Terna Engineering College

Computer Engineering Department

Program: Sem VI

Course: Software Engineering Lab

LAB Manual

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No.11

A.1 Aim:

To prepare RMMM Plan for selected mini-project.

A.2 Prerequisite:

Knowledge about requirement engineering processes.

A.3 Outcome:

After successful completion of this experiment, students will be able to identify risk and document the RMMM (Risk Mitigation, Monitoring and Management) Plan.

A.4 Theory:

Introduction about Risk

Risk: A risk is a potential problem – it might happen and it might not

- Conceptual definition of risk
- Risk concerns future happenings
- **Risk involves a change in mind, opinion, actions, places, etc.**
- Risk involves choice and the uncertainty that choice entails

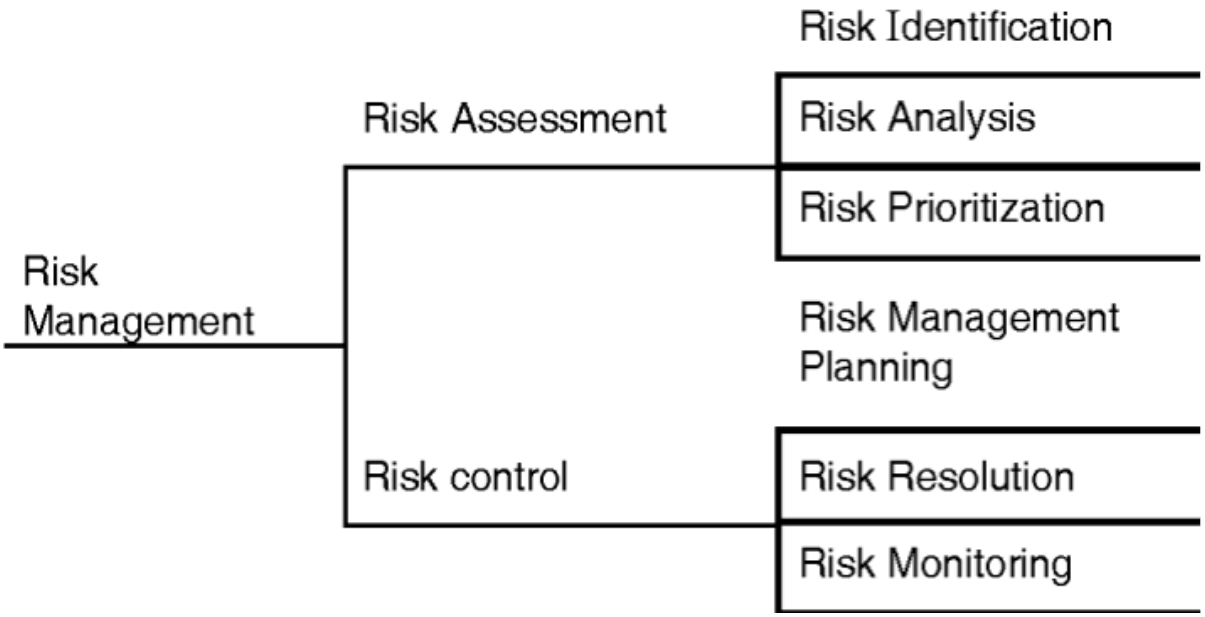

Fig: Risk management activities

Types of Risk:

Risk	Threats	Identify	Consequences
1. Project	Project plan	Potential budgetary, schedule, personnel (staffing and organization), resource, customer, and requirement problems and there on a software project.	The project
2. Technical or Product	Quality and timeliness of software to be produced	Potential design implementation, interface, verification, and maintenance problems; Specification ambiguity, technical uncertainty, technical obsolescence	The implementation may become difficult or impossible; the Quality or performance of the product decrease.

3. Business	Viability of software to be built	(i) Market Risk: Building an excellent product or system that no one wants. (ii)Strategic Risk: Building a product that no longer fits into the overall business strategy for the company. (iii)Management Risk: Losing support of senior management due to a change in focus or a change in focus or a change in people. (iv) Budget Risk: Losing budgetary or personnel commitment. (v) Building a product that the sales force doesn't understand how to sell.	Often jeopardize the projector

Risk Analysis

- Risk analysis includes a series of steps that allows software development teams to identify, understand and manage involved risks.
- Risk analysis refers to the identification and understanding of such problems and attempts to find out steps that need to be taken in case of risk carries during the development process.

The process of risk analysis includes the following steps:

1. **Risk identification**
2. **Risk Assessment**
3. **Risk Management**

   a. Avoiding the risk
   b. Reducing damage caused by the risk
   c. Accepting the risk

Risk identification is a systematic attempt to specify threats to the project plan (estimates, schedule, resource loading, etc.). By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.

There are two distinct types of risks for each of the categories that have: **Generic risks and product-specific risks.**

Generic risks are a potential threat to every software project.
Product-specific risks can be identified only by those with a clear understanding of the technology, the people, and the environment that is specific to the project at hand.
One method for identifying risks is to create a risk item checklist

Risk Mitigation, Monitoring and Management (RMMM Plan):

- The RMMM plan may be a part of the software development plan or maybe a separate document.
- Risk mitigation is a problem **avoidance activity**
- Risk monitoring is a project **tracking activity**

Risk monitoring has three objectives

- To assess whether predicted risks do occur
- To ensure that risk aversion steps defined for the risk are being properly applied
- To collect information that can be used for future risk analysis

- The findings from risk monitoring may allow the project manager to ascertain what risks caused which problems **throughout the project**
- **Risk mitigation (avoidance) is the primary strategy and is achieved through a plan**

Risk item checklist

Product size risks

1. Estimated size of the product in LOC or FP?
2. Degree of confidence in estimated size estimate?
3. Estimated size of product in several programs, files, transactions?
4. Percentage deviation in the size of product from average for previous products?
5. Size of the database created or used by the product?
6. Several users of the product?
7. Several projected changes to the requirements for the product? Before delivery? After delivery?
8. Amount of reused software?

Business impact risks

9. Effect of this product on company revenue?
10. Visibility of this product by senior management?
11. Reasonableness of delivery deadlines?
12. Several customers, who will use this product and the consistency of their needs relative to the product?
13. Several other products/systems with which this product must be interoperable?
14. The sophistication of end-users?
15. Amount and quality of product documentation that must be produced and delivered to the customer?
16. Governmental constraints on the construction of the product?
17. Costs associated with late delivery?
18. Costs associated with a defective product?

Customer-related risks

19. Have you worked with the customer in the past?
20. Does the customer have a solid idea of what is required?
21. Has the customer taking the time to write this down?
22. Will the customer agree to spend time in formal requirements gathering meetings to identify the project scope?
23. Is the customer willing to establish rapid communication links with the developer?
24. Is the customer willing to participate in reviews?
25. Is the customer technically sophisticated in the product area?
26. Is the customer willing to let your people do their job, that is, will the customer resist looking over your shoulder during technically detailed work?
27. Does the customer understand the software engineering process?

Development environment risks

28. Is a software project management tool available?
29. Is a software process management tool available?
30. Are tools for analysis and design available?
31. Do analysis and design tools deliver methods that are appropriate for the product to be built?
32. Are compilers or code generators available and appropriate for the product to be built?
33. Are our testing tools available and appropriate for the product to be built?
34. Are software configuration management tools available?
35. Does the environment make use of a database or repository?
36. Are all the software tools integrated?

37. Have members of the project teams received training in each of the tools?
38. Are local experts available to answer questions about the tools?
39. Is online help and documentation for the tools adequate?

Process issue risks

40. Does your senior management support a written policy statement that emphasizes the importance of a standard process for software development?
41. Has your organization developed a written description of the software process to be used on this project?
42. Our staff members signed up for the software process as it is documented and willing to use it?
43. Is the software process used for other projects?
44. Has your organization developed or acquired a series of software engineering training courses for managers and technical staff?
45. Are published software engineering standards provided for every software developer and software manager?
46. Have document outlines and examples been developed for all deliverables defined as part of the software process?
47. Are formal technical reviews of the requirements specification, design, and code conducted regularly?
48. Are formal technical reviews of test procedures and test cases conducted regularly?
49. Are the results of each formal technical review documented, including defects found and resources used?
50. Is there some mechanism for ensuring that work conducted on a project conforms with software engineering standards?
51. Is configuration management used to maintain consistency among system/software requirements, design, code, and test cases?
52. Is a mechanism used for controlling changes to customer requirements that impact the software?
53. Is there a documented statement of work, software requirements specification, and software development plan for each subcontract?
54. Is a procedure followed for tracking and reviewing the performance of subcontractors?

Staff size and experience

55. Are the best people available?
56. Do the people have the right combination of skills?
57. Are enough people available?
58. Are staff committed for the entire duration of the project?
59. Will some staff be working only part-time on this project?
60. Do staff have the right expectations about the job at hand?

61. Have staff received the necessary training?
62. Will turnover among staff be low enough to allow continuity?

Technical issue risks

63. Are facilitated application specification techniques used to aid in communication between the customer and developer?
64. Are specific methods used for software analysis?
65. Do you use a specific method for data and architecture designs?
66. Is more than 90% of your code written in a high order language?
67. Are specific conventions for code documentation defined and used?
68. Do you use a specific method for test case design?
69. Are software tools used to support software planning and tracking activities?
70. Our configuration management software tools used to control and track change activity throughout the software process?
71. Are software tools used to support the software analysis and design process?
72. Are tools used to create software prototypes?
73. Are software tools used to support the testing process?
74. Are software tools used to support the production and management of documentation?
75. Our quality metrics collected for all software projects?
76. Are productivity metrics collected for all software projects?

Technology risks

77. Is the technology to be built new to your company?
78. Do the customer requirements demand the creation of new algorithms, input or output technology?
79. Does the software interface with new or unproven hardware?
80. Does the software to be built interface with a database system whose function and performance have not been proven in this application area?
81. Does the software to be built an interface with vendor-supplied software products that are unproven?
82. Is a specialized user interface demanded by-product requirements?
83. Do requirements for the product demand the creation of program components that are unlike any previously developed by your organization?
84. Do requirements demand the use of new analysis, design, or testing methods?
85. Do requirements demand the use of unconventional software development methods, such as formal methods, AI-based approaches, artificial neural networks?
86. Do requirements put excessive performance constraints on the product?

87. Is the customer uncertain that the functionality requested is “doable”?

Risk information sheet			
Risk ID : P02-4-32	Date : 5/9/02	Prob : 80%	Impact : high
Description : Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.			
Refinement / context : **Subcondition 1 :** Certain reusable components were developed by a third party with no knowledge of integral design standards. **Subcondition 2 :** The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components. **Subcondition 3 :** Certain reusable components have been implemented in a language that is not supported on the target environment.			
Mitigation / monitoring : 1. Contact third party to determine conformance with design standards. 2. Press for interface standards completion; consider component structure when deciding on interface protocol. 3. Check to determine number of components in subcondition 3 category; check to determine if language support can be acquired.			
Management / contingency plan/trigger : RE computed to be $20,200. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly. Trigger : Mitigation steps unproductive as of 7/1/02			
Current status : 5/12/02 : Mitigation steps initiated.			
Originator : Asmita T.		Assigned : Vaishali Wadghare.	

Fig. 1.43 : Risk information sheet (RIS)

Examples of Different Risk and RMMM actions:

1. **Risk: Late Delivery**
   - **Mitigation:** The cost associated with a late delivery is critical. Late delivery will result in late delivery of a letter of acceptance from the customer. Without the letter of acceptance, the group will receive a failing grade for the course. Steps have been taken to ensure timely delivery by gauging the scope of the project based on the delivery deadline.
   - **Monitoring:** A schedule has been established to monitor project status. Falling behind schedule would indicate a potential for late delivery. The schedule will be followed closely during all development stages.
   - **Management:** Late delivery would be a catastrophic failure in the project development. If the project cannot be delivered on time the development team will not pass the course. If it becomes apparent that the project will not be completed on time, the only course of action available would be to request an extension to the deadline from the customer.

2. **Risk: End Users Resist System**
   - **Mitigation:** To prevent this from happening, the software will be developed with the end-user in mind. The user interface will be designed in a way to make use of the program convenient and pleasurable.
   - **Monitoring:** The software will be developed with the end-user in mind. The development team will ask the opinion of various outside sources throughout the development phases. Specifically, the user-interface developer will be sure to get a thorough opinion from others.
   - **Management:** Should the program be resisted by the end-user, the program will be thoroughly examined to find the reasons that this is so. Specifically, the user interface will be investigated and if necessary, revamped into a solution.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	**Name:** AMEY THAKUR
Class: Comps TE B	**Batch:** B3
Date of Experiment: 09/04/2021	**Date of Submission:** 09/04/2021
Grade:	

B.1 Question of Curiosity:

1. **Identify any 5 risks associated with your selected mini-project, and specify their category.**

Ans:

The risks associated with Digital Book Store are:

- Lack Of Development Experience (Product Size Risk)
- Loss Of Database (Product Risk)
- Change of Requirements By Customers (Technical Risk)
- Late Delivery (Business Risk)
- Developing Wrong User Interface (Technical Risk)

2. **Develop a risk projection table for risk identified.**

Ans:

Risks	Category	Probability	Impact
Lack Of Development Experience	TI	60%	2
Loss Of Database	PS	50%	1
Change of Requirements By Customers	PS	80%	3
Late Delivery	BU	50%	2
Developing Wrong User Interface	DE	70%	2

	Impact Values
PS - Product Size	1. CATASTROPHIC
TI - Technical Issues	2. CRITICAL
BU - Business Impact	3. MARGINAL
DE - Development Environment	4. NEGLIGIBLE

3. Prepare an RMMM plan for risk identified.

Ans:

1. Lack Of Development Experience

- Mitigation: To prevent this from happening, the development team will be required to learn the languages and techniques necessary to develop this software. The member of the team that is the most experienced in a particular facet of the development tools will need to instruct those who are not as well versed.

- Monitoring: Each member of the team should watch and see areas where another team member may be weak. Also if one of the members is weak in a particular area it should be brought to the attention of that member, to the other members.

- Management: The members who already had good training and developed skills will help other members who don't.

2. Loss Of Database

- Mitigation: To prevent this from happening, developers who are in contact with the database, and/or use functions that interact with the database, should keep in mind the possible errors that could be caused due to poor programming/error checking. The backup of data should be primordial. A copy of all the databases must be maintained side by side on a different system; Also, it must be updated simultaneously. A recovery tool must be designed which could recover the database according to the checkpoints.

- Monitoring: Each user should be sure that the database is left in the condition it was before it was touched, to identify possible problems. The first notice of database errors should be brought to the attention of the other team members.

- Management: A special team is allocated for the recovery of the database as soon as possible. The secondary databases which were maintained as a copy must be made primary so that the working of the software is not affected.

3. Change Of Requirements By Customers:

- Mitigation: To prevent this from happening, meetings (formal and informal) will be held with the customer on a routine business. This ensures that the product we are producing, and the requirements of the customer are equivalent.

- Monitoring: The meetings with the customer should ensure that the customer and our organization understand each other and the requirements for the product.

- Management: Should the development team realize that their idea of the product requirements differs from those of the customer, the customer should be immediately notified and whatever steps necessary to rectify this problem should be taken. Preferably a meeting should be held between the development team and the customer to discuss at length this issue.

4. Late Delivery Of Product:

- Mitigation: A late delivery will result in penalties and will not receive the letter of acceptance. Steps have been taken to ensure timely delivery by evaluating the scope of the project based on the delivery deadline.

- Monitoring: A schedule has been established to monitor project status. Falling behind schedule would indicate a potential for late delivery. So the schedule must be followed closely during all development stages to avoid delayed delivery of the product.

- Management: Late delivery would be a catastrophic failure in the project development. If the project cannot be delivered on time, the development team will not pass the course. If it becomes apparent that the project will not be completed on time, the only course of action available would be to request an extension to the deadline from the customer and it will also receive a negative impression from the customer.

5. Developing Wrong User Interface

- Mitigation: The software will be developed with the end-user in mind. The user interface will be designed in a way to make use of the program convenient and pleasurable and it should meet the user's requirements.

- Monitoring: The development team will ask the opinion of various outside sources throughout the development phases. Specifically, the user-interface developer will be sure to get a thorough opinion from others.

- Management: The program will be thoroughly examined to find the reasons that this is so. Specifically, the user interface will be investigated and if necessary, revamped into a solution.

B.2 Conclusion:

We have successfully identified risks associated with the Digital Bookstore and prepared an RMMM (Risk Mitigation, Monitoring and Management) Plan.