

Software Testing

1. **Unit Testing**: Isolating individual code units to ensure they work as expected.

Unit Testing: Isolating individual code units to ensure they work as expected. Think of it as testing each brick before building a wall.

Integration Testing: Verifying how different modules work together. Imagine testing how the bricks fit into the wall.

System Testing: Putting it all together, ensuring the entire system functions as designed. Now, test the whole building for stability and functionality.

Acceptance Testing: The final hurdle! Here, users or stakeholders confirm the software meets their needs. Think of it as the grand opening ceremony for your building.

2. **Performance Testing**: Assessing speed, responsiveness, and scalability under different loads.

Performance Testing: Assessing speed, responsiveness, and scalability under different loads. Imagine testing how many people your building can safely accommodate.

Security Testing: Identifying and mitigating vulnerabilities to protect against cyberattacks. Think of it as installing security systems and testing their effectiveness.

Usability Testing: Evaluating how easy and intuitive the software is to use. Imagine testing how user-friendly your building is for navigation and accessibility.

3. **Regression Testing**: Ensuring new changes haven't broken existing functionality.

Regression Testing: Ensuring new changes haven't broken existing functionality. Imagine checking your building for cracks after renovations.

Smoke Testing: A quick sanity check to ensure basic functionality before further testing. Think of turning on the lights and checking for basic systems functionality before a deeper inspection.

Exploratory Testing: Unstructured, creative testing to uncover unexpected issues. Imagine a detective searching for hidden clues in your building.

Revision #1

Created 23 April 2024 12:28:51 by sedawk

Updated 23 April 2024 12:29:19 by sedawk