



CTIS411 SENIOR PROJECT-I
SOFTWARE PROJECT MANAGEMENT
PLAN
(SPMP)

INeed

Advisor: Neşe Özçelik

Team#7

Enes Kuluk
Oğulcan Delice
Mert Şağban
Mert Yılmaz

Fall 2019 - 2020

Table of Contents

Table of Contents	2
1 Project Effort & Cost Estimation	3
1.1 Process & Product Decomposition Based Estimation	3
1.2 Function Point Analysis (FPA) / Constructive Cost Model (COCOMO) Based Estimation	7
1.3 Use Case Based Estimation	8
1.3.1. Unadjusted Use Case Weight (UUCW)	8
1.3.2. Unadjusted Actor Weight (UAW)	9
1.3.4. Environmental Complexity Factor	10
1.3.5. Use Case Point	11
2 Project Scheduling	12
3 Project Measuring	13
4 Project Controlling	14
4.1 Configuration Management	14
4.2 Verification and Validation	15
4.3 Quality Management	16
5 Software Development Environment	17

1. PROJECT EFFORT & COST ESTIMATION

1.1 Process & Product Decomposition Based Estimation

To view Work Breakdown Structure, please check out “WBS.png”.

1.2 Function Point Analysis (FPA) / Constructive Cost Model (COCOMO) Based Estimation

Exponent Factor Estimation

Attribute	Scale Factor	Value
Precedentedness	Nominal	3.72
Development Flexibility	High	2.03
Architecture / Risk Resolution	Nominal	4.24

Team Cohesion	High	1.10
Process Maturity	Nominal	4.68
	Exponent Factor	0.2487

Purpose: Calculate Exponent Factor

Exponent Factor = $0.01 * (\text{Values}) + 0.091$

Exponent Factor = 0.2487

Category	Factor	Assessment	Parameter
Product Attributes	Required Software Reliability	High	1.10
	Database Size	High	1.14
	Product Complexity	Nominal	1.00
Platform Attributes	Execution Time Constraints	Nominal	1.00
	Main Storage Constraints	High	1.05
	Platform Volatility	Nominal	1.00
Personal Attributes	Analyst Capability	High	0.85
	Applications Experience	Nominal	1.00
	Programmer Capability	High	0.88
	Programming Language Experience	Nominal	1.00
Project Attributes	Use of Modern Programming Practices	Nominal	1.00
	Use of Software Tools	High	0.90
	Required Development Schedule	Nominal	1.0
	Effort Multiplier		0.88640244

Purpose: Calculate Effort Multiplier (EP)

EP = 0,88640244

Calculation of Unadjusted Function Points(UTF)

		Weighting Factor		
		Simple	Average	Complex
Inputs	Click Search Need	3		
	Specify criteria on search	3		
Outputs	Database brings out available needs according GPS information	4		
	Database offers options	4		
	Database returns desired content	4		
Inquiries	Click Search		4	
	Unadjusted Function Points:	22		

$SLOC = UFP * JavaMultiplier$

$SLOC = 22 * 53$

$SLOC = 1166$

$KLOC = SLOC / 1000$

$KLOC = 1,166$

$Effort = 2,94 * KLOC^{ExponentFactor} * EffortMultiplier$

$Effort = 2,94 * 1.03893392903 * 0,88640244$

$Effort = 2.70748589489$

1.3 Use Case Based Estimation

Factor	Use Case Name	Transaction Number	Complexity
F1	Request Help	2	Simple 5
F2	Accept Need	2	Simple 5
F3	Approve Need	2	Simple 5
F4	Report User	4	Simple 5
F5	View Activities	6	Simple 5
F6	Select Need	2	Simple 5

F7	Share Location	4	Simple 5
F8	Enable Location	6	Simple 5
F9	Ban User	2	Simple 5
F10	Search Need	4	Average 10
F11	Offer Help	2	Average 10

Table 1.1

Purpose: Calculate Unadjusted Use Case Weight (UUCW)

UUCW = 65

Actor Type	Example	Weight (W)	Impact (W * U)
Simple	Person-DB Transaction	1	1
Average	Person text-based GUI	2	2
Complex	Another system via API	3	0

Table 1.2

Purpose: Calculate Unadjusted Actor Weight (UAW)

UAW = 3

Unadjusted Use Case Points (UUCP) = UUCW + UAW

UUCP = 65 + 3

UUCP = 68

Factor	Description	Weight (W)	Given Value (GV) (Between 0-5)	Impact (W * GV)
F1	Distributed System	2.0	0	0
F2	Response Time or Throughput Performance Objectives	1.0	5	5
F3	End User Efficiency	1.0	4	4
F4	Complex Internal Processing	1.0	0	0
F5	Code Must Be Reusable	1.0	4	4
F6	Easy to Install	.5	5	2.5
F7	Easy to Use	.5	5	2.5
F8	Portable	2.0	5	10
F9	Easy to Change	1.0	5	5

F10	Concurrent	1.0	5	5
F11	Includes Special Security Objectives	1.0	2	2
F12	Provides Direct Access for Third Parties	1.0	4	4
F13	Special User Training Facilities are Required	1.0	0	0
Total Factor: (TF)				44

Table 1.3

Purpose: Calculate Technical Complexity Factor (TCF)

$$TCF = 0.6 + (TF/100)$$

$$TCF = 0.6 + (44/100)$$

$$TCF = 1.04$$

Factor	Description	Weight (W)	Given Value (GV) (Between 0-5)	Impact (W * GV)
F1	Familiar with the Project Model that is Used	1.5	3	4.5
F2	Application Experience	.5	3	1.5
F3	Object-Oriented Experience	1.0	3	3
F4	Lead Analyst Capability	.5	3	1.5
F5	Motivation	1.0	5	5
F6	Stable Requirements	2.0	3	6
F7	Part-Time Staff	-1.0	4	-4.0
F8	Difficult Programming Language	-1.0	3	-3.0
Total Environment Factor: (TEF)				14.5

Table 1.4

Purpose: Calculate Environmental Complexity Factor (ECF)

$$ECF = 1.4 + (-0.03 * EF)$$

$$ECF = 1.4 + (-0.03 * 13.5)$$

ECF = 0.965

2. PROJECT SCHEDULE

To view project schedule, please check out the "INeedGanttChart.mpp" file.

3. PROJECT MEASURING

		WHAT	WHEN	HOW	WHO
PROJECT METRICS		Meetings with the advisor (customer)	Weekly	Meeting	All of Team Members
		Completion rate	Weekly	MS Project	All of Team Members
		# of Requirement Changes	After meetings	Meeting	All of Team Members
		Efficiency	After release	During testing	Mert Yılmaz
PROCESS METRICS		Errors Handled	Before each iteration	During integration testing	All of Team Members
		# of Bugs	Weekly	During coding and testing	Oğulcan Delice
PRODUCT METRICS	Dynamic metrics	Response Time	After each iteration	During testing	Enes Malik Kuluk
		Scalability	Weekly	After requirements finished	All of Team Members
	Dynamic metrics	Desing Integration	After first iteration and weekly update	Front-end tools	Enes Malik Kuluk, Mert Yılmaz
		LOSC (line of source codes)	Weekly	Git	Ö. Mert Sağban, Oğulcan Delice
		# of Classes	After each iteration	Git	Ö. Mert Sağban
I NEED					

4. PROJECT CONTROLLING

4.1 VERSION MANAGEMENT

We are going to use Git as a version management tool. With this tool, team can effectively control changes that may occur in requirements, team and project schedule. After each incrementation it will be implemented on a different branch. It provides managing the code by monitoring to find bugs, task assignments among the team, controlling the process. Each team member are going to be working on their own. And our advisor Neşe will have access to Git to control commits.

4.2 VERIFICATION & VALIDATION

During this phase, project INeed will be analyzed and validated .This phase will be implemented after initial implementation. All team members are involved in this process.

Verification will be done by testing. Automated test methods, code reviewing and integration test will be used. Interface testing will be done manually by team members. For android part, unit testing will be done with Espresso framework. For web part, both front end and back end part will be tested by MochaJS since it supports Node.Js debugger. For compatibility testing, many browsers and android devices will be used in order to test compatibility. Performance metrics will be adjusted according to stress and load testing. Web server and database will be stressed in order to find any bottlenecks or performance issues regarding optimization.

To check project's sufficiency, validation test will be performed. Acceptance test will be made with advisor to meet satisfaction of project owner. Also blackbox and whitebox testing methods will be used for internal and external validation.

4.3 QUALITY MANAGEMENT

To provide a high quality product, during the development of the project, some standards will be used. Google's web design and javascript standards will be followed during development phase to ensure a quality product. For android development, Google Java code style and AOSP Java Code style which is related with Google java code style will be used. Over these standards, we plan to make internal and external meetings in order to have high quality product. Internal meetings include only developers while external meetings include both developers and advisor(project owner). Internal meetings are about planning of development phase. On the other hand, external meetings are about current status of the project, evaluation of project to meet expectations and to have a high quality product.

5. Software Development Environment

Tool	Version
Windows	10
Ubuntu	19.04
WebStorm	2019.3
Visual Studio Code	1.40
Android Studio	3.5.2
FireBase Database	
Google Maps API	3.39
React.js	16.12.0
Git	2.24.0
Node.js	13.1.0

We will create both android application and web page. For Android, Android Studio will be used for android development. Web part will be developed on Webstorm and Visual Studio Code. React.js will be used for front end of web part. For server-side development, Node.js will be used. For database, FireBase Realtime database will be used. It is a NoSql database which is suitable for our usage. For geolocation tracking and marking, Google Maps API will be used. Version control will be handled with Git. All of these operations will be done on both on Windows and ubuntu operating systems.