

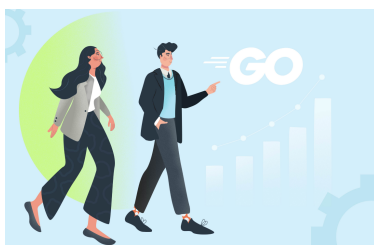
19 January 2022

Software Development

Risk management in software development projects



Among the various technology fields, **software development** might be one of the most inherently difficult to predict and plan. By nature, digital products are intangible and software engineering projects often involve many stakeholders. **Software development** processes tend to include multiple stages - including design, documentation, programming, and testing phases - and all of them require a high level of technological and management expertise. Because of the complex nature of digital projects, a wide variety of risks need to be considered and managed from the outset of any software project. Let's dive deeper into the different **types of risk in software development**.



Need an experienced **Golang** development team for your project?

[See our expertise](#)

Why is risk management important in software development projects?

Let's start with a brief explanation of what **software development risks** are. In short, these are factors that can affect the success of a digital project. They can occur both internally (when they are the result of situations inside the company) and externally (when they are subject to outside influence). There is also the question of personal risks relating to the efforts and quality of individual team members.

There are numerous reasons why identifying **software development risks** has become so important. A **software development risk management** plan helps the team evaluate the entire project, plan for success, maximize results, meet

CodiLime Glossary

some common networking terms clearly explained

Subscribe to our newsletter

Sign up to our Networks'nCloud Insider and get the latest news from us once a month

First name *

Last name *

Business email *

☐ I consent to receive marketing communications from CodiLime.*



Submit

Types of risk in software development

There are different **types of risks in software development**. We can categorize them along three dimensions:

Technical software development risks – these are risks introduced by flawed elements in the technology architecture or tools.

Operational software development risks – these occur in processes when software development goals aren't pursued with focus and clarity.

Business **software development risks** – these are those risks stemming from the company's wider business interests and operations.

Technical software development risks

Technical risks can be described as the difference between the actual and desired design of products and processes. Such risks are not immediately noticeable but carry serious negative consequences. Technical software development risks often lead to failure of functionality and performance. Unfortunately, they may be difficult to address. The most common technical risks in **software engineering** and development are:

Poor code quality – poor code quality is often the result of rushed work or a lack of knowledge and professionalism on the part of team members. Sometimes they are caused by constant changes in software requirements, overly complex multi-part projects, or an insufficient level of development and community support. You can minimize this type of risk by [testing code frequently](#), setting code standards with the team, or resolving bugs once they are found.

Wrong technology stack chosen for the project – sometimes developers are forced to place too much emphasis on popular technologies. However, do not forget that each team has its own core knowledge or experience in specific technologies or solutions. It is also worth mentioning that each technology/solution has unique features that will work better for specific types of projects (e.g. you would probably not choose to use C language to code web applications, despite its extensibility and speed because building an application in the C programming language tends to take much more time).

Working on existing source code – this usually occurs when your team takes over an ongoing project with existing source code. This may lead to serious technical issues. Your developers need time to investigate the code and assess its quality. To mitigate the risk, request all the existing documentation and discuss any issues with the previous development team. If the technology is new, the risk increases.

Integration issues – this may happen when you have to integrate the software with third-party systems or a CMS. The risk is lower if the technologies involved are popular and supported by large communities.

Operational software development risks

This type of software development risk includes project management, scheduling, and organizational issues that may occur at any stage of the

lead to the project's failure as they strongly influence project outcomes.
Among the operational risks in software development, you will find:

Overly ambitious or incorrect deadlines – you can mitigate this risk by creating a detailed plan that allows you to set realistic deadlines.

Bad timing – such scheduling problems include incorrect time calculations, lack of resources, unexpected and urgent expansion of the project scope, etc.

Low productivity caused by delays, employee burnout, and many other factors – in order to prevent the project from low productivity, you should create a well-paced plan and communicate it effectively.

Poor or inadequate project management – this includes poor communication and interaction within the team, lack of leadership, insufficient experience of product managers, etc.

Changing the project scope to the extent it becomes something completely different – i.e. scope creep; this may lead to missing project deadlines and extended project timeframes.

Communication with stakeholders – either the stakeholders aren't available to the development team (which causes information delays or late delivery) or they require too much communication (which may lead to unnecessary discussions).

Communication problems and mistakes - factors such as language barriers, cultural differences, or simply a lack of communication can make the whole process poorer and less effective.

Compromising on design caused by skipping crucial steps and rushing into working on development tasks.

Lack of developers with the necessary skills in the team, or an unstable workload.

Business software development risks

Business risks in software development are those issues that are not technology- or process-related:

Unmet stakeholder expectations – such risks can be a result of inaccurate estimates in relation to: costs, deadlines, or outcomes. Unmet expectations can be avoided if you clarify the probability of your estimations, including details of any influencing factors.

Undefined metrics – every single business opportunity should be quantifiable. Before the project is started, the 'worth it factor' for investing time and money should be calculated. Business goals should be measured when evaluating **software development** outcomes.

Inconsistent priorities – a lack of clarity on which elements of a software solution matter most can lead to unnecessary software releases. To mitigate this risk, create a sequence of work products based on business priorities.

Budget issues – it is crucial to monitor projects so that they don't go over budget.

Lack of engagement – the development team must engage with all relevant stakeholders, including third-party outsourcing partners. A lack of engagement in most cases results in a high risk of failure.

No partnership contract – in software engineering projects, partnership is a

all parties involved are 'on the same page' and the best way to ensure that is an agreed, written contract.

Unpredictable external risks – finally, there are also external **software development** risks worth considering. They can include changes in laws, new government regulations, economic shifts, changes in consumer behavior and priorities, and natural disasters. Although rare, they cannot be excluded from the list of potential risks because when they do occur, the project is highly likely to be impacted. It can be challenging to avoid them, but there are actions that can be taken to minimize such risks – for example, you can obtain insurance and stay up to date about legislative changes and current events.

[Which metrics should be considered in a dynamic startup environment?](#) Check out the recent blog post that ensures you keep on top of your project's progress.

Risk management in agile software development

Now, let's focus on how to identify and manage risks in software development. The main objective behind risk management actions is to know and understand what can possibly go wrong, the reason behind it, what the impact will be, and how to fix it. A **risk management strategy** may save you a lot of money as you reduce the cost of emergencies. It helps you work faster and build a better reputation as you ensure that you have everything under control.

For successful and systematic **risk management**, you should follow the steps below.

Identify and classify the risks

To avoid failure, you should identify risks with precision. Describe in detail the possible threats to your **software development** project. Be accurate and efficient. Keep in mind that any risk is just a potential problem that can be mitigated. Relevant and planned corrective actions are the best start in **risk management**. Of course, identifying and classifying risk should be an ongoing process.

Assess the likelihood of risks and their potential impact on your software development project

If you fail to properly assess a risk it can lead to a crisis. During this phase, you should prioritize the risks according to the level of impact on your project and the probability that certain risks will occur.

Create a detailed software development risk management plan

If you want to reduce the impact of the risks on project development, plan ahead. It's essential not only to set predetermined thresholds (the stage at

Constantly monitor and mitigate the risks

Constant monitoring allows you to identify and deal with possible risks immediately. This is why you should regularly review and update your risk management strategy – to ensure that your plans are objective, timely, and accurate. Risk mitigation is useful for reducing the effect of negative consequences when they cannot be avoided altogether.

Conclusion

Risk is always a potential problem. In **software development** a risk is any event that may compromise the success of your digital project. Relying on guesswork and crisis management is not an effective approach. In order to manage and mitigate risks properly, you should always have a **risk management plan**. The proper implementation of such a plan can help you prevent the impact of risks if and when they arise and, at the same time, improve the quality of your digital product. Even if every project comes with numerous software development risks, most of them can be prevented before they do any damage.



Tomasz Mika

Business Development Manager

[More from this author](#)

Read also

11 minutes

[How long \(for real\) does it take to make an application?](#)



Krzysztof Sajna

12 minutes

[The difference between Quality Assurance and Quality Control in sof...](#)



Multiple authors

Browse by category

Highlights

A starter pack containing the impactful networking and cloud technologies...

[See posts →](#)

Deep dive

Real-life challenges and research cases solved and presented by CodiLime's...

[See posts →](#)

Business insights

Current trends tracked and packaged in a series of articles to give you the...

[See posts →](#)

First name *

Last name *

Message

Tell us a bit about your project

Company *

Business email *

Phone number

Nepal (नेपाल)



+977



I consent to receive marketing communications from CodiLime.

Submit

lime



SERVICES

Software engineering

Network professional services

R&D

COMPANY

About

News

Careers

KNOWLEDGE BASE

Case studies

Blog

© CodiLime 2022 - All rights reserved.

[Privacy Policy](#) / [Terms of service](#) / [Information Security Policy](#)

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

[Privacy policy](#)

Ok