Risk management for technical projects: a risk register that people actually use

20 November 2026 · 6 min read · blogs.iictp.org/project-risk-register-template-and-examples

Every project carries risks; the question is whether you have written them down before they turn into problems. A risk register is a short, living list of what could go wrong, how likely and how bad it would be, and who is doing what about it. It only works if the team looks at it regularly.

TL;DR The short version

- A risk register is a short, living list of what could go wrong, how likely and bad it is, and who is acting on it.
- Write each risk as cause, event and effect, rank by likelihood times impact, and choose to avoid, reduce, transfer or accept it.
- Remember single points of knowledge and third-party changes, and review the top five risks every week or two.

Read online, listen to it, or change the text size: https://blogs.iictp.org/project-risk-register-template-and-examples

Risk vs issue

A risk has not happened yet: it might. An issue is happening now. When a risk occurs, close it in the register and open an issue. Keeping the two apart stops the register from turning into a bug list.

What each entry needs

- **Risk statement:** "Because of [cause], [event] may happen, which would lead to [effect]."
- **Likelihood and impact:** a simple 1–5 scale for each. Multiply them to rank.
- **Owner:** one named person, not a team.
- **Response:** what you will do about it, and by when.
- **Trigger:** the early sign that it is starting to happen.

Four ways to respond

- **Avoid:** change the plan so the risk cannot occur, for example dropping an untested dependency.
- **Reduce:** make it less likely or less harmful, for example a spike, a prototype or a feature flag.
- **Transfer:** move the impact elsewhere, for example a vendor support contract or insurance.
- **Accept:** decide consciously to live with it, and record a fallback if it happens.

Technical risks people forget

- A single person who understands a critical system.
- Third-party APIs, licences or pricing that can change.
- Data migration and rollback.
- Security and compliance review late in the project.
- Performance at production scale, not test scale. See design for the failure you will have.

Keep it alive

Review the top five risks in an existing meeting every week or two rather than holding a separate risk meeting. Remove risks that have passed. A register that is only updated for audits is decoration.

> Related credential: Certified Technical Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. IICTP. TPBOK, Engineering Governance & Agile Delivery. iictp.org/standards/tpbok.
2. IICTP. TPBOK, Infrastructure & Security. iictp.org/standards/tpbok.

Published by IICTP, an independent technical credential body, on 20 November 2026. The current version of this post is at https://blogs.iictp.org/project-risk-register-template-and-examples. More guides: blogs.iictp.org.