

Week7\_reading

The different types of software testing

- Manual vs. Automated testing
  - Manual testing is done in person**, by clicking through the application or interacting with the software and APIs with the appropriate tooling.
  - Automated tests** are performed by a machine that executes **a test script that was written in advance**.
  - Automated testing is a key component of continuous integration and continuous delivery
- The different types of tests
  - Unit tests
    - They consist in **testing individual methods and functions** of the classes, components, or modules used by your software.
  - Integration tests
    - Integration tests verify that **different modules** or services used by your application **work well together**.
  - Functional tests
    - Functional tests focus on the **business requirements** of an application.
    - They **only verify the output of an action** and do not check the intermediate states of the system when performing that action.
  - End-to-end tests
    - End-to-end testing **replicates a user behavior** with the software in a complete application environment.
  - Acceptance testing
    - Acceptance tests are **formal tests** that verify if a system **satisfies business requirements**.
  - Performance testing
    - Performance tests evaluate how a system performs **under a particular workload**.
  - Smoke testing
    - They are meant to be **quick to execute**, and their goal is to **give you the assurance** that the major features of your system are working as expected.
- How to automate your tests
  - To automate your tests, you will first need to write them **programmatically** using **a testing framework** that suits your application.
  - PHPUnit, Mocha, RSpec** are examples of testing frameworks that you can use for PHP, Javascript, and Ruby respectively.
- Exploratory testing
  - An **exploratory testing session** should not exceed two hours and should **have a clear scope** to help testers **focus on a specific area** of the software. Once all testers have been briefed, various actions should be used to check how the system behaves.
- A note about testing
  - A good testing suite should try to break your app and help understand its limit.
  - And finally, **tests are code too!** So don't forget them during **code review** as they might be the final gate to production.

Types of Software Testing

Seven Principles of Testing

- Testing shows presence of defects
  - By testing you can **show presence of defects** in a product but you can **never prove** that the product under test is **defect free**.
- Exhaustive testing is impossible
  - Testing all possible scenarios (all combinations of preconditions and inputs) is not feasible.
  - Techniques like risk analysis** should be **used to prioritize** and focus the **testing efforts**.
- Early testing
  - The testing activities should **be started as early as possible** during the software or system development life cycle.
- Defect clustering
  - The **density of modules** should be used to **decide where to focus the testing effort**.
  - Usually a small number of modules contains most of the defects.
- Pesticide paradox
  - The test cases should be **reviewed and updated** on regular basis.
  - New tests** should be developed to exercise **new parts of the software** or system.
- Testing is context dependent
  - Different kind of systems** are tested **in a different way**.
- Absence – of – errors fallacy
  - if a **system does not fulfill the user needs** and expectations then **having a bug free system does not help**.

QA Roles and Responsibilities: Who Do You Need on Your Software Testing Team?

Key QA Roles and Responsibilities

- QA Engineer
  - Tests software to **detect bugs and errors**. Checks **whether a product complies with the requirements**.
  - A detective who knows where the bugs can hide, even where no one expects them to. Tests the system using attention, deduction, and sometimes special software.
- Test Analyst
  - Guru of project documentation**. The first one to decide what to test and how. Knows exactly what the product should do.
  - Systemizes the information** to ease the QA engineer's life.
- Test Architect
  - Looks for ultimate solutions** that will meet the client's demands and align with the team's resources.
  - Has a complete vision of the software system. Knows every little feature and how it interacts with other features.
- Test Manager
  - Takes **full responsibility for the project's success (or fail)**.
  - Prepares test strategy, defines the scope of work** for other members, controls test execution.
- QA Team Lead
  - The Supervisor**. **May take part in any process mentioned above**, but usually just checks the status and manages the team. Conducts interviews. Hires and mentors new members. Deals **mostly with managerial tasks** rather than tech tasks.