

- 1. I use it to talk about three areas primarily - progressive elaboration of designs and requirements, waterfall engineering processes, and the importance of testing and verifying requirements in development. Which of these three areas feels the most important in your experience with software projects? What benefits, issues, and practices do you feel are important to that specific area?**

Testing is the activity of checking whether actual results meet expected results and identify if products satisfy the requirements. Different with elaboration of designs and waterfall engineer processes that would likely be used “during” implementation process, testing is the method that happens “after” finishing the initial implementation. Therefore, in my opinion, testing and verifying requirements is the most significant area in software development, and the rest of the paper will analyze reasons in failure costs and software quality perspectives.

Testing is important because the cost of software failure is huge and irreversible. On April 26, 1994, a Chinese airline A300 crashed due to a software bug, causing 264 people dead(Tomas, 2016). In April 1999, a software bug caused the launch of a \$1.2 billion military satellite to fail(Tomas, 2016). If the software tester could find those bugs and improve the quality of the software before their products are published, it is possible those disasters would be prevented. Therefore, software testing can not only find bugs , but also prevent problems and reduce risks.

Software testing could ensure that the published products meet users requirements. The quality itself means that the software meets the requirements, the beauty or exquisite skills of the code does not mean the high quality of the software(Crosby, n.d.). It is possible that the software has really comprehensive functions, but the user interface and accessibility are terrible. Also, different countries/companies may have different requirements for the same software, it is essential for testers to ensure the application meets the requirements of different standards and operating habits. Therefore, as a software tester, the most important goal is to start from the use of customers: testing programs from the customer's point of view, thinking how customers will use the product and consider what problems will customers face.

To conclude, testing is an indispensable part in software development. By checking the code comprehensively, programmers could reduce the risk of failure and improve the quality of the product. Just like hardware products need a perfect product carrier, an excellent software product not only needs to have powerful functionality, but also needs to bring a comfortable user experience to customers.

2. In our kickoff of our projects, we worked with several planning tools: project charter, project brief, WBS, requirements. There were also some tools for planning we did not use: project plans, Gantt charts, etc. Which one project management tool do you feel is the best contributor to establishing the start of a project? What benefits, issues, and practices for that tool do you think are important to be aware of?

The project charter is a document mostly issued by the project sponsors. It gives the main requirements and guidance for the project work (from the lecture slide). Also, the project charter stipulates the authority of the project manager and the resources availability for the project (from the lecture slide). Different with project WBS and requirements that reveal the scope of the project procedure and decompose the work, project charter specifies the goal for the software development that sponsors want to achieve. In addition to that, Bhagia has concluded the importance of project charter: before the project charter is approved, the project is not a project (2018). Therefore, in my opinion, the project charter is the symbol of the start of software development.

Project charter is significant because it formally authorizes the project manager for the application. Project manager plays an indispensable role in the process of implementing a software application. Project charter helps the manager understand the direction of the application, also, it specifies the needs of the project and what goal to achieve to be considered successful. In addition, by reading the charter, managers would have a better understanding of work splitting and resources management, if there is a problem in the process of implementation, it would be easier for the manager to find the right person to talk. Thus, project charter solves the management problem to a certain extent .

Different with project contract, project charter does not include any money and benefits related content, therefore, it is a document for software developers that explains the content of the application. Statement-of-work(SOW) plays an important role in project charter. SOW is the description of the product, the main content includes project requirements, project output and work explanation, it tells programmers the expectations, deliverable and timeline of the application (Fournier, 2018). By writing an efficient project charter, it would help reduce risk in software development. According to Fournier, by writing all the contents in the work list, SOW could ensure developers and sponsors covered by insurance so that it decreases the risk of fines and penalties associated with developers and non-permanent workers (2018).

To conclude, project charter is an important component of software development, it recognizes the existence of the project and allows the project manager to start the

Xinyu Jiang
CSCI5040 Midterm
10/21/2020

project. In addition, state-of-work tells the requirements of the software development. Therefore, project charter is the fundamental guidance of the software application and best contributes to the start of the project.

3. We will be spending most of the two semesters of our software work using Agile methods for project management, control, and monitoring. We have started a Scrum process that we will use through the middle of the second semester, and then we will shift to Kanban for the tail end of the next semester – so you can experience both. How do you feel about the effectiveness of Agile methods, either in general, or a particular methods like Scrum or Kanban? What benefits, issues, and practices for the method do you feel are most important to consider for keeping the method working?

Agile Development is a human-centric, iterative and gradual software development method. It will guide programmers to complete the development step by step through collaboration between self-organizing teams (What is AGILE? - What is SCRUM? , n.d.). Scrum is a software application method that extends the principles of agile method, it is a lightweight development for aAgile development; To be general, Scrum development is a subset of agile method (What is AGILE? - What is SCRUM? , n.d.). By using Scrum, the project team could finish the work efficiently. However, because the Scrum implementation period is usually short, it also brings some problems, and the rest of the paper will analyze Scrum development, and suggest ways to keep Scrum working.

The most significant benefit is that Scrum application improves the efficiency of the project team. In Scrum, products' requirements are defined as product backlogs. All product backlogs start from a simple idea and gradually refined until it can be implemented(sprint backlog). In the sprint backlogs, it specifies the priority of the work, so that programmers know the task for the next step. Thus, Scrum applications could frequently give sponsors semi-finished products that contain functions that can work. By doing that, it helps customers verify if the product meets the requirement and allows the application to change requirements to adapt to changing needs. Also, it reduces the cost of requirement change compared to changing the implementation after the base is done.

There are also disadvantages of the Scrum method. Different from Waterfall applications that use documents as project guidance, Agile method pays attention to communication and ignores the importance of documentation (Stein, 2018). Since scrum applications are built based on project meetings, it is difficult to start the work through communication with a lot of people, thus, using Scrum in a large project team would be quite inefficient. In addition to that, because of high frequent meetings, if a programmer cannot finish his work, it would be restless to join a meeting. In that case, Scrum would also bring high pressure to programmers.

To conclude, there are a lot of advantages of using the Scrum method in software applications, but it also brings many problems. Scrum development is especially suitable for scenarios where the requirements are difficult to meet at one time, by dynamically changing the implementation, the programmer would generally achieve the goal. In addition, Scrum application is suitable for small project teams that have a comfortable communication

Xinyu Jiang
CSCI5040 Midterm
10/21/2020

atmosphere. It is difficult for a large project team to start the work. Therefore, before deciding to use the Scrum method in a software development, it is necessary to consider the background of the project team, and choose the most suitable application method.

References

1. Bhagia, R. (2018, October 10). What is the Importance and Benefits of The Project Charter? Retrieved October 21, 2020, from <https://www.knowledgehut.com/blog/project-management/importance-and-benefits-of-the-project-charter>
2. Crosby, P. (n.d.). The Fun Uncle of the Quality Revolution. Retrieved October 20, 2020, from <https://www.skymark.com/resources/leaders/crosby.asp>
3. Fournier, J. (2018, September 14). What is a SOW and how can SOW Management Benefit my Contingent Workforce Program? Retrieved October 21, 2020, from <https://www.hcmworks.com/blog/what-is-a-sow-and-how-can-sow-management-benefit-my-contingent-workforce-program>
4. Stein, D. (2018, September 28). The Importance of Documentation: Agile vs Waterfall. Retrieved October 21, 2020, from <https://medium.com/@darrenrothstein/the-importance-of-documentation-agile-vs-waterfall-727124380c4e>
5. Thomas, N. (2016, July 04). What is Testing and why is it Important? Retrieved October 20, 2020, from <https://webandcrafts.com/blog/what-is-testing-and-why-is-it-important/>
6. What is AGILE? - What is SCRUM? (n.d.). Retrieved October 21, 2020, from <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>