

Unit 1 Collaborative Discussion 1

Peer Responses

Contents

Peer Responses 1

 Response to Shan Swanlow 1

 Response to Sharon 2

 Response to Grace Clarke’s response to Sharon 2

 Response To Gennaro Coppola 3

 Response to Andrey Smirnov 3

 Response from Gennaro Coppola 4

 Response from Taylor Edgell 4

Response to Shan Swanlow

Hi Shan,

I agree with the statement that poor top-level management contributes to information loss and introduces ambiguity. A lack of proper planning often results in a weak task backlog. In order to prevent weak backlogs and ensure the best level of prioritisation, what would you recommend as an approach to better prioritise tasks?

I ask because often engineering teams prioritise new features over existing issues that require maintenance (Besker et al., 2019) which can lead to high maintenance costs as seen in the case with the U.S Air Force's ECSS system that had poor cooperation between developers and end-users.

References

Besker, T., Martini, A. & Bosch, J., (2019). Technical debt triage in backlog management. In 2019 IEEE/ACM International Conference on Technical Debt (TechDebt):13-22).

Response to Sharon

Hi Man Sze Wong,

I enjoyed reading about your two case studies which highlighted quite well the impact of insufficient test resources on the success of projects, with Toyota even suffering a drop in their listed stock price. Considering the role of project managers and testing, the book by Baumgartner et al. (2021) makes a few interesting statements. For example. "Many traditional roles, such as project manager and test manager, are vanishing". And "nowadays bugs are often discussed directly with the developers...and, if possible, fixed immediately. Bugs that cannot be fixed immediately are entered directly into the backlog." The last statement is interesting because it implies that agile environments should suffer less from a lack of testing resources.

What do you think may have contributed to the lack of testing resources for Toyota and Los Angeles Airport?

References

Baumgartner, M., Klonk, M., Mastnak, C., Pichler, H., Seidl, R. & Tanczos, S., (2021). Role of Testers in Agile Projects. In Agile Testing: 79-111. Springer, Switzerland.

Response to Grace Clarke's response to Sharon

Hi Grace,

I thought your point concerning "post-mortems" (Dalcher and Brodie, 2007) is interesting because it raises the question of who you consider responsible for project post-mortems? We could argue that it is the project manager or even the development teams. However, few development teams can likely exercise much control in improving project outcomes. Angara et al. (2020) consider that "project manager should also consider their previous judgement track record" when planning each sprint. I found their idea of generating project sentiment to reduce projects risks--resulting from poor communication, deadlines or customer expectations--to be a compelling concept to support project post-mortem.

References

Angara, J., Prasad, S. and Sridevi, G., 2020. DevOps Project Management Tools for Sprint Planning, Estimation and Execution Maturity. *Cybernetics and Information Technologies*, 20(2), pp.79-92.
Dalcher, D. and Brodie, L. 2007, *Successful IT projects*. Thomson Learning. Middlesex University Press

Response To Gennaro Coppola

Hi Gennaro,

I agree with your opinions listed, especially that incorrect project requirements can lead to project failure. Hovorushchenko and Pomorova (2018) also state that "Software projects often fail because of...inadequate formulation of the requirements;...incorrect understanding or insufficient analysis of the specification and project..." It seems then that customers often face difficulties formulating their requirements during the early stages of a project, which leads to incorrect project requirements.

To help steer consistent project requirements, TOGAF has the concept of requirements management which concerns itself with managing the changes to enterprise requirements, identifying, storing, and delivering them to each relevant phase of the TOGAF Architecture Development Module (TOGAF, 2011). Considering the TAURUS project's unclear requirements, have you come across any other framework that helps project managers manage requirements during a project's lifecycle?

References

Hovorushchenko, T. & Pomorova, O. (2018). Information Technology of Evaluating the Sufficiency of Information on Quality in the Software Requirements Specifications. In ICTERI Workshops:555-570.

TOGAF (2011). ADM Architecture Requirements Management. Available from <https://pubs.opengroup.org/architecture/togaf91-doc/arch/chap17.html> [Accessed 23 Mar. 2022]

Response to Andrey Smirnov

Hi Andrey,

I enjoyed reading your post that contained several examples of what successful project management should incorporate. Specifically, the reference to Schmitz et al. (2014) was well-placed to show the real-world need for project managers to include senior management buy-in because they are considered critical decision-makers in terms of operational activities, policies, budget and customer/product priorities. And also that functional requirements must align with IT investments. In addition, Alqaisi (2018) presents a concept of a circle of support that helps project managers to successfully interact with various stakeholders because "most of the time project managers do not have formal authority and working with the executive sponsors can make things happen".

References

Alqaisi, I.F. (2018). The effects of stakeholder's engagement and communication management on projects success. In MATEC Web of Conferences (Vol. 162, p. 02037). EDP Sciences.

Schmitz, A., Tada, B. & Hess, R. (2014) IT System Failures: The FBI's Virtual Case File Case Study. Available from: https://www.academia.edu/12133977/IT_System_Failures_The_FBI_s_Virtual_Case_File [Accessed 23 Mar. 2022].

Response from Gennaro Coppola

Hello Michael,

Thank you for your comment; I agree with your observations; about any framework to manage requirements, I can suggest IBM Doors: Engineering Requirements Management DOORS Next – Overview | IBM.

IBM guidelines also define the attributes that a requirement should have as: Specific, Testable, Clear and concise, Accurate, Understandable, Feasible and realistic, Necessary.

I believe this is a solid starting point to evaluate and manage requirements during a project.

Kind regards,
Gennaro

Response from Taylor Edgell

Hi Michael,

Thank you for your post, it was very informative in relation to the information covered in the original paper we were set to read.

I agree completely with your three identified reasons for software project failures of poor software quality, lack of communication and low test scope. Your comments of poor software quality were of great interest to myself. You make a good point that a low quality foundation hinders the creation of new features.

One question in relation to this point would be do you think this poor software quality could be linked to the poor acquisition of project requirements at the initial stages? If the full project scope is not fully understood by the developers this could potentially mean that the core software is not fit for purpose. It has been noted that "It is estimated that 85 percent of the defects in developed software originate in the requirements" (Young, 2002). I would be interested to hear your thoughts as to if you think problems in the process could be developed in as early a stage as this.

References:

Young, R.R., 2002. Recommended requirements gathering practices. CrossTalk, 15(4), pp.9-12.