

We have discussed the definition of risk and how to calculate the risk levels. If you haven't read our article on [Risk Definition](#), then I would suggest you read that first before you jump on to this one. In this article, we will talk about **Project Risk and Product Risk** with some practical examples, so you get a solid understanding of this topic.

*What is Project Risk?*

*What is Product Risk?*

*Who should identify Product/Project risk?*

## What is Project Risk?

Project risks are uncertain situations that can impact the project's ability to achieve its objectives. What are these Objectives? Every Software project has an objective. It could be building a new eCommerce website with a defined set of acceptance criteria. It includes functional and non-functional characteristics of the software. Any event that may risk these objectives classifies as Project Risk.

There is often confusion on whether a Test Manager should involve himself in Project Risks, or should he limit himself to [testing risks](#)?

Testing is a part of the project, like development or product management. Any risk that will impact the development could have an impact on testing as well. As such, the QA Manager must be aware of all the project risks that can have an impact on testing. Who Identifies these risks?

Before we answer this question, we need to see what all risks can occur in a project. It's crucial that you understand these risks and how they can affect testing.

### **Project Issues:**

**Delays in Delivery and Task completion** - Heard this story before? The testing team was supposed to get stories on Monday, and it's already Friday !! It's the most common project risk where there is a delay in completing the Development Task for a story. Therefore, there is a delay in the story delivery to the testing team.

**Cost Challenges** - Project funds and resources might get allocated to other high priority projects in the organization. There could also be cost-cutting across an organization, which could lead to reduced funds and resources for the project. It will impact testing resources, as well. Can you relate to the risk that six folks would now do a work that was supposed to be done by ten people? Of course - the timeline always remains the same!

**Inaccurate Estimates** - Estimation of Home Page development for a website was 20 days of development and 7 days of testing. When actual work started, the team figures out that they will need 35 days of development and 12 days of testing. Can you relate to this? When a project

begins, then the high-level estimation happens according to which allocation of resources and funds takes place. These estimates likely turn out to be inaccurate when actual work starts. It could lead to delays, quality issues, or cost overruns for both development and testing teams.

## **Organizational Issues:**

**Resource Skillset Issues** - Imagine you are on an automation project which requires 10 QA resources skilled with Selenium. You end up with 3 resources who know Selenium and rest 7 useful resources who got three days of Selenium training. Sounds familiar? It is a vital issue where the skill set of resources doesn't match project needs. The training is also not sufficient to bridge that gap. Quite evident that this will lead to quality and on-time delivery issues.

**Personnel issues** - This could be HR and people-oriented policies of the organization. It also includes any workplace discrimination that may affect people

**Resources Availability** - Often, business users, subject matter experts, or key developers/testers may not be available due to personal issues or conflicting business priorities. It has a cascading impact on all the teams.

## **Political Issues:**

**Team Skills** - What will happen if Developers and testers don't talk to each other? There will be back and forth on defects and will lead to everybody's time wastage. It often happens that Dev Managers and Test Managers don't get along well. It cascades down to the team as well, and such an environment hampers effective test execution.

**Lack Of Appreciation** - What if you do all the hard work, but only development efforts get appreciation in team meetings?

## **Technical Issues:**

**Poor Requirements** - Poor definition of the requirements leads to different interpretations by the clients and development/testing teams. It leads to additional defects and quality issues.

**Tech Feasibility** - The requirements may not be feasible for implementation. There could be technical constraints due to which some of the client requirements may not meet as expected.

**Environment Readiness** - If the Test Environment is not ready in time, then it would lead to delays in testing. Moreover, at times, the testing team might need to test in a dev environment, which could lead to data issues.

**Data Issues** - If there is a delay in Data conversion and Data migration, then it would affect the testing team's ability to test with real data. E.g., If we are moving our website from one platform to another, it would need data migration. If a delay happens in this activity, in the testing environment, then it would affect testing.

**Development Process** - Weakness in the development process could impact the quality of deliverables. It could be due to a lack of skill set of the architect. It's could also be due to scrum master not setting up the right processes.

**Defect Management** - Poor Defect Management by Scrum Master or Project Managers could also lead to accumulated defects. Sometimes, the team picks up lower priority defects, which are easy to fix. It helps them to show up numbers. However, the high priority defects pile up. It leads to a lot of regression issues, and defect reopens might also occur.

## **Supplier Issues:**

**Delivery Constraints** - A third party that is required to supply services or infrastructure is not able to deliver in time. It could lead to delays and quality issues. E.g. For an eCommerce website, a third party provides images. The site is all ready and tested, but cannot go live as images are not ready!

**Contractual Issues** - Contractual issues with Suppliers could also affect the deliverable. E.g., A supplier comes on contract to fix any defect in 7 days. However, the project team needs to fix P1 defects in 2 days. It is a classic example where the contract does not happen as per project needs. It leads to delays in delivering the software.

These were the broad set of risks that come under project risk. I hope you got a good understanding of these. Subsequently, we will discuss yet another risk called product risk.

## What is Product Risk?

Product risks result from problems with the delivered product. Product Risks associate with specific quality characteristics of the product. Therefore they are also known as **Quality Risks**. These characteristics are :

*Functionality as per client requirements*

*Reliability of software*

*Performance Efficiency*

*Usability*

*Security of software*

*Compatibility*

*Ease of Maintenance*

*Portability*

### Examples of Product Risks

Let's discuss some practical cases of product risk, so you get a better understanding of it.

*Software doesn't perform some functions as per specification. E.g., When you place an order on an eCommerce website, then order confirmation email triggers, but SMS functionality does not work.*

*Software does not function as per user expectations. E.g., A user intends to buy a product and adds it to his Cart. During checkout, the product goes out of stock and subtracts from the Cart. However, the user is not shown any message to tell him what went wrong.*

*Computation Errors - These are common issues that a developer has not written the correct computational logic in code. E.g., When a discount applies to a product, then the discount gets applied to shipping costs as well.*

*\*A loop structure is coded wrong - E.g., A loop that was to run nine times, runs ten times as the developer has used the condition  $\leq 10$  instead of  $< 10$*

*Response times are high for critical functions. E.g., You are placing an order on a website. The order is successful, and your money deducts. However, the order confirmation screen takes a min to load. This response time is too high for a customer.\**

*User Experience of the product doesn't meet customer expectations. E.g., When a user is searching for a product on a website, he wants to filter out results. E.g. Filter with Size = Medium , Brand = Nike , Color = Blue. However, he cannot select all these filters at one go. He Selects Size, and the page refreshes. He then selects Brand and page again gets refreshed with*

*updated results. It works as per functional requirements. However, it leads to poor user experience.*

As you can see, product risks are nothing but defects that can occur in production.

## Who should identify Project Risk and Product Risk?

We know what the product and project risks are, and it's type. But who should identify these risks, and who should plan for its mitigation? It has always been a contentious topic of how much the QA team should be involved in risk assessment. Should this be left for Scrum Masters and Project Managers?

From the testing perspective, there are two types of risks.

**Direct Risk** - *These are the risks that originate in testing. E.g., Test Lead not available in the Integration phase is a direct testing risk. The QA manager should raise it and work for its mitigation.*

**Indirect Risk** - *Delay in the Development Story completion can happen. It is a direct development risk. The Architect or Dev Manager should raise it and work for its mitigation. However, do you see a QA impact over here? The QA Manager can raise a new testing risk that the impact on testing will happen if stories don't deliver on time. He can also call out QA's impact on the risk that Dev Manager has raised. The QA Manager should then work on its mitigation as well.*

So as you see, QA Lead / Manager needs to know all the product/project risks. These may or may not originate in testing. However, they could still impact test deliverables. The QA Manager should analyze the impact of these risks on testing, and plan for its mitigation.