Week_4_pf
Explain the importance of quality assurance in software testing and the challenges you might encounter before and after QA.

From what we learnt so far is that quality assurance is a part of SDLC. It is so important to ensure that a software works as required, reliable and secured.
Sanika in his post defines: “Software quality assurance (SQA) is a methodology to ensure that the quality of the software product complies with a predetermined set of standards.”. I completely agree with him: “SQA tests every block of this process individually to identify issues before they become major problems.”. He also explained SQA principles:

- Defect prevention - it means trying to discover any critical issues in early stages of the development. This principle emphasizes on finding and fixing a root cause and not symptoms.
- Continuous improvement - like in Japanese philosophy any person should continuously improve himself, little by little. We cannot write software which is good from the scratch and bugless. One of the killing aspects is time. Our customers want a program as fast as possible. So there are always trade offs.
- Stakeholder involvement - no secret that without feedback it is impossible to create something meaningful and good. Everybody: developers, customers, testers, architects, managers etc should be involved, to provide constructive feedback and ideas.
- Risk-based approach. SImply to prioritize the tasks.

One of the most structural mechanisms is - quality standard. It includes: requirements, acceptance criteria and performance metrics. (Sanika J. April 20, 2023).
From the writing assignment, is a testing plan for each feature. Such a plan should be reviewed by all stakeholders, especially by developers and testers. Next a test. Testing should include: unit testing, integration testing, system testing, e2e testing and even accessibility testing.

Every single defect should be tracked in Jira and set the importance and priority of fixed.

I think the most important challenge is keeping high quality software. For example, critical bugs make our customers unhappy. Fixing such bugs requires a lot of money and effort. These spoil the company reputation.

I think there is no such thing as an after QA. I see the entire process as continuous movement with cycles. During development of a new feature we can enter a bug in existing functionality. Another interesting phenomenon is the “Pesticide paradox”. It happens when testing / unit testing always tests the same thing. For example: The same URL which does not contain special characters. A problem with this is that we never know that if we insert special characters, the urls will be incorrectly parsed and requests won’t serve as expected. One piece of advice is to use random strings, to add more unit tests especially when bugs are opened.

Self - reflection
This week it was challenging to write a test plan for our site for login page, search and a basket. I have added a login page and searched into github pages, however a shopping card taken from amazon. Easier to work with something real.
Discussion forum was the next challenge. To find an article was not a problem, the problem was to summarize. The guys did a lot of research from different areas.
Bottom line, this week was a great experience.

References:

1. What is software quality and why it is important - https://www.turing.com/blog/software-quality-assurance-and-its-importance/
2. The software challenges - https://ieeexplore.ieee.org/document/8343638
3.