University of Prince Mugrin

College of Computer and Cyber Sciences

Department of Software Engineering

SE464- Software Project Management

Project – Semester I (Fall 2022)

Smart Fire Alarm System

Team Members:

Jana Aldubai 4010372

Salwa Shama 4010405

Samah Shama 4010403

Sana Shama 4010404

Instructor:

Dr. Osama Qaed

20 December 2022

Chapter 1: Introduction

1.1 System Overall

Our project aims to build a smart fire alarm system that can detect fires in its early stages, even if it is light. It is designed to help people prepare and take the necessary precautions in case of any fire in the building. This idea will greatly help save many lives and reduce material losses significantly.

In case of a fire in the building, the device will sound an alarm informing the residents of the building of the need to evacuate the building and move to a safe place and direct them to the nearest emergency exit. At the same time, the device deals with fire, as it can distinguish between types of fires and spray the chemical depending on the type of fire. The device is also connected to the air conditioning of the building, where it tries to purify the air from smoke until the civil defense arrives. The device is also connected to the civil defense department, where in the event of a fire, the device sends a signal to the civil defense department informing them about the location and details of the fire directly without human intervention.

Chapter 2: Scope Management Plan

As a starting point for our team, we decided to carry out the project scope document, which we considered the baseline for our project and because it would be utilized as part of our contract with the customer. We also wanted to make sure that we and they were on the same page. (We assumed that the feasibility study was conducted from other parity, as in usual cases, and that the community recognized the project's value to us.)

The project scope document, as it is known in the project management field, contains the main characteristics of the project, which are represented in four major components: project goals and project requirements; assumptions and constraints; project outputs and control points; and project rosters.

2.1 Project Goals amd Project Requirements

Our team met to do the first assignment task, which was starting with the first section of the scope document, which is project goals and project requirements. In this part, we had to define the project's boundaries—those that are under our responsibility and those that fall within the control of third parties; in other words, we had to add scope to our project. Therefore, our team decided to use SMART techniques, which stands for simple, measurable, agreed upon, realistic, and time-bound, to list the goals. At the end, we came up with the following:

No.	Project Goals
1	Developing smart fire alarm system to be embedded in public building within a year.
2	Developing smart fire alarm system that meet the requirements of the National Fire Protection Association (NFPA).

Table 2.1.1 Project Goals

Regarding the requirements side, we have many requirement techniques that assist us in eliciting the requirements. As this project contains hardware, it was preferable to do prototyping to overcome the uncertainty that is inherent in these types of projects. In addition, we thought the observation would give us a big picture of how some sectors handle the management of people in a fire crisis. Here is the first set of requirements that we figured out:

No.	Project Requirements
1	The system shall be able to create risk map that directs victims to exit points.
2	The system shall be to calculate the risk level.
3	The system shall be able to send notifications to fire stations.
4	The system shall be to send data to other related systems, which are HVAC and sprinkler systems.
5	The system shall be able to generate alerts.

Table 2.1.2 Project Requirements

For the second requirement, we have the requirement traceability matrix (RTM) listed here.

Requirement No.	Name	Category	Source	Status
R2	Risk calculation	Software	Project charter and NFPA specification	Not Yet Started (NYST)

Table 2.1.3 Requirement Traceability Matrix

At the end of the first section, we were able to reduce the project's uncertainties and incorrect expectations. We also used a variety of techniques to write the first component, including SMART method, prototyping and observation to elicit the requirement, as well as a requirements traceability matrix to make sure that every requirement was factored into the equation.

2.2 Assumptions and Constraints Constraints

In the second section of the scope document, which is assumptions and constraints, we have to investigate what our constraints and limitations are, in addition to making some assumptions that can be true or false.

Starting with the assumption that we can do roughly anticipatory analysis in terms of three factors, namely manpower, cost, and time, we can look at similar systems and come up with the following:

No.	Project Assumptions
1	Budgeted $100,000 for hardware and software, as well as $500,000 for labor.
2	Estimate hours required to complete the project 1800.
3	All stakeholder will be available during alpha testing.

Table 2.2.1 Project Assumptions

There were three types of limitations that faced any projects: external, organizational, and internal (from the product itself). Here are some examples:

No.	Project Constrains
1	The system should be deployed by the end of 2024.
2	The system should be developed by following NFPA standers.
3	The process of training and supporting for the use of the system will last for one month.

Table 2.2.2 Project Constrains

At the end of the second section, we have a reasonable explanation why we established certain objectives over others and why we chose one structure over another for the task [2].

2.3 Project Outputs and Control Points

In the third section, it provides an overview of how the schedule will be broken down into milestones and deliverables, which is project outputs and control points. We present to you the main outputs from our respective; in terms of deliverables, we have SRS and a user manual, as shown in the deliverable table. Also, SRS and kick-off meetings are considered our milestones in developing smart fire alarm system.

Del. ID	Deliverable Name	Nature	Dissemination level	Delivery Date (Project Month)
1	SRS	R	CO	0
2	User manual	R	PU	12

* Dissemination Level: PU: Public; CO: Confidential

Table 2.3.1 Deliverable Table

Milestone Number	Milestone Name	Date (Month from start)	Means of Verification
MO.1	Kick-off	1	Kick-off meeting minutes available, collaboration tools available.
MO.2	SRS	0	SRS document available.

Table 2.3.2 Milestone Table

2.4 Project Roster

The last section we have in our scope document is project roster, as we know there are multi-way to present and analysis stakeholders such as RAIC chart, power interest grid and so on. In our report we will use on power interest grid technique.

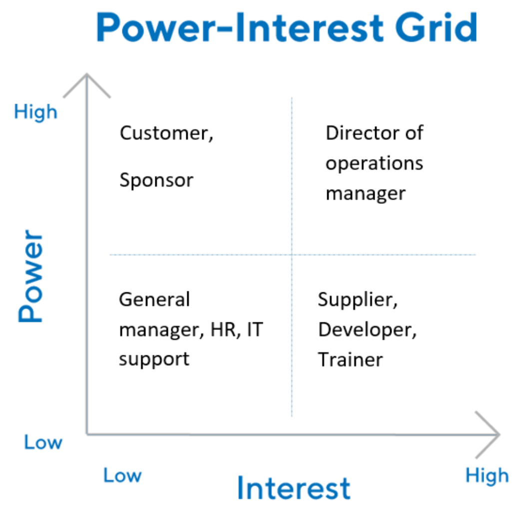

Figure 2.4.1 Power-Interest Grid

2.5 Work Breakdown Structure (WBS)

The study has been closed for the project scope document. However, there is still one powerful technique that is established at this stage and that will be used later in the schedule plan, which is work break structure (WBS). So, our WBS style is process-oriented, and it has two levels. The first level describes the phases of SDLC, and then we go in depth to the test phase, which includes unit, system, and integration tests. as illustrated in the following diagram.

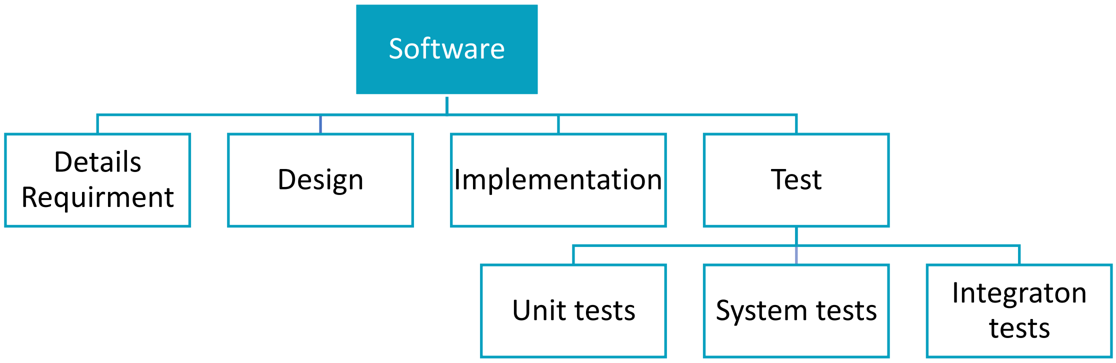

Figure 2.4.2 Work Breakdown Structure

That is all for the scope plane. The next section of the report highlighted the schedule plan.

Chapter 3: Scheduale Management Plan

3.1 Function Point(FP) and COCOMO

We tried to find historical data on the internet to estimate the time but we didn't find, so we started putting the initial plan by using different estimation techniques which are Function Points FP and COCOMO.

First, to calculate the FP we identified the five measurement parameters for our project which are: user inputs, user outputs, user inquiries, internal logical files, and external interfaces.

- User inputs:

1- The user can cancel the fake alarm.

- User outputs:

1- The percentage of smoke from each detector.
2- The risk levels.
3- The specific place of fire if exist.
4- The number of people existing in the building.
5- An alert when a fire exists.
6- A map leading to safe exits.

- User inquiries:

1- The cancelation from the user will turn off the alarm when a fake fire exists.

- Internal logical files:

1- A file containing the smoke percentage.
2- A file containing the map to safe exits.

- External interfaces:

1- Fire station.
2- HAVAC system.
3- Sprinkler system.

After specifying measurement parameters, we calculated the non-functional characteristics of our system by answering the 14 questions.

1. Does the system require reliable backup and recovery?
2. Are data communications required?
3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?
6. Does the system require online data entry?
7. Does the online data entry require the input transaction to be built over multiple screens or operations?
8. Are the master files updated online?
9. Are the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code to be designed reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

1: 1, **2:** 5, **3:** 2, **4:** 5, **5:** 5, **6:** 0, **7:** 0, **8:** 1, **9:** 3, **10:** 3, **11:** 5, **12:** 2, **13:** 2, **14:** 0

Now we calculated FP using the below table taking into consideration average values:

Measurement Parameter	Low	Average	High
1. Number of external inputs (EI)	7	10	15
2. Number of external outputs (EQ)	5	7	10
3. Number of external inquiries (EQ)	3	4	6
4. Number of internal file (ILF)	4	5	7
5. Number of external interfaces (EIF)	3	4	6

Table 3.1.1 Measurement Parameter

FP = UFP*VAF

UFP=(1*10)+(6*7)+(1*4)+(2*5)+(3*4)=78

VAF= (65+1+5+2+5+5+0+0+1+3+3+5+2+2+0)/100= 0.99

FP=(78)(0.99)= 77.22 ≈78

Then we converted the function point into lines of code by considering the use of the C programming language [3].

LOC= (26880*78)/128 = 16380

KSLOC=16380/1000=16.38

Then we calculated the effort, duration, and number of team members needed to build our project using the COCMO technique. We considered our project to be an embedded project as it has complex hardware interfaces and various technical and development risks.

	APM	BPM	ATDEV	BTDEV
Organic	2.4	1.05	2.50	0.38
Semidetached	3	1.12	2.50	0.35
Embedded	3.6	1.20	2.50	0.32

Table 3.1.2 COCOMO Base Model

PM= $(APM)(KSLOC)^{BPM}$ $=(3.60)(16.38)^{1.20}$ =103.15≈ 104 person per month

TDEV=$(ATDEV)(PM)^{BTDEV}$ $=(2.50)(104)^{0.32}$ =11.05≈12 month

TEAM=PM/TDEV=104/12 =8.66≈9 person

3.2 Task Dependency

Then we started identifying the dependencies among the activities in our WBS. The first activity has no dependency on other activities as it is the first activity. The second activity can’t start until the first activity starts (start-to-start). The third activity can’t start until the second activity starts (start-to-start). The fourth activity can’t start until the third activity starts (start-to-start), and also can’t finish until the third activity finishes (finish-to-finish). The fifth activity can’t start until the fourth activity starts (start-to-start), and also can’t finish until the fourth activity finishes (finish-to-finish). The sixth activity can’t start until the fifth activity finishes (finish-to-start).

Then we calculated the start and end of each activity taking into consideration the dependencies among the activities.

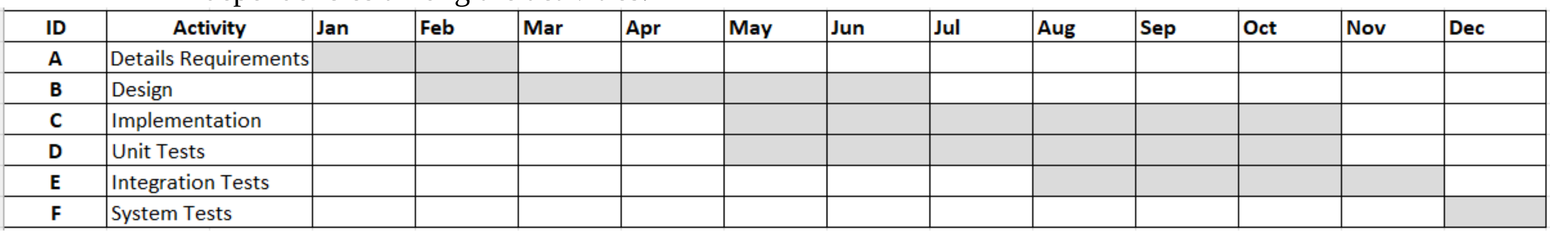

Figure 3.1.3 Task Dependency

3.3 Critical Path

Then we identified the critical path which is A→B→C→D→E→F

Activity	Predecessor	Duration
A	-	2
B	A(SS1)	5
C	B(SS3)	6
D	C(SS),C(FF)	6
E	D(SS3),D(FF)	4
F	E(FS)	1

Table 3.3.1 Task Dependency

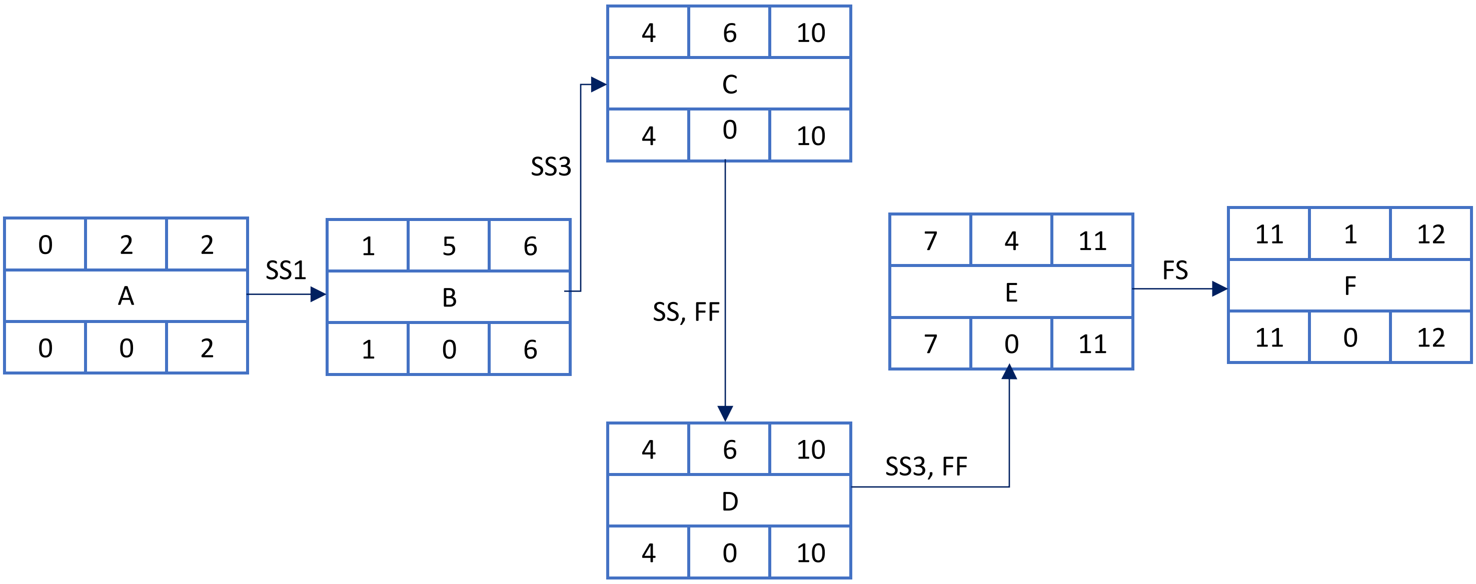

Figure 3.3.2 Critical Path

3.4 Allocate Resources

Then we started allocating the needed resources for our project. One of the important resources is manpower. From the COCOMO we calculated the manpower for the whole project which is 9 persons with effort 104 per month, but here we calculated the manpower for each activity.

A: (2/12)(104) = 17.33 ≈ 18 person per month

B: (5/12)(104) = 43.33 ≈ 44 person per month

C: (6/12)(104) = 52 person per month

D: (6/12)(104) = 52 person per month

E: (4/12)(104) = 34.66 ≈ 35 person per month

F: (1/12)(104) = 8.66 ≈ 9 person per month

The second important resource in our project is the equipment. We need smoke detectors and human sensors to count the number of heads in the building. The number of smoke detectors and head counters depends on the area of the building itself and the area the detectors and counters can read.

3.5 Gantt Chart

Then we draw a simple Gantt chart for the start and end of each activity of our project.

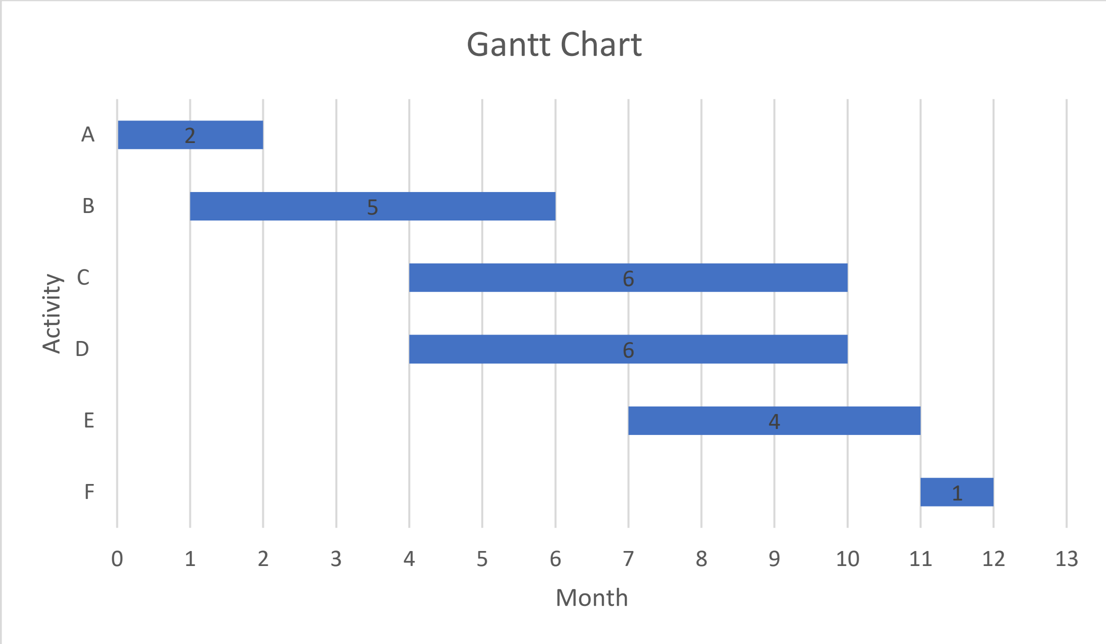

Figure 3.4.1 Gantt Chart

Chapter 4: Cost Management Plan

Project budget: “is the estimated monetary resources needed to achieve a project's goals and objectives”.[1] In the early stage of planning, we reviewed some of our similar projects and some of the historical data to review how that project’s budget was handled, to find out what went well, and learn from any previous mistakes. In addition to that we set down with people who are experts in the manufacture of fire equipment, and we discuss with them the expected cost or the budget the project will require, and we conclude with this:

4.1 Direct and Indirect Cost of the Project

A. Direct Cost

We need to hire some electrical engineer to work on a project, either exclusively or for an assigned number of hours, their salary is a direct cost. Also, we need to have an employee who puts together the component of fire detector parts the employee’s work is considered direct labour. To create the fire detector, we must consider purchase of materials and equipment that the employees need it. For example, the employee needs plastic, small electronic alarm horn and a printed circuit board with a set of electronic components which is considered a direct material.

Also, our company develops a customized software for this type of fire detector, so we need specific assets, such as purchased frameworks or development applications, all of those will consider it as a direct cost.

B. Indirect Cost:

We want also to rent and utility payments to keep your business going. In addition, we must buy some computers, business insurance, and marketing campaigns, these costs are not directly related to producing a specific product or performing a service, so they are indirect costs.

4.2 Cost Element Structure (CES)

The experts advise us to use a cost element structure (CES), that is, a hierarchical structure that defines precisely what are the cost items to consider in each project.[2]

The hierarchical structure below, it is cost element structure (CES) for our project

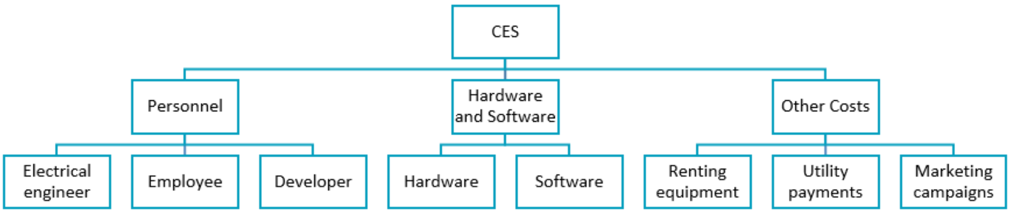

Figure 4.2.1 Cost Element Structure

Electrical engineer: (70+50) x10=1200

Employee: (50+30) x 50=8000

Developer: (50+30) x 50=8000

Total cost for a personal package:17200+ 10 for risk = 17210

Hardware: 500 x 10=5000

Software: 100 x 3=300

Total cost for a Hardware and Software package: 5300 + 10 for risk = 5310

Renting equipment: 500 x 10=5000

Utility payments: 400 x 3=1200

Marketing campaigns: 100 x 10=1000

Total cost for a Other Costs package: 7200 + 10 for risk = 7210

Total Cost = (17210+ 5310+ 7210) *1.1=32,703

What is a grateful advice! it helps us to present the budget in a standardized way and allows us to aggregate and present financial data at different levels of details.

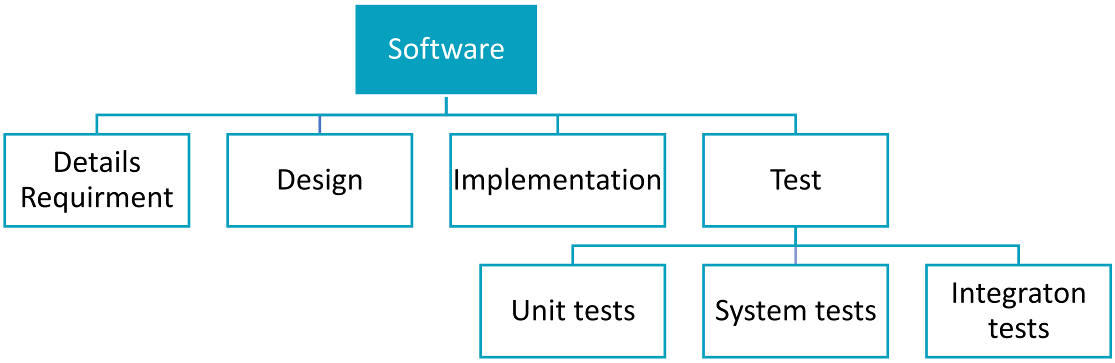

Figure 4.2.2 Work Breakdown Structure

1. Details Requirement phase will cost:800,000
2. Design phase will cost: 700,000
3. Implementation phase will cost: 650,000
4. Test phase will cost:
    4.1 Unit test: 970,000
    4.2 System test: 970,000
    4.3 Integration test: 970,000

4.3 Determining the Project Costs

Moreover, one of the greatest things we come out from meeting is: we could determine or estimate the cost of our project by using a special formula: by estimating effort and multiplying by the loaded cost per unit of effort and by adding other costs, such as hardware, business insurance, and utility payments.

$$\text{Project budget} = \sum_{j=1}^{m} \text{Hours}_j * (\text{Cost}_j + \text{Overhead}_j) + \sum_{i=1}^{n} C_i$$

	Unit Cost	Overhead	Effort/Unit	Total
Personnel				
Electrical Engineer	70$	50$	10	1200$
Employee	50$	30$	50	8000$
Developer	50$	30$	50	8000$
Total Personal Cost			**17200$**	
Total Personal Cost with risk			**17210$**	
Hardware and Software				
Hardware	500$		10	5000$
Software	100$		3	300$
Total Hardware and Software			**5300$**	
Total Hardware and Software with risk			**5310$**	
Other Costs				
Utility payments	400$		3	1200$
Renting equipment	500$		10	5000$
Marketing campaigns	100$		10	1000$
Total Other Costs			**7200$**	
Total Other Costs with risk			**7210$**	
TOTAL			**29,730$**	
TOTAL with risk			**32,703$**	

Table 4.3.1 Project Costs

4.4 Managing Project Costs

Budgets are typically created in the initiation and planning phases of any project. As with any other project management document, you need to continue to review and control the budget throughout the life cycle of your project.[1]

All previous advises help our team to describes how we can manage and control spending during the project. It also helps to presents our cash flow, as well as the expected financial income from the project and its expenses.

We apply all of these advises of managing the cost of our project for two main reasons:

The first reason is: it allows us to make sure that we have available money when we need it.[2]

The second reason is: make sure that the project expenditures do not exceed the specified budget, and if that happens, we will take appropriate measures to deal with this problem.[2]

This information allows our project manager to determine the amount brought forward and the financial needs of a project, as illustrated in table below:

- The first group combines the expenses over time.
- The second group records the expected incomes.
- The third group determines the financial needs. To illustrate, "Balance" is the net balance at the end of the project, computed as the difference between incomes and expenses. For "Financial Need" it indicates how much money the project needs to carry out activities. It come from adding the balance of the current period to the credits and debits accumulated.[2]

	Q1	Q2	Q3	Q4	Total
Expense					
Expense 1	10,000$	30,000$	50,000$	10,000$	**100,000$**
Expense 2	20,000$	40,000$	60,000$		**120,000$**
Total Expense	**30,000$**	**70,000$**	**110,000$**	**10,000$**	**220,000$**
Income					
Payment	**50,000$**			**200,000$**	**250,000$**
Total Incomes	**50,000$**	**0$**	**0$**	**200,000$**	**250,000$**
Balance	**20,000$**	**-70,000$**	**-110,000$**	**190,000$**	**30,000$**
Financial need		**-50,000$**	**-180,000$**		

Table 4.41 Budget Structure

Finally at the end of the meeting one of the experts told us: "Budgeting in the project management world is a complex process involving many different parties and documentation but following the best practices can help break it down and can help set you up for budget management success".[1]

Chapter 5: Risk Management Plan

As we mentioned earlier, we are responsible for developing the smart fire alarm system for the commercial complexes. During the work in this project, we received a phone call that told us the price of sensor shipment that is required to integrate them with developing smart fire alarm system is increased by 30% as a result of imposing a new tax on the exporting sensors! Unfortunately, this issue threatens the project's ability to be completed within the allocated budget, but we are lucky; we already prepared for this issue.

Months earlier, before we worked on the project, and after the scope management plan, schedule management plan, and cost management plan were completed, we were willing to make the risk management plan because we knew that no project is risk-free and because we understood that the risk management plan is a core part of project management. As far as we know, the risk management areas include three main steps: risk identification, analysis and prioritization of risks, as well as a mitigation plan.

Figure 5.1 Risk Management Life Cycle

5.1 Risk Identification

We used one of the most effective techniques for identifying possible risks, which is brainstorming. We made sure the people involved in this discussion had skills and experience from other similar projects. The more diverse teams in this session the more perspectives that help to identify risks that we alone may not have thought of. We brainstormed and asked ourselves what could hurt the project goals and answered by using if/then statements, like if a given event happens then there is how the project is impacted, and we used also one of efficient tools that is called risk register. The risk register is a table that contains a list of potential risks that are collected during brainstorming sessions. And guess what?! We determined this issue—increasing the price of sensor shipment—during brainstorming and wrote it in a risk register alongside the worry about the possibility that the only software architect may pull out of the project, which is considered a single point of failure, and more and more risks that are generated as a result of the internal and external dependencies of the project.

We used another useful tool during brainstorming that is called a cause-and-effect diagram (also called a fishbone diagram) that shows the possible causes of a risk and sorts them into useful categories such as people, technology, materials, transportation, money, time, and environment. This tool helps us determine the main root cause of this risk so we can focus on it later in developing the mitigation plan. The following diagram was created by using a fishbone diagram style during the risk identification steps:

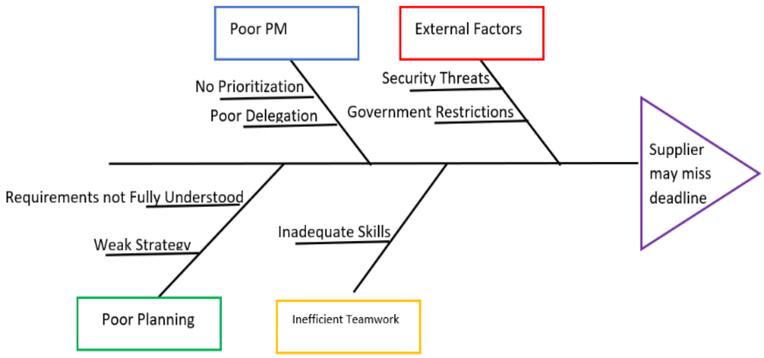

Figure 5.1.1 Fishbone Diagram

It is important to remember that even with these powerful tools, there are an endless number of risks, and it is very difficult to count and plan against them all, so the next step in risk management will help us prioritize them and focus on the most important ones.

5.2 Analysis and Prioritization The Risk

We used in this phase a probability and impact matrix to assess and prioritize risks. This is a great tool that is used to calculate the risk exposure using ordinal data. There are some risks that we identified in the previous step of developing a smart fire alarm system. The first is that increasing price of sensor shipment and the second is the possibility that the only software architect may pull out of the project. So, we prioritized between these risks by using a risk exposure methodology that follows these steps:

First of all, we as a team determined the possibility of occurrence of every risk; according to the first risk, we estimated it to be medium, and the second risk, we estimated it to be low.

Probability		
	Qualitative	Quantitative (if measurable)
Low	Very low chance of risk occurring.	Less than <10% chance of risk occurring.
Medium	Medium chance of risk occurring.	10%-49% chance of risk occurring.
High	High chance of risk occurring.	50%-100% chance of risk occurring.

Figure 5.2.1 Probability Chart

Secondly, we use the impact matrix that is different from company to company to determine the risk impact on the project; the following matrix is specific to our company:

Types of Impact	Low	Medium	High
Financial	Low financial impact, costing the company $0-$14,000	Medium financial impact, costing the company $15,000-$29,000	High financial impact, costing the company $30,000 or more
Operational	Low impact to project operations, causing delays of a few days to a few weeks	Medium impact to project operations, with potential to delay project by a month or more	High impact to project operations, with potential to cause project failure
People	Low impact to employee attrition, with 5%+ of employees quitting	Medium impact to employee attrition, with 25%+ of employees quitting	High impact to employee attrition, with 50%+ employees quitting

Figure 5.2.2 Impact Chart

By using this matrix, we decided the first risk has a high impact on our project and the second risk has a high impact, too.
Last step we did it is that using conversion matrix to assess risks as (low, medium, high) in our project. The following is the conversion matrix:

Inherent Risk				
		Impact		
		Low	Medium	High
Probability	High	Medium	High	High
	Medium	Low	Medium	High
	Low	Low	Low	Medium

Figure 5.2.3 Probability and Impact Matric

As we see the first risk has high rank and the second risks has medium rank, so we focused on these risks and made plan to mitigate their effect on the project.

5.3 Development Mitigation Plan

We planned for the risks that would have a high and medium impact on developing the smart fire alarm system and how to handle them if they actually materialized. To elaborate more on the process in this step, let's take the risk of increasing the price of sensor shipment as a result of imposing a new tax on the exporting of sensors as a case study here. We make four different types of plans to handle this risk, the first plan is avoiding plan, this plan states that we can avoid this risk totally by looking for another country supplier that doesn't have a tax on the exporting items. The second plan is minimizing plan, this plan states we can buy some sensors that we want from the country that has tax on its stuff and buy the remine amount from another country that doesn't have this tax. The third plan is a transfer plan that states we could find another supplier that buys from the country that already has a tax and purchase from them instead. The fourth plan is acceptance, which states we totally accept the risks and put extra funds to pay for them.

Figure 5.3.1 Risk Type

Now, let's go back to our problem and choose between these four plans to mitigate the risk that occurred. We decide to choose the accepting plan since we are developing a critical system and need these high-quality sensors that we can't find anywhere else, and the stakeholders in the project agree to buy more to have a robust system.

The output of the risk management is the following denouement:

FIRE ALARM
SYSTEM

Operations & Training Risk Management Plan

Author: Jana, Salwa, Samah, Sana
Status: Draft / Final

Created: Dec19
Updated: Dec19

Objective

The objective of this document is to outline the risks to Fire Alarm System operations and develop a plan to mitigate those risks.

Executive Summary

In normal circumstances, we would expect to complete the Fire Alarm System operations within one year. This includes don't going over the project budget, and don't behind the training schedule. See below for risks to this project, and the mitigation plan to address them.

Scenario	Risk to project (L/M/H)	Mitigation Plan
the price of sensor shipment that is required to integrate them with developing fire alarm system is increased by 30%.	**high**	Accept the risk. Buying the sensors with 30% additional cost.

Chapter 6: Measure Project Progress

6.1 Earned Value Analysis

As we mentioned earlier, the fire alarm system will be developed within one year, so now that the project has lasted for four months, it is the best time to measure our progress in the project to avoid any early problems. We collected data about every task in the project whatever using direct measures like lines of code in term of software size and indirect measures like requirements stability to find the percentage of the competence, and its actual cost, also we try to find data that in ratio type to help us in having occurrent measuring, then we made sure the collecting data is valid and now we will analyze this data by using Earn Value Analysis Techniques with supporting WBSs in the previous activities to track both schedule and cost of the project, we want to pay the attention on that the WBSs in the previous activities are divided by using the 40 stuff-Hour Rule of Thumb. The following table present the collecting data:

									Here								
Activity	Budget	Actual Cost	Actual Percent Complete of Tasks (%) not The Supposed One to Complete	Earn Value	Plan Value	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Details Requirment	800,000	1,000,000	100%	800,000	800,000												
Design	700,000	560,000	80%	560,000	420,000												
Implementation	650,000	0	0%	0	0												
Unit Test	970,000	0	0%	0	0												
Integration Test	970,000	0	0%	0	0												
System Test	970,000	0	0%	0	0												
Cumulative Value	5060000	1,560,000		1,360,000	1,220,000												

Table 6.1.1 Earned Value Analysis (1)

Term	Value		comment
PV	1,220,000		
CV	-200000		Over the budget
SV	140000		Ahead of schedule
SPI	1.1147541		Ahead of schedule
CPI	0.87179487		Over the budget
BAC	5,060,000		
ETC	3,807,223		
EAC	5367223.18		
VAC	-307223.183		

*The project ahead of the schedule and over the budget

Table 6.1.2 Earned Value Analysis (2)

To find how much we are ahead of the schedule from PV value, we can divide PV for individual / the average salary of manpower per hour, assuming it is 1.7$/h in this project with having 110 men, so we are ahead of schedule by (140000/110*1.7) = 748.6 hours --> 31.19 days --> one month. We conclude that we are one month ahead of schedule and $200,000 over budget.

Chapter 7: Conclusion and References

Conclusion

At the end we applied all what we learned in Software Project Management SE464 in our project starting from identifying the scope, the schedule, the cost, the risks till measuring the progress. We managed the project in detail each stage using many techniques that were mentioned in our lectures and learned more information when applying these concepts during our project.

References

[1] “Project Planning: Putting it all together,” *Coursera*. [Online]. Available: https://www.coursera.org/learn/project-planning-google. [Accessed: 18-Dec-2022].

[2] A. Villafiorita, *Introduction to software project management*. Auerbach Publications, 2016.

[3] “Function Point to Lines of Code Conversion,” *Pabipedia*, 06-Mar-2010. [Online]. Available: http://pabipedia.wikidot.com/function-point-to-lines-of-code-conversion. [Accessed: 20-Dec-2022].