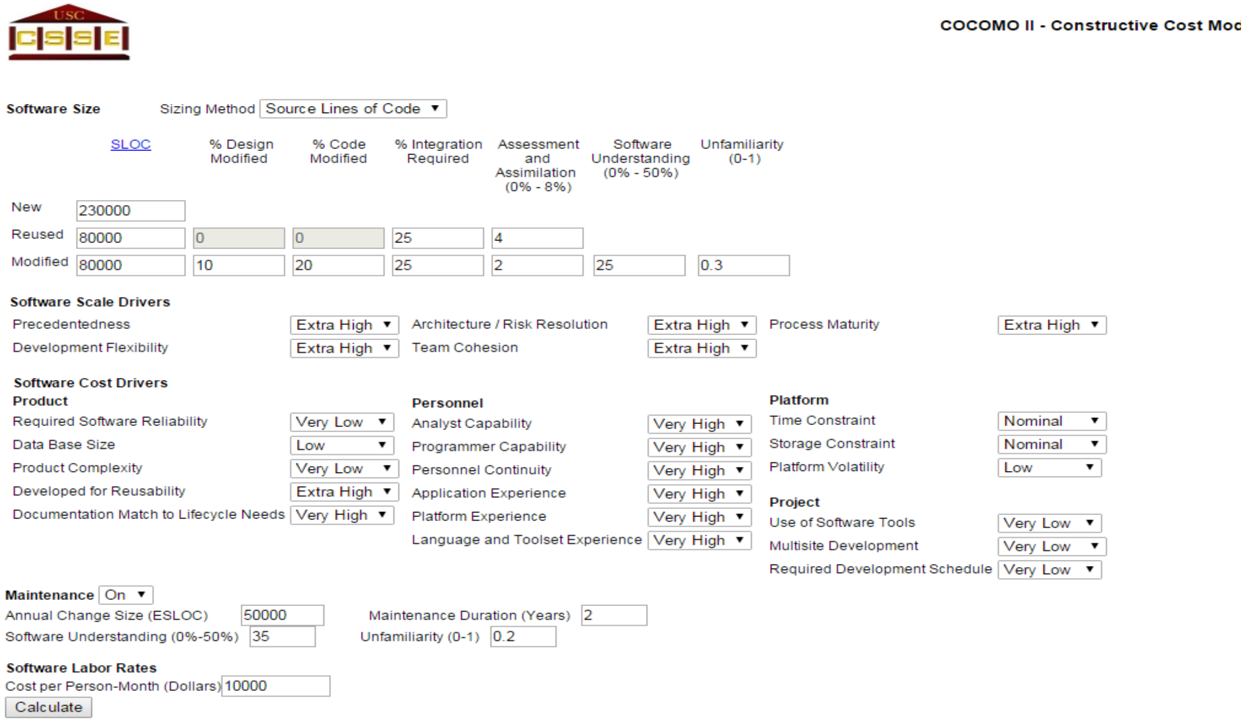
Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	409.3	1403.4	3703.3	818.6
Environment/CM	292.4	935.6	1851.7	292.4
Requirements	1111.0	2105.1	2962.7	233.9
Design	555.5	4210.1	5925.3	233.9
Implementation	233.9	1520.3	12591.4	1111.0
Assessment	233.9	1169.5	8888.0	1403.4
Deployment	87.7	350.8	1111.0	1754.2

Figure 4 Result of software estimation in worst case scenario [3]

In the worst case scenario, Effort, cost and schedule increased drastically and project became too large to display a staffing profile and staffing profile is not displayed. Also Effort, Schedule, cost increased almost 100 times to general case scenario.

Project under Ideal Conditions



COCOMO II - Constructive Cost Moc

Software Size Sizing Method Source Lines of Code ▾

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	230000						
Reused	80000	0	0	25	4		
Modified	80000	10	20	25	2	25	0.3

Software Scale Drivers

Precedentedness	Extra High ▾	Architecture / Risk Resolution	Extra High ▾	Process Maturity	Extra High ▾
Development Flexibility	Extra High ▾	Team Cohesion	Extra High ▾		

Software Cost Drivers

Product		**Personnel**		**Platform**	
Required Software Reliability	Very Low ▾	Analyst Capability	Very High ▾	Time Constraint	Nominal ▾
Data Base Size	Low ▾	Programmer Capability	Very High ▾	Storage Constraint	Nominal ▾
Product Complexity	Very Low ▾	Personnel Continuity	Very High ▾	Platform Volatility	Low ▾
Developed for Reusability	Extra High ▾	Application Experience	Very High ▾	**Project**	
Documentation Match to Lifecycle Needs	Very High ▾	Platform Experience	Very High ▾	Use of Software Tools	Very Low ▾
		Language and Toolset Experience	Very High ▾	Multisite Development	Very Low ▾
				Required Development Schedule	Very Low ▾

Maintenance On ▾

Annual Change Size (ESLOC) 50000 Maintenance Duration (Years) 2

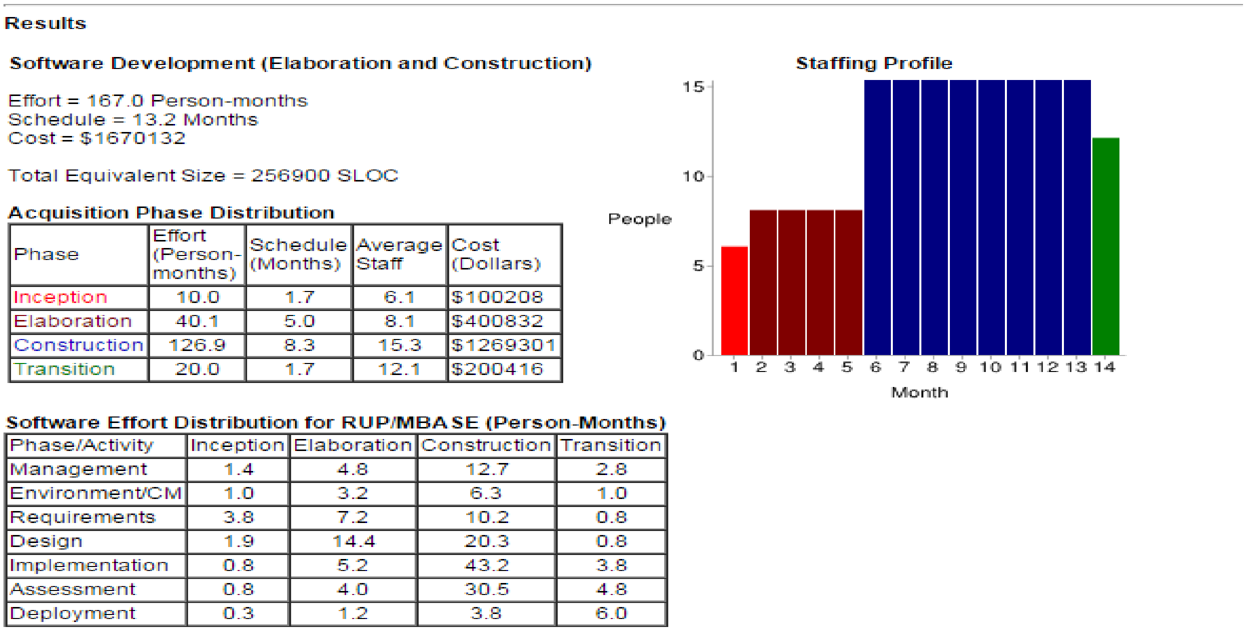
Software Understanding (0%-50%) 35 Unfamiliarity (0-1) 0.2

Software Labor Rates

Cost per Person-Month (Dollars) 10000

Calculate

Figure 5 Software estimation in ideal case scenario [3]



Results

Software Development (Elaboration and Construction)

Effort = 167.0 Person-months
Schedule = 13.2 Months
Cost = $1670132

Total Equivalent Size = 256900 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	10.0	1.7	6.1	$100208
Elaboration	40.1	5.0	8.1	$400832
Construction	126.9	8.3	15.3	$1269301
Transition	20.0	1.7	12.1	$200416

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	1.4	4.8	12.7	2.8
Environment/CM	1.0	3.2	6.3	1.0
Requirements	3.8	7.2	10.2	0.8
Design	1.9	14.4	20.3	0.8
Implementation	0.8	5.2	43.2	3.8
Assessment	0.8	4.0	30.5	4.8
Deployment	0.3	1.2	3.8	6.0

Figure 6 Result of software estimation in ideal case scenario [3]

In the ideal case scenario, Effort, cost and schedule decreased drastically and project became too small and can be completed in almost ¾ th the time span of general scenario. Also cost required for project completion also decreased.

Ideal case < General Case < Worst case

This holds for all the three factors (Effort, Schedule and cost) of software development. Which can be visualized in staffing profile.

References:

1. COCOMO II/Chapter 3/Boehm *et al.*
2. http://sunset.usc.edu/research/COCOMOII/expert_cocomo/drivers.html
3. http://csse.usc.edu/tools/COCOMOII.php