

As you work through this module, you will inevitably come across terms and pieces of information which are novel and worth sharing. You can place such information in the Module Wiki to benefit the whole group.

In the wiki, you can place a term/word/phrase, as well as its definition and/or any information you have sourced on the topic (clearly referenced).

To add a document/link click on 'Add a new entry'. You will then need to add a Concept title and Definition, add your attachment and click 'Save changes'.

Browse the glossary using this index

[Special](#) | [A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#) | [ALL](#)

Wiki Entry: Risks and Risk Mitigation

Risks and Risk Mitigation

Recent research indicates that many of the most significant risks in software and enterprise projects are organisational and human-centred rather than purely technical. Anton and Nucu (2020) highlight key risks such as poor or incomplete requirements definition, weak early risk identification, lack of governance and management support, and limited stakeholder engagement. These findings are reinforced by Biable et al. (2023), who argue that inadequate consideration of ethical and stakeholder-related factors during requirements engineering can significantly increase project risk.

These risks closely align with the early stages of the traditional Software Development Life Cycle (SDLC), particularly requirements analysis and system design. When risks are not addressed at these stages, they often propagate into later phases, leading to increased rework, delivery delays, and long-term technical debt (Vidoni, Codabux and Fard, 2022). This highlights the importance of integrating risk management activities throughout the SDLC rather than treating them as isolated tasks.

To capture and categorise these risks, the ISO 31000 risk management framework combined with a structured risk register, as discussed in the Unit 3 Lecturecast, is appropriate. This framework supports continuous risk identification, assessment, and mitigation, enabling risks to be categorised as technical, organisational, human, or ethical and prioritised based on likelihood and impact.

Added Risk: Incomplete or ambiguous software requirements during project initiation.
Suggested Mitigation: Conduct structured requirements validation sessions with stakeholders and apply iterative reviews at each SDLC phase to ensure requirements remain clear, agreed upon, and traceable.

References:

Anton, G. and Nucu, A. (2020) 'Enterprise Risk Management: A Literature Review and Agenda for Future Research', *Journal of Risk and Financial Management*, 13(11), pp. 281.

Biable, S. E., et al. (2023) 'Proposed Ethical Framework for Software Requirements Engineering', *IET Software*, 17(4), pp. 526–537.

Vidoni, M., Codabux, Z. and Fard, F. H. (2022) 'Infinite Technical Debt', *The Journal of Systems and Software*, 190, pp. 111336.

Keyword(s): Risks, Risk Mitigation

> [Comments \(0\)](#)



[Site Accessibility Statement](#)
[Privacy Policy](#)

