Site: **Your Learning Platform**
Module: **Software Engineering Project Management October 2025 (SEPM_PCOM7E October 2025)**
Glossary: **Module Wiki**

A

Add your Essex calendar to your Outlook account (work or private)

Did you know you can add your Essex agenda to your Outlook/Hotmail/Microsoft account? Read the super small howto.

Add your Essex Agenda to Outlook.pdf

Agile Scrum

Agile Scrum is an iterative and incremental framework within agile software development that emphasises adaptive planning, early and frequent delivery of value, and continuous improvement through structured feedback. It defines clear accountabilities (Product Owner, Scrum Master, Developers), events (Sprint, Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective), and artefacts (Epics, Product Backlog, Sprint Backlog, Increment) with associated commitments (Product Goal, Sprint Goal, Definition of Done) (Schwaber and Sutherland, 2020).

Scrum remains widely adopted within agile practices, with 42% of organisations using hybrid models that incorporate Scrum alongside DevOps and other frameworks, demonstrating its adaptability across distributed and hybrid environments (Digital.ai, 2024). Larger organisations show greater adoption of hybrid approaches (49%), while medium-sized companies display a similar preference (45%).

References

• Digital.ai (n.d.) 17th State of Agile Report. Available at: **https://digital.ai/resource-center/analyst-reports/state-of-agile-report** (Accessed: 23 October 2025).

• Schwaber, K. and Sutherland, J. (2020) The Scrum Guide. Available at: **https://scrumguides.org/scrum-guide.html** (Accessed: 23 October 2025).

R

Risks and risk mitigation

Wiki Entry: Risks and Risk Mitigation – Anton & Afloarei Nucu (2020)

1. Main risks identified in Anton & Afloarei Nucu (2020) and how they fit into the SDLC

Anton and Afloarei Nucu (2020) conduct a systematic literature review of Enterprise Risk Management (ERM) and therefore do not present their own taxonomy of risks. Instead, they synthesise findings from empirical studies that identify enterprise-level risk motivators for ERM adoption. A key referenced study, Khan et al. (2016, cited in Anton & Afloarei Nucu, 2020), identifies the following risk drivers:

- Probability of financial distress
- Low earnings performance
- Growth-opportunity vulnerabilities
- Lack of board independence / weak governance

These risk types can be mapped to the traditional Software Development Life Cycle (SDLC) as follows:

Planning & Analysis: Financial distress and weak governance can lead to unstable budgets, unrealistic scoping, and poor requirements decisions.

Design & Implementation: Growth-opportunity risks and weak board oversight reduce organisational responsiveness, limiting architectural quality and increasing the likelihood of technical debt.

Testing & Maintenance: Low earnings performance often correlates with underinvestment in long-term quality, scalability and maintenance, increasing operational and security risks after deployment.

2. Framework for capturing and categorising the risks

From the Unit 3 lecturecast, the Risk Management Process (RMP) framework encompasses four core phases: identification, analysis, evaluation and treatment. This framework is particularly suitable for capturing Anton & Nucu's identified enterprise risks because it:

- Identification phase: Systematically surfaces financial, governance and strategic growth risks across all organisational systems (strategy, structure, technology, intellectual resources)
- Analysis phase: Examines how each risk (e.g. weak board independence) propagates through SDLC phases and impacts budget stability, architectural decisions and long-term maintenance costs
- Evaluation phase: Prioritises risks by likelihood and impact, aligning with organisational risk appetite
- Treatment phase: Develops mitigation strategies tailored to each SDLC phase

This approach aligns with the Hoffmann et al. (2016) model discussed in the lecturecast, which treats the organisation as an integrated system of strategy, structure, technology, intellectual resources and management systems.

3. Risk + suggested mitigation (Wiki contribution)

Risk: Fragmented ERM governance resulting in unclear risk ownership across SDLC phases.

Mitigation: Establish formal risk ownership roles aligned to the SDLC phases (Planning Risk Owner, Development Risk Owner, Deployment Risk Owner). Integrate risk identification, analysis and evaluation checkpoints at each SDLC phase gate to prevent decision-making gaps and ensure continuity of enterprise-level oversight, following the RMP treatment and monitoring protocols.

References

Anton, S.G. and Afloarei Nucu, A.E. (2020) 'Enterprise Risk Management: A Literature Review and Agenda for Future Research', Journal of Risk and Financial Management, 13(11), p. 281. Available at: https://doi.org/ 281

Hoffmann, R., Kiedrowicz, M. and Stanik, J. (2016) 'Risk management system as the basic paradigm of the information security management system in an organization'. Available at: https://doi.org/10.1051/MATECCONF/20167604010

W

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.