

**GINA CODY School of Engineering and Computer Science
Department of Computer Science and Software Engineering
Concordia University
SOEN 6481 Systems Requirement. Specification
Delivery #2**

**Requirements evaluation and risk analysis
[ETR-Electronic Tool Rental]**

Table of contents

Task 1 – Identifying and finding inconsistencies in vision document	2
Defect Table	2
Inconsistency Table	5
Task 2 – Interaction matrix: Documenting conflicts	8
Task 3 – Conflict resolution	9
Task 4 – Conflict evaluation	12
Task 5 – Risk management	14
Risk Identification	14
Component Inspection	14
Risk Checklist	14
Risk Tree	14
Quantitative assessment of the risks	15
Risk control	16
Appendix	21

Task 1 – Identifying and finding inconsistencies in vision document

1. Defect Table

Time spent during inspection : 90 mins

Defect #	Location	Defect type	Classification	Description	Status	Date corrected
1	Positioning/product Position Statement (Section 2.2, Page 2)	Ambiguity	Major	Not sure if the rental tool system is a software product or web application. Since, software developed applications are client based only. Web developed applications are client-server based.		
2	Stakeholder Description/ Stakeholders summary Section 3.1, Page 3)	Omission	Major	Missing some stakeholders like competitors that are already available in the market eg HomeDepot. Failed to mention negative stakeholders such as data brokers, hackers and investors such as venture capitalists. Negative users are the one who tries to access the system illegally and tries to tamper with information.		
3	User summary/user environment (Section 3.2, Page 4)	Opacity	Major	Versions of system platforms to be used for developing rental tool web applications are not mentioned. They should be mentioned in order to keep track of future changes and in order to modify the web application in future. (This is an RD item whose dependencies are invisible.)		
4	User summary/user environment (Section 3.2,	Unintelligibility	Major	PhpStorm IDE is to be used for developing a web application not for software development. (This RD item is stated in an		

	Page 4)			incomprehensible way for Software developers who need to use PHPStorm to develop a web application).		
5	User summary/ user environment (Section 3.2, Page 4)	Contradiction	Minor	Since, the initial release of the project will work only on Google Chrome and Microsoft edge. Therefore end users are expected to have these browsers on their devices. Also, a web application might not always support the latest browser version as soon as the new browser is release into the market.		
6	Product overview/Assumption and dependencies Section 4.2, Page 6)	Omission	Minor	Network Connection: Few assumptions are missing related to network connectivity that it was assumed that a user has a stable internet connection in order to access the website.		
7	Product overview/Assumption and dependencies (Section 4.2, Page 6)	Omission	Minor	Assumptions related to deployment are missing. Resources required to host the website should be available such as Ip address, domain name).		
8	Product Features (Section 5)	Omission	Major	Feature to recover password in case end-users forget it, has not been mentioned under features.		
9	Other Product requirements (Section 6, Page 7)	Omission	Minor	More features can be added to the product regarding support of the product such as what happens if the tool does not work or if the tool breaks while the customer is using it.		
10	Other Product	Unfeasibilit	Major	Project might take more than the		

	requirements (Section 6.2, Page 7)	y		scheduled deadline to make it more secure. This RD item cannot be implemented within the assigned schedule.		
11	Other Product requirements (Section 6.2, Page 7)	Ambiguity	Minor	Secure web application from negative users has not been mentioned. And technique is not defined to protect data from hackers eg AES technique.		
12	Other Product requirements (Section 6.2, Page 7)	Noise	Minor	In Scalable, there is no information regarding server usage in the problem world feature.		
13	User summary/ user environment (Section 3.2, Page 4)	Poor modifiability	Major	Tool is dependent on the database version. If the database needs to be updated then the PHP version needs to be updated accordingly.		
14	Other Product requirements (Section 6.3, Page 7)	Omission	Major	Glossary of Terms is not added for better understanding the term.		

2. Inconsistency Table

Time spent during inspection: 90 mins

#	Location	Inconsistency type	Classification	Description	Status	Date corrected
1	S1: Introduction/Scope (Section1, Page2) The system will allow customers to	Terminology Clash	Weak	Here system and website, both refer to Electronic tool rental. Here, the same concept is		

	<p>rent tools.</p> <p>S2:Positioning/Problem Statement (Section 2.1, Page2) :Online rental tools website which acts as a one stop shop and helps in choosing cost effective rental tools, and provide options to reserve multiple tools in advance.</p>			<p>given different names in different statements.</p>		
2	<p>S3: Product Overview/Product Perspective (Section 4.1, Page 5) : The following diagram depicts three main users and how those users will interact with the system and how the data is being exchanged between them.</p> <p>S4: Stakeholder Descriptions/User Environment (Section 3.3, Page 4): To develop the project, MVC pattern will be used to isolate business logic from the user</p>	Structure Clash	Strong	<p>Here, the first statement states that the context diagram depicts how the system works and the user will interact with the system without following any pattern and the second statement defines the project will follow MVC style architecture to solve stakeholder's problems. Here, The same concept is given different structures in different statements</p>		

	interface.					
3	<p>S5: Product Features (Section 5, Page 7): The system should allow branch employees to update the status of tools as available, rented, out of order or not available.</p> <p>S6 Product Features (Section 5, Page 7): The system administrator should have the control to change the permission of a particular branch employee to make edits to the content on the website and update their role.</p>	Terminology clash	Weak	Here, update and edit depicting the same meaning however different names are used in different statements. Since editing or updating the existing information with the new information depicts the same meaning.		
4.	<p>S7: Product Features (Section 5, Page 6): All the end-users should be able to create an account with their email.</p> <p>S8: (Product features): After successful sign up, users should be able to</p>	Designation Clash	Weak	Here users and end-users are the same thing. Users and end-users are used in two statements but share the same concept.		

	login with their username or email and password.					
5.	<p>S9: Product Features (Section 5, Page 6): While making a reservation system should pop-up and display the important details to the customer that they should be at least 18 years old and should have valid government issued photo ID that is valid at the time of rental.</p> <p>S10: Product Features (Section 5, Page 6): Branch employees should have access to scheduled requests and customer information in order to validate their information such as age, physical government ID and rental details.</p>	Terminology Clash	Weak	Here, government issued photo ID and government ID both are one thing. This concept has been written in two different names in different statements.		

Other comments/recommendations:

No comments or recommendations.

Task 2 – Interaction matrix: Documenting conflicts

S1: The system will allow customers to rent tools.

S2: Online rental tools website which acts as a one stop shop and helps in choosing cost effective rental tools, and provide options to reserve multiple tools in advance.

S3: The following diagram depicts three main users and how those users will interact with the system and how the data is being exchanged between them.

S4: To develop the project, MVC pattern will be used to isolate business logic from the user interface.

S5: The system should allow branch employees to update the status of tools as available, rented, out of order or not available.

S6: The system administrator should have the control to change the permission of a particular branch employee to make edits to the content on the website and update their role.

S7: All the end-users should be able to create an account with their email.

S8: After successful sign up, users should be able to login with their username or email and password.

S9: While making a reservation system should pop-up and display the important details to the customer that they should be at least 18 years old and should have valid government issued photo ID that is valid at the time of rental.

S10: Branch employees should have access to scheduled requests and customer information in order to validate their information such as age, physical government ID and rental details.

Based on the defects and inconsistencies below is the Interaction matrix:

Statement	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total
S1	0	1	0	0	1	0	0	0	0	0	2
S2	1	0	0	0	0	0	0	0	0	0	1
S3	0	0	0	1	1000	1000	1000	1000	0	0	4001
S4	0	0	1	0	1000	1000	1000	1000	1000	1000	6001
S5	1	0	1000	1000	0	0	0	0	0	0	2001
S6	0	0	1000	1000	0	0	0	0	0	0	2000
S7	0	0	1000	1000	0	0	0	1	0	0	2001
S8	0	0	1000	1000	0	0	1	0	0	0	2001
S9	0	0	0	1000	0	0	0	0	0	0	1000
S10	0	0	0	1000	0	0	0	0	0	0	1000
Total	2	1	4001	6001	2001	2000	2001	2001	1000	1000	20,008

The matrix element:

S_{ij} has a value 1 if statement S_i conflicts with statement S_j , 0 if these statements are distinct and do not overlap, and 1000 (say) if they overlap without conflicting.

The total number of non-conflicting overlaps and conflicts is given by the quotient and remainder of the integer division of Total by 1000 = $20008 / 1000$

Total conflicts: 8

Total non-conflicting overlaps: 20

Task 3 – Conflict resolution

Following are the conflicts specified in the previous section and their corresponding operators to transform the conflicting statements:

1. **S1:** The system will allow customers to rent tools.

S2: Online rental tools website which acts as a one stop shop and helps in choosing cost effective rental tools, and provide options to reserve multiple tools in advance.

Operators	Description
Drop lower-priority statements	Drop lower-priority statements by reducing the priority of S2 which is depicting the system in detail. However, initially we want to focus only on the main goal of the system which is for customers to rent tools.
Specialize conflict source or target	The conflict can be resolved by specializing conflict source object “system” into “Electronic rental tool system”. Resolved statement: Electronic rental tool system helps customers to rent cost effective tools and provide options to reserve multiple tools in advance.

2. **S1:** The system will allow customers to rent tools.

S5: The system should allow branch employees to update the status of tools as available, rented, out of order or not available.

Operators	Description
Avoiding boundary condition	The boundary condition for strong conflict was seen in the case when a branch employee is updating the status of a tool while a customer is renting the tool. Avoiding this

	boundary condition might be achieved by keeping the content session locked for a time period when the branch employee is editing it and the customer gets the notification to wait in order to rent the tool.
Restore Conflicting statements	This conflict can be resolved by restoring conflicting statements by allowing customers to rent tools only after branch employee's notification.

3. **S3:** The following diagram depicts three main users and how those users will interact with the system and how the data is being exchanged between them.

S4: To develop the project, MVC pattern will be used to isolate business logic from the user interface.

Operators	Description
Specialize conflict source or target	<p>This conflict can be resolved by Specializing the Conflict Source by highlighting the context diagram which depicts how the users are interacting with the system and how data is flowing between them using MVC pattern.</p> <p>Resolved Statement: The context diagram depicts how the system works as a whole, highlighting the high-level features and isolating business logic from user interface using the MVC pattern.</p>
Drop lower-priority statements	Drop lower-priority statements by reducing the priority of specifications of Project tools and technology to be used for the project since initially we want to focus more on highlighting the overall functionality of end users.

4. **S7:** All the end-users should be able to create an account with their email.

S8: After successful sign up, users should be able to login with their username or email and password.

Operators	Description
Avoiding Boundary Condition	Avoid Boundary Condition by providing a separate login and sign up portal for customers, branch employees and system administrators which will be verifying the credentials from separate tables stored in the database.
Weakening Conflicting Statements	<p>Weaken Conflicting statements by introducing the new requirement of adding “Type of User” option to differentiate the different end user’s Login Page. Assigning unique UserID to every user and maintaining their user type in the database and defining their particular roles and responsibilities.</p> <p>Resolved Statement would be “ All the end-users should be able to create an account with their username, password and Type of user after which they will be prompted to a particular assigned portal within their defined role”.</p>

Task 4 – Conflict evaluation

Weighted matrices technique for evaluating alternative options for the above documented

conflicts: $totalScore(opt) = \sum_{crit} (Scores(opt, crit) \times Weight(crit))$

1. Consider the conflicts evaluation table for conflict between two statements (S1, S2)

Evaluation Criteria NFR	Significance Weighting	Alternative Option 1: Statement highlighting the key aspect of the system.	Alternative Option 2: Statement highlighting detailed objectives of the system.
Interactive	0.4	0.4	0.9
Useability	0.3	0.4	0.8
Convenience	0.3	0.6	0.5
Total	1.0	0.46	0.75

So, alternative option 2 of “highlighting detailed objectives of the system” seems to emerge according to above estimates.

2. Consider the conflicts evaluation table for conflict between two statements (S1, S5)

Evaluation Criteria NFR	Significance Weighting	Alternative Option 1: Content session locked for a time period when the branch employee is updating the status of tools.	Alternative Option 2: Customers can rent tools only after a branch employee’s notification.
Data availability	0.5	0.6	0.9
Fast Response	0.3	0.7	0.6
Cost effective	0.2	0.3	0.6
Total	1.0	0.57	0.75

So, alternative option 2 of “allowing customers to rent tools only after branch employee’s notification” seems to emerge according to above estimates.

3. Consider the conflicts evaluation table for conflict between two statements (S3, S4)

Evaluation Criteria NFR	Significance Weighting	Alternative Option 1: Context diagram highlighting high-level features and details of servers, and models on how to isolate business logic from user interface.	Alternative Option 2: Context diagram focusing more on highlighting the overall functionality of end users.
Distribution	0.4	0.8	0.5
Less Maintainability	0.3	0.7	0.8
Interoperability	0.3	0.9	0.5
Total	1.0	0.80	0.59

So, alternative option 1 of “highlighting high level features along with details of technology in the context diagram” seems to emerge according to above estimates.

4. Consider the conflicts evaluation table for conflict between two statements (S7,S8)

Evaluation Criteria NFR	Significance Weighting	Alternative Option 1: Separate login and sign up portal for customers, branch employees and system administrators with separate tables.	Alternative Option 2: Add “Type of user” option along with username and password in order to differentiate their roles and login pages.
Usability	0.4	0.7	0.9
Minimal inconvenience	0.3	0.6	0.8
Confidentiality	0.3	0.8	0.7
Total	1.0	0.70	0.81

So, alternative option 2 of adding “UserType” option along with username and password to differentiate login pages for each user seems to emerge according to above estimates.

Task 5 – Risk management

1. Risk Identification

a. Component Inspection

- i. Hardware Failure: The Hardware device from which end users: customers, system administrators and branch employees are accessing the web-application might get damaged, degraded or incompatible with the system, then its hardware failure risk.
- ii. Server Failure: In the current system, we are using MYSQL database server which is a single server and so there are high chances of its getting failed due to a single point of failure.
- iii. Network Failure: Consider a scenario where the system lost its connection due to power failure or unstable or poor network connection.

b. Risk Checklist

- i. Development Constraint: Cost and deadline : Overall development cost might increase because of exhaustion of resources and deadline may also get extended due to poor management of project.
- ii. Security : Confidentiality : Since there is no security technique implemented in the current system, it is open to be exploited by bad users such as hackers, injection attacks and data leaks.

c. Risk Tree

From the following risk tree, the possible risks identified are : hardware failure, server failure, network failure, and security failure.

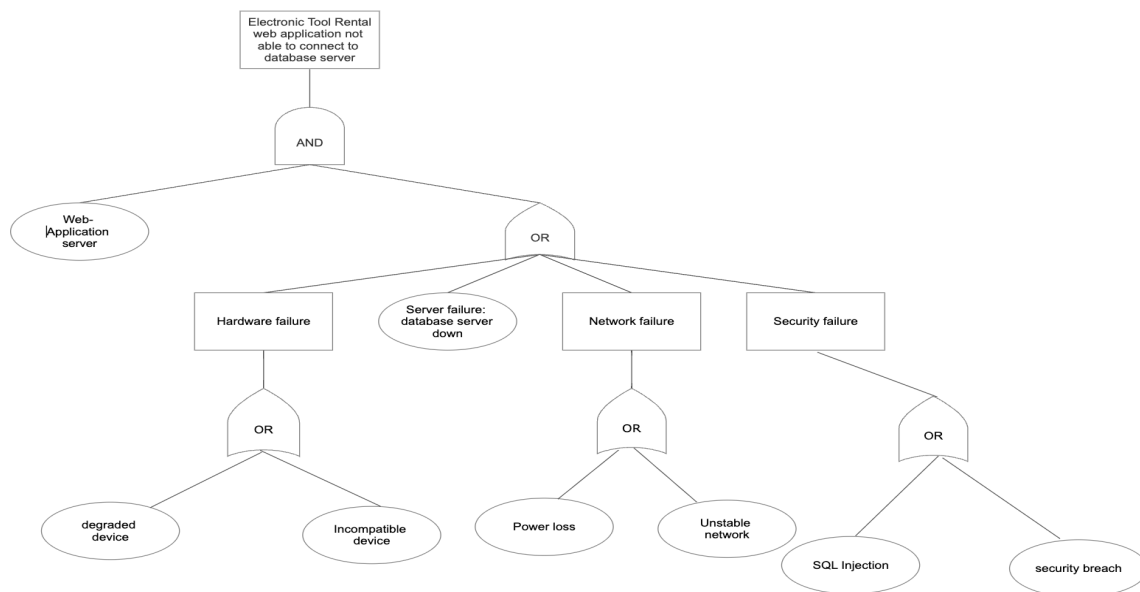


Figure 1: Risk tree

2. Quantitative assessment of the risks

Risk	Rationale	Likelihood/ risk probability	Severity (1-10)/ impact	Risk exposure = Risk probability * impact
Hardware failure	There is a possibility that the hardware device which end-users are using to interact with the system is damaged or incompatible with the system. (LIKELY)	40%	8	3.2
Server failure	System is more likely to face single point of failure due to MYSQL database server.(LIKELY)	60%	8	4.8
Network failure	There are high chances of network instability and hence leads to communication failure.(LIKELY)	40%	7	2.8
Development risk: cost and deadline	There are chances of exhaustion of all the resources and exceeding planned cost while development of web application.(POS	30%	6	1.8

	SIBLE)			
Security risk: confidentiality	There are chances of security breach on network if encryption techniques are not implemented on time.(LIKELY)	35%	9	3.15

3. Risk control

To evaluate better countermeasure for particular risk, Risk reduction leverage will be used:

$$\text{Risk reduction leverage (RRL)} = \frac{RE_{\text{before}} - RE_{\text{after}}}{\text{Cost of risk reduction}}$$

$$RE = \text{risk probability} \times \text{amount at stake}$$

1. Hardware Failure Risk

- a. Problem Statement : Consider a situation where end-users devices are incompatible with the system or they are using degraded devices on which system functionality might not work properly. The probability of hardware failure is 40%. Hardware risk is estimated to cost at 8.

Risk Exposure (before) = 3.2

- b. Countermeasures

- i. **Countermeasure 1:** Using an avoid risk tactic, we can introduce a new requirement as a prerequisite for end-users to upgrade to the latest devices that are compatible with the system along with the browser version.

Estimated cost: 3

Risk probability: 4%

Risk exposure (after) = impact * risk probability = 0.04 * 8 = 0.32

Risk Reduction Leverage(1) = $(RE_{\text{before}} - RE_{\text{after}}) / \text{Cost of Risk Reduction}$
= $(3.2 - 0.32) / 3 = 0.96$

Countermeasure 2: Using a reduced risk likelihood tactic, we can introduce a new requirement for customers that they should check the hardware maintenance and for system administrator and branch

employees, they should inform the technical support team whenever there is any hardware failure.

Estimated cost: 2

Risk probability: 5%

Risk exposure (after) = impact * risk probability = $0.05 * 8 = 0.40$

Risk Reduction Leverage(2) = $(RE_{\text{before}} - RE_{\text{after}})/\text{Cost of Risk Reduction}$
 $= (3.2 - 0.40)/2 = 1.4$

Since $RRL2 > RRL1$ which indicates a cost effective reduction measure. Also $RRL2$ is greater than one. Hence Countermeasure 2 looks more promising.

2. Server Failure Risk

- a. Problem Statement: Consider a situation where a system is more likely to face a single point of failure due to downtime of MYSQL database server. The probability of server failure risk is 60%. It is estimated to cost 8.

Risk Exposure (before) = 4.8

- b. Countermeasures

- i. **Countermeasure 1:** By the reduce consequence likelihood tactic, we can introduce a new requirement to reduce the consequence of single point failure by introducing cloud storage e.g. MongoDB uses concepts like sharding and replication in order to deal with single point of failure.

Estimated cost: 4

Risk probability: 2%

$RE_{\text{after}} : \text{impact} * \text{risk probability} = 0.02 * 8 = 0.16$

Risk Reduction Leverage(1) = $(RE_{\text{before}} - RE_{\text{after}})/\text{Cost of Risk Reduction}$
 $= (4.8 - 0.16)/4 = 1.16$

- ii. **Countermeasure 2:** By the reduce risk likelihood tactic, this risk can be countered by introducing a new requirement to use redundant server components and replication. Replication ensures that two servers remain synchronized and if one server is failed, the request can be routed to the second server.

Estimated cost: 8

Risk probability: 4%

$RE_{\text{after}} = \text{impact} * \text{risk probability} = 0.04 * 8 = 0.32$

Risk Reduction Leverage(2) = $(RE_{\text{before}} - RE_{\text{after}})/\text{Cost of Risk Reduction}$
 $= (4.8 - 0.32)/8 = 0.56$

Since $RRL1 > RRL2$ which indicates a cost effective reduction measure. Also $RRL2$ is greater than one. Hence Countermeasure 1 looks more promising.

3. Network Failure Risk

- a. Problem Statement: Consider a situation where there are high chances for the system to face the communication failure due to network issues which are caused by spikes in traffic that can overwhelm a network that is not scalable or due to low bandwidth. Also, power loss which is not an uncommon occurrence can also lead to network failure.

The probability of network failure risk is 40%. It is estimated to cost 7.

Risk Exposure (before) = 2.8

- b. Countermeasures

- i. **Countermeasure 1:** By the reduce consequence likelihood tactic, these consequences of risk can be reduced by introducing a new prerequisite requirement to encourage users to have high speed and stable internet connection. If not, they should buy a service with the best Internet service provider available.

Estimated cost: 6.5

Risk probability: 6%

$RE_{\text{after}} = \text{impact} * \text{risk probability} = 0.06 * 7 = 0.42$

Risk Reduction Leverage(1) = $(RE_{\text{before}} - RE_{\text{after}}) / \text{Cost of Risk Reduction}$
 $= (2.8 - 0.42) / 7 = 0.36$

- ii. **Countermeasure 2:** By the reduce risk likelihood tactics, a new requirement can be introduced to users that they should use laptops which are designed to be energy efficient and using features like Compressed memory to stay fast and save power. Or they should further optimize energy usage by putting laptops to sleep when not in use. So that in power failure too they can stay connected to the system.

Estimated cost: 2

Risk probability: 8%

$RE_{\text{after}} = \text{impact} * \text{risk probability} = 0.08 * 7 = 0.56$

Risk Reduction Leverage(2) = $(RE_{\text{before}} - RE_{\text{after}}) / \text{Cost of Risk Reduction}$
 $= (2.8 - 0.56) / 2 = 1.12$

Since $RRL2 > RRL1$ which indicates a cost effective reduction measure. Also RRL2 is greater than one. Hence Countermeasure 2 looks more promising.

4. Development risk: Cost and Deadline

- a. Problem Statement: Consider a situation where there are high chances for the development team to consume all the resources in order to implement advanced functionality and they exceed planned cost as well as scheduled deadline for each delivery.

The probability of development risk is 30%. It is estimated to cost 6.

Risk Exposure (before) = 1.8

b. Countermeasures

- i. **Countermeasure 1:** By the reduce risk likelihood tactic, the risk can be reduced by introducing a new requirement as, first designing the system for basic functionality and as it keeps growing, you can add more functionality later.

Estimated cost: 3

Risk probability: 2%

$$RE_{\text{after}} = \text{impact} * \text{risk probability} = (6 * 0.02) = 0.12$$

$$\text{Risk Reduction Leverage(1)} = (RE_{\text{before}} - RE_{\text{after}}) / \text{Cost of Risk Reduction} \\ = (1.8 - 0.12) / 3 = 0.56$$

- ii. **Countermeasure 2:** By the avoid risk tactic, risk can be avoided by introducing a new requirement to develop a system using well-known technologies by experienced people having advanced tech stack. Tech stack for server side development consists of web frameworks, databases, programming languages. For frontend it includes, framework, and PHP. It significantly impacts web application cost in the long run. Also, with experienced people in development, the project will be completed as scheduled without passing the deadline.

Estimated cost: 1.5

Risk probability: 1%

$$RE_{\text{after}} = \text{impact} * \text{risk probability} = (6 * 0.01) = 0.06$$

$$\text{Risk Reduction Leverage(2)} = (RE_{\text{before}} - RE_{\text{after}}) / \text{Cost of Risk Reduction} \\ = (1.8 - 0.06) / 1.5 = 1.16$$

Since $RRL2 > RRL1$ which indicates a cost effective reduction measure. Also $RRL2$ is greater than one. Hence Countermeasure 2 looks more promising.

5. Security Risk: Confidentiality

- a. Problem Statement: Consider a situation where a web application is accessible easily and left vulnerable to data breach and waiting to be exploited by bad actors. The probability of security failure risk is 35%. It is estimated to cost 9.

Risk Exposure (before) = 3.15

b. Countermeasures

- i. **Countermeasure 1:** By reduce consequence likelihood tactic, we can introduce new requirement to use web application firewall to protect web application from security attacks

Estimated cost: 4

Risk probability: 3%

$$RE_{\text{after}} = \text{impact} * \text{risk probability} = 0.03 * 9 = 0.27$$

$$\text{Risk Reduction Leverage(1)} = (\text{RE}_{\text{before}} - \text{RE}_{\text{after}}) / \text{Cost of Risk Reduction} = (3.15 - 0.27) / 4 = 0.72$$

- ii. **Countermeasure 2:** By the reduce risk likelihood tactic, this risk can be reduced by introducing a new requirement that website developers must be educated about potential security problems. In order to build a safe web application they must understand SQL injections, cross-site scripting, and cross site resource forgery.

Estimated cost: 3

Risk probability: 1%

$$\text{RE}_{\text{after}} = \text{impact} * \text{risk probability} = 0.01 * 9 = 0.09$$

$$\text{Risk Reduction Leverage(2)} = (\text{RE}_{\text{before}} - \text{RE}_{\text{after}}) / \text{Cost of Risk Reduction} = (3.15 - 0.09) / 3 = 1.02$$

Since $\text{RRL2} > \text{RRL1}$ which indicates a cost effective reduction measure. Also RRL2 is greater than one. Hence Countermeasure 2 looks more promising.

Appendix

Task	Description	Time taken (in hours)
Task 1	Identifying and finding inconsistencies in the vision document.	3
Task 2	Documenting conflicts.	1
Task 3	Conflict resolution	1.5
Task 4	Conflict evaluation	1.5
Task 5	Risk Management	5
	Total	12