



SOFE 3490U-001
Software Project Management
Lab 4

Fall Monitor

Github group # (Group 101)

Priyadharshini Ramalingam 100670614

Pranjal Saloni 100653360

Alex Dafoe 100599423

Risk Analysis

Risk #1: Scope Creep

- Uncontrolled change to the project's scope; urgent projects may be attempted on a best effort basis that neglects rigorous management of project change.

Countermeasure: Assign teams to every project so each project has people working on it with undivided attention.

Risk #2: Requirements risks

- Requirements may not be understood properly by every member in the team.

Countermeasure: In the very beginning, thoroughly analyze the whole project and put down all potential problems that may occur in the future.

- Client / Stakeholder requirements may change.

Countermeasure: Proper understanding of the product within the team and communication amongst team members. Inform clients about everything that can somehow affect the project, including potential risks.

Risk #3: Dependency risks

- Some fall detectors might behave slightly differently depending on the smartphone model / device in which it is installed.

Countermeasure: Make the product scalable, so it can be run on all platforms.

Risk #4: Technical risks

- Smartphones cannot be overloaded with continuous sensing commitments that undermine performance of the phone, i.e., by depleting battery power.

Countermeasure: It is essential to manage the sleep cycle of sensing components in order to trade off the amount of battery consumed.

- The product may be too complicated to build.

Countermeasure: Brainstorm product specifications, and steps that need to be taken to complete the project with all the team members involved. Hire experienced individuals for development. Have trusted employees be in charge of every task. Keep constant communication within the team.

- Errors may be overlooked or missed during the testing phase.

Countermeasure: Have sufficient time assigned for debugging and testing the product.

- To save time, developers may sometimes neglect design processes.

Countermeasure: Getting started early on the design process, and making sure that the developers have full knowledge of the product requirements and specifications.

Risk #5: Budget

- Actual cost may exceed target cost, cost overrun and / or inaccurate budget estimation.

Countermeasure: Having a spreadsheet of all the cost estimations, resources, historical data etc. before the start of the project. Review the estimation process before the development stage starts. Thoroughly plan the project ahead of the time, stick to planned scope and closely monitor progress at every stage.

Risk #6: Security / privacy

- Fall detection can lack strategies to ensure data privacy.

Countermeasure: Making the product with more security, and privacy policies.

Risk #7: Employee, staff and resources

- Employees may lack productivity. Some deadlines may not meet.

Countermeasure: Running a strict recruitment process, having a group of people be monitored by their bosses could avoid certain risks.

- Resources such as systems, staff, skills of employees may stay untracked.

Countermeasure: Performing proper tracking at early stages.

- Employees may lack knowledge of the product and their responsibilities.

Countermeasure: Train each employee, explain each team member their scope of work and responsibilities.

Risk #8: Market risks

- Can include competition, foreign exchange, commodity markets, and interest rate risk, as well as liquidity and credit risks.

Countermeasure: Study the market before getting started on the project; plan cost estimation etc. based on it.

Risk #9: Unavoidable risks

- Risk #15: These are risks, which can't be controlled or estimated; starting from technologies being discontinued to even changes in government policy.

Risk Assessment

Risk #2: Requirements risks

Countermeasures and tasks:

Task 1: Have an agile/scrum meeting with stakeholders, so everyone in the team understands what the product requirements are.

Task 2: Thoroughly analyze the whole project and put down all potential problems that may occur in the future.

Task 3: Have frequent discussions with the customer so if there are any changes in the requirements, the team is notified.

Task 4: If there are changes, go back on the meeting table and record those changes. The team leader must explain those changes to the team.

Task 5: Perform user testing prior to putting the product out in the market to ensure all requirements are met.

Task 6: Go back and fix, if requirements are not met.

Risk #7: Employee, staff and resources

Countermeasures and tasks:

Task 1: Develop training material (such as handbooks etc.) beforehand. Set up a team to develop and host proper training essentials and finalize training period with the team.

Task 2: Make training mandatory for all employees. And have a time set (2 weeks) within which all employees must finish training.

Task 3: Have an assessment to ensure training is completed.

Task 4: If training is missed by any employee or if training is not completed, have backup training days.

Task 5: Once training is completed, assign tasks and deadlines to each individual employee.

Task 6: Track their responsibilities, performances and deadlines. Have it all monitored by the team leader.

Resources

The Fall Monitor project we have undertaken requires various resources to uphold the incremental development plan and schedule previously outlined in the last two reports. In terms of labour, our team requires two developers, a project manager, a system analyst and a software tester. To create the software for this project. Their tasks are detailed below.

Developers-2

Description: Developers write code and develop the end product.

Tasks:

- Creating Database System
- Writing Documentation
- Writing Unit Tests
- Coding of Application

Project Manager-1

Description: Project manager manages the team and completion of the product, overall responsibility for the successful initiation, planning, design, execution, monitoring, controlling and closure of a project.

Tasks:

- Requirements gathering
- Recruiting
- Meeting with client
- Risk Avoidance
- System Architecture Design (Planning)
- Acceptance Testing

System Analyst-1

Description: System analyst uses analysis and design techniques to solve business problems using information technology.

Tasks:

- Requirements gathering
- System Architecture Design/Software Selection (Planning)
- Hardware Selection

- Risk Avoidance

Technical Lead-1

Description: Leads a development team, and is responsible for the quality of its technical deliverables.

Tasks:

- Quality assurance
- Requirements gathering
- Unit Testing
- Writing Internal Documentation
- Coding of application
- End user documentation
- User training
- System architecture design
- Hardware deployment

Software Testers-1

Description: Software testers are quality assurance experts who put applications through the wringer to root out bugs, poor performance and funky interface issues.

Tasks:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

Association and Deadlines:

Developer and Software Tester are both involved in the quality assurance stage of software development and deployment. Conducts automated and manual tests to ensure the software created by developers is fit for purpose. Software testing involves the analysis of software, and systems, to avert risk and prevent software issues and develop and implement applications and programs for the backend processing systems.

System Analyst and Technical Lead has the knowledge of computer systems and technologies. They work together to manage the ability to communicate with clients, colleagues and management to explain complex issues clearly and concisely. A logical, analytical and creative approach to problems and to work both in a team and alone and to manage your own workload, career motivation and a willingness to continue to further your knowledge and skills.
























Project manager takes charge of every step in the making, works with all other team members and takes care of documentation and team planning

Deadline to develop the code by the developer and the technical lead to analyse it and software testers to run multiple testings and the project manager to manage complex issues takes a total of 3 months to finish.

- Days to finish backend processing and fixing the code by developers - 33 days
- Days to finish the testing by software testers - 18 days
- Days to finish unit testing, writing internal documentation, coding of application, end user documentation, user training, system architecture design, hardware deployment by the technical lead - 58 days
- Day to finish system architecture design and working with clients by the project manager - 31 days

*** Activities and Gantt chart below ***

Activities and resources:

		Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish ▾	Predecessors ▾	Resource Names
1			Requirements Gathering	1 wk	Mon 20-02-03	Fri 20-02-07		Technical Lead,System Analyst, Project Manager A[25%]
2			Recruiting	2 wks	Mon 20-02-03	Mon 20-02-17		Project Manager A[25%]
3			Hardware Selection	1 wk	Thu 20-02-27	Wed 20-03-04	1,2	System Analyst
4			Hardware Deployment	6 days	Tue 20-04-07	Tue 20-04-14	3	Technical Lead[50%]
5			Create Database Schema	1 wk	Mon 20-02-10	Mon 20-02-17	1	Developer A,Developer B
6			Database setup	3 days	Tue 20-02-18	Thu 20-02-20	5	Developer A,Developer B
7			Risk Mitigation - Budget overruns	1 day	Wed 20-02-26	Wed 20-02-26	9	Project Manager A,System Analyst
8			Risk Mitigation - Changing requirements	1 day	Tue 20-02-25	Tue 20-02-25	9	Project Manager A,System Analyst
9			Design System Architecture	2 wks	Mon 20-02-10	Mon 20-02-24	1	Technical Lead,System Analyst, Project Manager A[75%]
10			 Program Application	30 days	Tue 20-02-25	Mon 20-04-06	9	Developer A,Developer B, Technical Lead,Software Tester A
11			Write Internal Documentation	6 wks	Tue 20-02-25	Mon 20-04-06	9	
12			Unit Testing	6 wks	Tue 20-02-25	Mon 20-04-06	9	
13			Code Application	6 wks	Tue 20-02-25	Mon 20-04-06	9	
14			Write End User Documentation	3 wks	Tue 20-04-07	Mon 20-04-27	13	Technical Lead[50%]
15			User Training	3 days	Wed 20-04-2	Fri 20-05-01	17	Technical Lead[50%]
16			Acceptance Testing	2 days	Wed 20-04-2	Thu 20-04-30	17	Software Tester A,Project Manager A
17			System Testing	5 days	Wed 20-04-2	Tue 20-04-28	18	Software Tester A
18			Intergration Testing	1 wk	Wed 20-04-1	Tue 20-04-21	4,6,12,13	Software Tester A

Gantt chart:

