Top of Form

Bottom of Form

**Unit 3: An Introduction to Risk and the System Development Life Cycle**

**Objectives:**

Review the some of the various methodologies used to implement the SDLC – focusing on the difference between Agile and Waterfall methods.

Discuss where risk affects each phase of the SDLC – and how some methodologies mitigate against it.

Evaluate modern approaches – with a special focus on DevOps.

**Outcomes:**

Describe what the SDLC is – and the different implementation methodologies available.

Explain how risks affect the SDLC.

Describe how to mitigate risks in a selected SDLC methodology.

**Reflection**

The Software Development Life Cycle, often known as SDLC, is a method for producing the highest-quality software at the lowest possible cost and in the shortest possible period of time. The Software Development Life Cycle (SDLC) provides a well-organized sequence of phases that enables an organization to swiftly produce high-quality, completely tested, and production-ready software. SDLC's advantages include a decrease in the cost of building software as well as an improvement in the product's quality and production speed. SDLC is able to achieve these apparently conflicting goals by sticking to a strategy that removes the typical obstacles encountered in software development projects. This plan will begin with a review of the existing systems to identify improvement opportunities (Altvater, et. al., 2020; Synopsys.com. et. al., 2019).

There are six phases of SDLC which are Requirement analysis, Planning, Software design such as architectural design, Software development, Testing, and Deployment. The waterfall model, the spiral model, DevOps, Lean, and the Agile model are all popular examples of SDLC models (Half, R. 2019).

Now here we are writing some points, because of these points risk effects the SDLC. We are here mentioning all the risk which faced in SDLC different phases. Failure of a project is often the result of unrealistic expectations about its money, timeline, resources, and scope. Incomplete prerequisites, ignoring nonfunctional requirements, conflicting user requirements, gold plating, insufficiently detailed description of the actual working environment, due to a misunderstanding of the industry specific language, making use of everyday language to communicate needs, inconsistent requirements data and RD, as well as RD that cannot be modified, are the risks that may lead to the failure of a project (GeeksforGeeks et. al., 2018).

We can mitigate the risk during SDLC model by using these steps which varies from SDLC model :CAST et. al., 2017).

* **Accept:** Acknowledge and accept the reality that a risk is impacting the project. Make a decisive decision to continue with the project regardless of the risk and without adjustments. In this situation, the approval of the project management team is required.
* **Avoid:** Adjust the project's scope, timetable, or constraints as required to mitigate the effect of the risk.
* **Control:** Take measures to mitigate the intensity of the impact or delay the development of the threat.
* **Transfer:** Your firm should conduct a transfer in order to delegate responsibility, obligation, or authority to other parties who are ready to assume the risk.
* **Continue Monitoring:** Monitoring the project environment for factors that might increase the risk's impact is a strategy often applied to low-impact risks.

**References**

Altvater, A. (2020). *What is SDLC? Understand the Software Development Life Cycle*. [online] Stackify. Available at: https://stackify.com/what-is-sdlc [Accessed 26 Jun. 2022].

Synopsys.com. (2019). *What Is the Software Development Life Cycle (SDLC) and How Does It Work? | Synopsys*. [online] Available at: https://www.synopsys.com/glossary/what-is-sdlc.html [Accessed 26 Jun. 2022].

Half, R. (2019). *6 Basic SDLC Methodologies: Which One is Best?* [online] Roberthalf.com. Available at: https://www.roberthalf.com/blog/salaries-and-skills/6-basic-sdlc-methodologies-which-one-is-best[Accessed 26 Jun. 2022].

Geeks for Geeks. (2018). *Integrating Risk Management in SDLC | Set 1*. [online] Available at: https://www.geeksforgeeks.org/integrating-risk-management-in-sdlc-set-1/#:~:text=Unrealistic%20budget%2C%20time%2C%20inadequate%20resources [Accessed 26 Jun. 2022].

CAST (2017). *Accelerate Application Modernization and Cloud Migration with Software Intelligence*. [online] Castsoftware.com. Available at: https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects. [Accessed 26 Jun. 2022].