Unit 2 Reflection

Requirements

This unit was continued the focus on reasons for project failures and suitable mitigations. Students also continued with an introduction into requirements gathering. I was pleasantly surprised to come across the Gherkin language introduced in this module, as I previously encountered this type of requirements gathering in former employment written by business analysts. Though I am familiar with structuring requirements in Gherkin syntax, I enjoyed learning about Gherkin, understanding it better from an academic perspective. A takeaway point is that Gherkin is versatile, usable by non-technical people and is often used to derive automated tests (Rice et al., 2017)

BDD, TDD and ATDD

There are difference software engineering approaches to delivering systems through requirements: behaviour-drive (BDD), test-driven (TDD), and acceptance-driven (ATDD). These approaches where introduced by the Extreme Programming (XP) framework where TDD is developer-focused (**inside-out, implementation**), BDD is customer-focused (**outside-in, behaviour**) and ATDD is team-focused (Moe, 2019). ATDD is best supported by the given-When-Then format used by Gherkin statements.

BDD is an agile software development methodology that encourages collaboration between project stakeholders like developers, software quality engineers, business analysts and non-technical stakeholders. It encourages conversation and concrete examples to formalize a shared understanding of how the application should behave and is based on TDD principles with ideas from domain-driven design and object-oriented analysis and design (Bruschi et al., 2019).

User Stories and BDD

According to Lucassen et al. (2015), user stories capture the “essential elements of a requirement: *who* it is for, *what* it expects from the system, and, optionally, *why* it is important.” The user stories are then accompanied by acceptance tests, which are generally written using BDD scenarios that consist of several steps. In developing BDD scenarios it is imperative to ensure they are not detailed, have few steps, are declarative, clear, avoid vague statements, deliver business value, and follow good English structure.

I continued to contribute to fellow student's research about project failures and found the biggest takeaway to preventing project failure is communication and management buy-in. Our team held (weekly) meeting to keep each other updated about requirements for delivery in unit 6 and general discussions about the module and project management. Overall I enjoyed this unit's introduction into the considerations around requirements gathering.

References

Bruschi, S., Le Xiao, M.K. & Jimenez-Maggiora, G. (2019). Behavior driven development (BDD): a case study in healthtech. *Pacific NW Software Quality Conference*.

Rice, B., Jones, R. & Engel, J. (2017). Welcome to behave! Available from https://behave.readthedocs.io/en/stable/index.html [Accessed 2 April 2022]

Lucassen, G., Dalpiaz, F., Van Der Werf, J.M.E. & Brinkkemper, S. (2015). Forging high-quality user stories: towards a discipline for agile requirements. *2015 IEEE 23rd international requirements engineering conference (RE):*126-135.

Moe, M.M. (2019). Comparative study of test-driven development (TDD), behavior-driven development (BDD) and acceptance test–driven development (ATDD). International Journal of Trend in Scientific Research and Development, (1):231-234.