What are the top five overall causes of Risk?

If not properly managed, the software development process bears risks for any organisation. It is vital to have a good understanding of the individual phases of the SDLC to minimise exposure to risks that hinder the development process. To manage the SDLC and its risks adequately, common problems and their causes for each phase have to be identified (Hijazi et al, 2014).

The first stage of the SDLC is the requirements analysis and definition phase. In this stage, project requirements are collected and analysed to define goals, timeline, cost and the overall project scope. In this phase, unclear requirements and communications are a significant cause of risk. Roy et al (2015) states that the instability of requirements can cause risks in the definition phase. By having continuously changing requirements, a valid project timeline and cost estimation can not be created. Furthermore, both analysts and developers will not understand the complete picture if the requirements are unclear and incomplete. According to Shahzad et al (2009), the instability and inaccuracy of requirements stem from the fact that users cannot describe more than 60% of the requirements at the beginning of the project.

In the design phase, the second phase of the SDLC, all requirements collected from the previous stage are addressed to establish an overall system architecture. Developers are involved in this process to verify that the requirements are comprehensive and define a suitable programming language and software components (Hijazi et al, 2014). Huang & Han (2008) highlight an important cause of risk for this phase: the development of a too complicated system. If the software system is too large and complex, developers may have trouble decomposing the system into its main components. This will ultimately lead to an overall disconnect between the developers and the system, and may cause a high risk of project failure.

The implementation and unit testing phase comprises of the actual implementation of the system by executing the previously defined design. Many risks here are caused by insufficient code readability and reusability. If the code that the developers implement is not well-documented, code reviews and further extensions to the system will be hard to execute and can cause stability issues in the long-term (Hijazi et al, 2014).

The fourth phase is the integration and system testing phase. This includes integrating and testing the unit tested modules to produce the complete software system. In this phase, an experienced testing team and an adequate testing environment are necessary. Else, problems within the system may not be identified and fixed accordingly and may be released to the users (Roy et al, 2015).

Finally, the operation and maintenance phase is the last and usually longest stage of the SDLC. In this phase, the software is rolled out to the customer and continuously maintained, should any issues arise. Team-turnover is a critical cause for risk in the overall SDLC; however, it is especially so in the longer-term maintenance phase. Experienced team members may, at any point, decide to pursue other opportunities that benefit them. This factor threatens the project's overall success and maintainability, especially if the code is not well-documented and not enough knowledge was transferred or shared with the rest of the team prior to them leaving (Shahzad, A. Al-Mudimigh & Ullah, 2010).

Reference List

Hijazi, H., Alqrainy, S., Muaidi, H. & Khdour, T. (2014) Identifying Causality Relation between Software Projects Risk Factors. *International Journal of Software Engineering and Its Applications* 8(2): 51-58.

Huang, S., & Han, W. (2008) Exploring the relationship between software project duration and risk exposure a cluster analysis. *Information and Management* 45(3): 175-182.

Roy, B., Dasgupta, R. and Chaki, N. (2015) A Study on Software Risk Management Strategies and Mapping with SDLC. *Advances in Intelligent Systems and Computing* 1(1): 121-138. DOI: https://doi.org/10.1007/978-81-322-2653-6_9 [Accessed 18 February 2021].

Shahzad, B., Ullah, I., & Khan, N. (2009). Software risk identification and mitigation in incremental model. *Proceedings of the International Conference on Information and Multimedia Technology*: 366–370.
DOI: https://doi.org/10.1109/ICIMT.2009.104 [Accessed 18 February 2021].

Shahzad, B., Al-Mudimigh, A., & Ullah, Z. (2010). Risk identification and preemptive scheduling in software development life cycle. *Global Journal of Computer Science and Technology* 10(2): 55–63.