Unit 4: Formative activities

Wiki Entry: Risks and risk mitigation. Read the articles by Verner et al. (2014) and Anton & Nucu (2020) and then answer the following questions:

1. **What are the main risks that the authors identify? How do these fit into the traditional SDLC model?**

The traditional SDLC model, a sequential approach where each phase must be completed before proceeding to the next, is a cornerstone in software development. This model, encompassing popular methods such as Waterfall, Iterative, Spiral, and V-shaped, is critical to the project's success. It thrives in scenarios where technology stability is expected, project duration is short, delivery speed isn't a priority, and the requirements are well-defined (GeeksforGeeks, 2018).

Anton & Nucu's (2020) report underscores the potential of Enterprise Risk Management (ERM) as a leading paradigm in corporate governance. ERM empowers organisations to identify, evaluate, and manage risks at the enterprise level. It's not just a risk management tool but a potential competitive advantage. It supports firms' growth by mitigating the risk of financial distress and low earnings performance and capitalising on growth opportunities and board independence.

On the other hand, Verner et al. (2014) note that global software development (GSD) projects are often complex and risky due to the increased complexity and complexity of the process. Projects usually fail due to communication and knowledge exchange issues between onshore and offshore team members. Risk factors for GSD include communication issues, poor change controls, lack of business know-how, and failure to consider all costs. However, clients have a crucial role to play in dealing with these risks.

To understand these risks and how they fit into the traditional Software Development Life Cycle (SDLC) model, here is an explanation:

1. Risk of financial distress and associated costs: this risk aligns with the requirements analysis phase in the SDLC. Financial constraints not adequately considered during requirements gathering can lead to project delays or budget overruns (Appsierra, 2024).

2. Risk of Low Earnings Performance: this risk in the SDLC corresponds to the design and development phases. If these processes are inefficient or ineffective, they may impact earnings and overall project success (Sanie, 2022).

3. Risk of Missed Growth Opportunities: this risk in the SDLC relates to the planning and feasibility stages. Failure to identify and assess growth opportunities during project planning can hinder long-term success (Hijazi et al. 2014).

4. Risk of Board Independence: this risk in the SDLC aligns with the governance and control phase. Effective governance ensures unbiased risk evaluation and decision-making through line management, compliance management, a management risk committee, and a board for control and monitoring (Guzzi, 2023).

5. Risk of communication issues: this SDLC risk aligns with the requirements analysis phase. Clear communication during requirements gathering ensures an accurate understanding of user needs (McGuire, 2024).

6. Risk of poor change controls: this SDLC risk aligns with the change management phase. Proper change control procedures prevent unauthorised or untested modifications (Guzzi, 2023).

7. Risk of lack of business know-how: this SDLC risk aligns with the planning and feasibility phase. A thorough understanding of the business context ensures alignment between technical solutions and organisational goals (Velimirovic, 2019).

8. Risk of failure to consider all costs: This risk in the SDLC is relevant during the project planning phase. Accurate cost estimation and budget planning are critical (Swersky, 2018).

2. Which of the frameworks discussed in the Unit 3 lecture cast would you use to capture and categorise the risks?

I recommend using the Risk Management Framework (RMF) as it is a comprehensive and adaptable approach that focuses on managing risks and integrating security,

privacy, and cyber supply chain management activities into the system development life cycle. This framework considers the effectiveness, efficiency, and legal constraints that may arise due to laws, directives, policies, standards, or regulations. The RMF can be applied to various systems, including new and legacy ones, and any technology within any organisation. The process involves system categorisation, control selection, control implementation, and continuous assessment and monitoring of control implementation and risks (NIST, 2016).

3. Add a risk and a suggested mitigation to the module wiki.

Operational risk is a type of risk that businesses face internally. It arises from internal procedures, people, and systems. Companies must manage it proactively by identifying potential risks, performing cost/benefit analyses, avoiding unnecessary risks, and delegating strategic planning to upper management. Mitigating operational risks is crucial to preventing financial losses and ensuring business continuity and success (Segal, 2023).

According to AlertMedia (2023), operational risk is vital in maintaining a company's resilience. It can arise from various sources, such as internal fraud, external fraud, process management failures, employment and safety practices, technological failure, damage to physical assets, and business practices. Although we may not have control over some risks, such as natural disasters, we are still responsible for managing these risks within our organisation.

Risk mitigation is a crucial aspect of business operations, as emphasised by Kirvan (2021). It minimises the impact of risks on people, processes, technology, and facilities. Companies must identify and develop a plan to address these risks, analysing their

likelihood and potential effects on business processes, employees, and financial results.

Implementing assertive strategies, such as actively involving stakeholders, documenting processes comprehensively, conducting thorough risk assessments, and monitoring key risk indicators, is essential to mitigating operational risks. Additionally, it is crucial to develop robust business continuity and disaster recovery plans, invest in employee training, automate repetitive tasks, establish robust internal controls, and regularly review access rights and permissions. Conducting scenario-based testing and simulations and evaluating vendor performance and compliance are also critical (Storkey, 2011).

4. **Choose an entry made by one of your colleagues in the wiki and comment on how you might mitigate it.**

References:

- GeeksforGeeks. (2018). *Software Engineering | RAD Model vs Traditional SDLC*. [online] Available from: https://www.geeksforgeeks.org/software-engineering-rad-model-vs-traditional-sdlc/ [Accessed 7 Apr. 2024].
- Anton, S.G. & Nucu, A.E.A. (2020). Enterprise Risk Management: A Literature Review and Agenda for Future Research. *Journal of Risk and Financial Management*, [online] 13(11), p.281. doi:https://doi.org/10.3390/jrfm13110281.
- Verner, J.M., Brereton, O.P., Kitchenham, B.A., Turner, M., & Niazi, M. (2014). *Shibboleth Authentication Request*. [online] Available from: https://www-sciencedirect-com.uniessexlib.idm.oclc.org/science/article/pii/S0950584913001341?via%3Dihub [Accessed 7 Apr. 2024].
- Appsierra (2024). *Requirement Analysis: Key Phase of Software Development Life Cycle*. [online] Available at: https://www.appsierra.com/blog/sdlc-requirement-analysis [Accessed 7 Apr. 2024].
- Sanie, M. (2022). *5 Phases of Software Development Life Cycle and Risk Assessment*. [online] Cprime. Available from: https://www.cprime.com/resources/blog/5-phases-of-software-development-life-cycle-and-risk-assessment/ [Accessed 7 Apr. 2024].
- Hijazi, H., Alqrainy, S., Muaidi, H. & Khdour, T. (2014). *RISK FACTORS IN SOFTWARE DEVELOPMENT PHASES.* [online] 10(3), pp.1857–7881. Available at: https://eujournal.org/index.php/esj/article/view/2624/2485.
- Guzzi, B. (2023). *SDLC vs Change Management Controls: What Auditors Should Know*. [online] Available from: https://www.auditboard.com/blog/sdlc-vs-change-management-controls/ [Accessed 7 Apr. 2024].
- McGuire, J. (2024). *Requirements Analysis for Software Development (Guide)*. [online] Pulsion Technology. Available from: https://www.pulsion.co.uk/blog/requirements-analysis-for-software-development/ [Accessed 7 Apr. 2024].
- Velimirovic, A. (2019). *What is SDLC? How the Software Development Life Cycle Works*. [online] PhoenixNAP Global IT Services. Available from: https://phoenixnap.com/blog/software-development-life-cycle.
- Swersky, D. (2018). *The SDLC: 7 phases, popular models, benefits & more [2019]*. [online] Raygun Blog. Available from: https://raygun.com/blog/software-development-life-cycle/ [Accessed 7 Apr. 2024].
- NIST (2016). *About the RMF - NIST Risk Management Framework | CSRC | CSRC*. [online] CSRC | NIST. Available from: https://csrc.nist.gov/projects/risk-management/about-rmf [Accessed 7 Apr. 2024].
- Segal, T. (2023). *Operational Risk Overview, Importance, and Examples*. [online] Investopedia. Available at: https://www.investopedia.com/terms/o/operational_risk.asp.
- AlertMedia (2023). *7 Operational Risk Examples & Mitigation Strategies*. [online] AlertMedia. Available from:

https://www.alertmedia.com/blog/operational-risk-examples/ [Accessed 7 Apr. 2024].

- Kirvan, P. (2021). *7 risk mitigation strategies to protect business operations*. [online] SearchCIO. Available from: https://www.techtarget.com/searchcio/feature/7-risk-mitigation-strategies-to-protect-business-operations [Accessed 7 Apr. 2024].
- Storkey, I. (2011). *Operational Risk Management and Business Continuity Planning for Modern State Treasuries; by Ian Storkey; IMF Technical Notes and Manuals TNM/11/05; November 09, 2011*. [online] Available from: https://www.imf.org/external/pubs/ft/tnm/2011/tnm1105.pdf.