Information Risk Management August 2021

« Collaborative Learning Discussion 2

Doug Leece

Initial post - Overall causes of risks to SDLC

333 days ago

2 replies

Last 318 days ago

While both papers identify risks can occur at any stage of the software development life cycle (SDLC), poorly developed requirements essentially sow the seeds for future project failures. (Hijazi et al,2014; Roy et al, 2015). Organizations experience primary losses when software project failure leads to cost, quality, and delivery speed impacts (Atkinson, 1999). Software projects are initiated to solve business problems or enhance competitive capability, consequently failures result in secondary losses that reside outside of the software development life cycle altogether.

The discussion topic “the top five overall causes of risks” implies there are underlying reasons for the risks Roy et al identified (2015). Many risks are associated with requirements and planning, common project management responsibilities and project management effectiveness within information technology continues to underperform compared to other industries (Varajao et al, 2014). Project manager involvement in initial SDLC phases such as requirements gathering and resource assignments, which when poorly done almost assures issues with later phases (Hijazi, 2014), logically implies project management is the biggest SDLC threat.

Such logic omits one key consideration, project managers must rely on the technical resources they have available from everything to estimating task timelines to design feasibility since they seldom have deep technical skills. Skilled technologist shortages will continue for several years to come (Allas et al, 2019), painting a bleak outlook for organizations pursuing software development since almost all identified technical risks are triggered by skill deficiency in one form or another.

References

Allas, T., Dimson, J. Foote, E. & Jeffery, B. (2019) The future of work: Rethinking skills to tackle the UK’s looming talent shortage. Available from: **https://www.mckinsey.com/featured-insights/future-of-work/the-future-of-work-rethinking-skills-to-tackle-the-uks-looming-talent-shortage** [Accessed 1 September 2021]

Atkinson, R. (1999), Project management: Cost, time and quality, two best guesses and a phenomenon, it is time to accept other success criteria. *International Journal of Project Management* 17(6): 77-82

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.

Roy, B., Dasgupta, R. & 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**.

Varajao, J., Dominguez, C., Ribeiro, P. & de Paiva, A. (2014) Failures in Software Project Management – are we alone? A comparison with construction industry. *The Journal of Modern Project Management* 2(1): 22-27

Reply

Maximum rating: 👍 (1)

2 replies

1

Post by **Doug Millward**
Feedback

319 days ago

Hi Doug
As always you raise some interesting points. Key questions for me are: who is best positioned to do the requirements gathering and analysis? In many organisations it would indeed be the PM or perhaps a business analyst/ architect - however, in some methodologies the task falls to the developers or technical architects - often via mechanisms such as 'planning poker' or similar - however, there are sources that question the validity of this approach and its effect on time and cost estimates. More recently some agile approaches have started using a language called gherkin which has the benefit of producing executable unit-tests from requirements. However, I absolutely agree that the requirements phase is key - for both SYSTEM and SOFTWARE development. :)

Reply

2

Post by **David Luvaha**
Peer Response

318 days ago

In support of paragraph two above, according to Roy et al (2015) overall causes of risks during SDLC are not limited to:

1. Risks connected to software programming like:
    (a) underestimating resources
    (b) unpredictability of users and customers
    (c) Vague requirements
    (d) Wrong designs
    (e) Poor or wrong work environment
    (f) Risks associated with developing systems
    (g) Insufficient management process
2. Technical risks linked to technology like:
    (a) In availability of premeditated framework
    (b) Slow acceptance of new technology
    (c) Supply Chain difficulties
    (d) Improper handling of change
    (e) Over reliance on technology to solve problems
3. Internal and External causes
4. Recognized and unknown causes
5. Risks of contradictory requirements

References

1. Roy, B., Dasgupta, R. & 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**.
Available from: https://www.researchgate.net/publication/283713387_A_Study_on_Software_Risk_Management_Strategies_and_Mapping_with_SDLC [Accessed 2 September 2021].

Reply

Add your reply

Your subject

Type your post

Choose Files No file chosen

Submit

Use advanced editor and additional options

Older discussion

Initial Post

Newer discussion

Initial Post