SOFTWARE PROJECT MANAGEMENT LAB 4

SUNIL TUMKUR 100620430
ERIC TSIM 100560172
HARSHWARDHAN GOSWAMI 100654736

TEAM NAME: OW

TEAMS

1) System Analyst: Requirements Gathering, Requirements Elicitation and Analysis
2) Software Developers: Design & Implementation
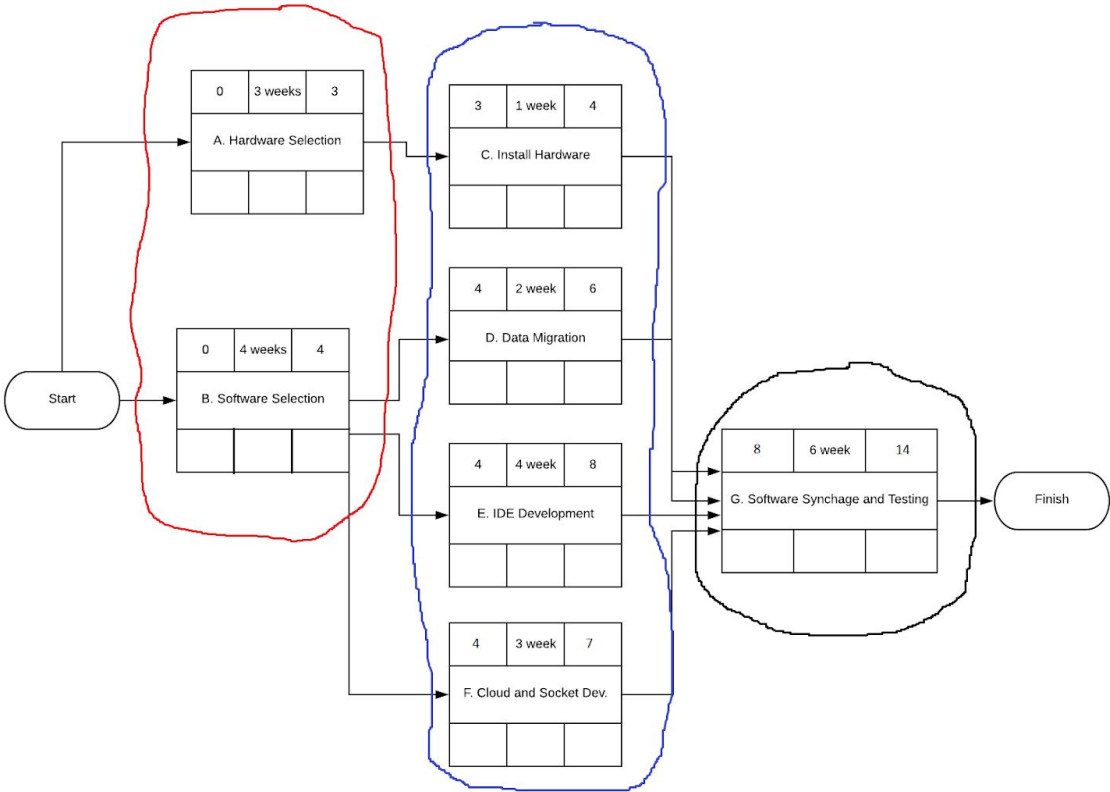3) Software Testers: Testing

RISK MANAGEMENT

Risk management is the process of identifying, assessing and controlling threats to an organization's capital and earnings. The easiest way to define is risk however is through the expectation of loss. There are risks in which we can foresee, and prepare for, and also risks where it cannot be estimated and controlled. As a result it is important for businesses to have strong risk mitigation strategies and plans in place to prepare for all possible and impossible strategies.

Some risks and countermeasures:

RISKS RELATED TO DESIGN	COUNTERMEASURES
-Employee turnover: In every project we have a certain number of developers working on a particular task. At some point, it is possible for a developer to take leave and be absent. This would take away all the knowledge that they have from that task and possibly hamper a project.	A countermeasure could be making sure that the business/company ensures that there are enough resources available and divided between teams. Dividing the resources between teams so each developer can communicate, collaborate, and share their knowledge is beneficial to the project's growth and success.
RISKS RELATED TO REQUIREMENTS GATHERING	**COUNTERMEASURES**
-Incomplete Requirements: Requirements that are unstable, ambiguous or unclear can lead to faulty requirements. Changes that are later being made in the process can have a significant effect on the total cost of the project, as it wasn't initially	A countermeasure could be making sure that the developers and stakeholders schedule a meeting to go over the requirements documentations and lay out the frameworks that are easy to read, understand and reduce the risk factor. As a result

mentioned and documented.	this may or may not increase the project delivery schedule, and affect the budget of the project.
RISKS RELATED TO REQUIREMENTS GATHERING	**COUNTERMEASURES**
-Design Compromising: Developers may rush certain parts of a design process such as the data structure and modularity. This may occur in the development of the software which can lead to threats and vulnerabilities in the future of the software.	A countermeasure could vary as it is based on the staff that is responsible for the design process of the software being developed. It is important for the team to prepare meetings that will make use of flowcharts, and different data structures in order to get a sense of what end goal you want for the software you are developing. The best way to battle design risks is to use modularity and make sure that the software is broken down into parts that can be tested as developed which would erase code repetition.

RESOURCES

1. Sunil Tumkur (RED)
   Role: Analyst
   Duties: Gather system requirements, requirement elicitation and analysis of both hardware and software

2. Harsh (BLUE)
   Role: Full Stack Developer
   Duties: Implementation

3. Eric (BLACK)
   Role: Software Tester
   Duties: Testing the code

2 Mitigation Tasks

1) Security and 100% availability of the socket for communication is needed to ensure a more ideal software.
2) Requirements may not be fully defined. For example, as a restaurant adds more items to the menu, the expandability of the storage will need to be considered

A. Latest start: 4 - 3 = 1 weeks, Float = 4 - 0 - 3 = 1 weeks
B. Latest start: 4 - 4 = 0 weeks, Float = 4 - 0 - 4 = 0 weeks
C. Latest start: 8 - 1 = 7 weeks, Float = 8 - 3 - 1 = 4 weeks
D. Latest start: 8 - 2 = 6 weeks, Float = 8 - 4 - 2 = 2 weeks
E. Latest start: 8 - 4 = 4 weeks, Float = 8 - 4 - 4 = 0 week
F. Latest start: 8 - 3 = 5 weeks, Float = 8 - 4 - 3 = 1 week
G. Latest start: 14 - 6 = 8 weeks, Float = 14 - 6 - 8 = 0 week

Microsoft Project Professional Screenshots

		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1			▲ Software and Hardware Requirement Elicitation	20.13 days	Mon 20-02-03	Tue 20-03-03		
2			Requirement Elicitation Meeting	1 hr	Mon 20-02-03	Mon 20-02-03		
3			Hardware Selection	3 wks	Mon 20-02-03	Tue 20-02-25	2	Sunil Tumkur
4			Software Selection	4 wks	Mon 20-02-03	Tue 20-03-03	2	Sunil Tumkur
5			▲ Implementation	25 days	Tue 20-02-25	Tue 20-03-31		
6			Initial Hardware	1 wk	Tue 20-02-25	Tue 20-03-03	3	Harsh Goswami
7			IDE Development	4 wks	Tue 20-03-03	Tue 20-03-31	4	Harsh Goswami
8			Cloud and Socket Dev.	3 wks	Tue 20-03-03	Tue 20-03-24	4	Harsh Goswami
9			Data Migration	2 wks	Tue 20-03-03	Tue 20-03-17	4	Harsh Goswami
10			▲ Testing	30 days	Tue 20-03-31	Wed 20-05-13		
11			Software Syncronization and Testing	6 wks	Tue 20-03-31	Wed 20-05-13	6,9,7,8	Eric Tsim