

Common manual testing techniques:

Common of Manual testing techniques are

- Black Box Testing
- White Box Testing
- Gray Box Testing
- Acceptance Testing
- Integration Testing
- Unit Testing
- Sanity Testing
- Smoke Testing
- Regression Testing
- User Interface (UI) Testing

Black box testing:

Black Box testing is a process of, Without the coding knowledge, checking an application if it is working properly or not.

White box testing:

White Box testing is done with knowledge of the internal structure of program. White Box Testing is done by developers at the coding level.

Gray Box Testing

Gray box testing combines both black box and white box testing methodologies in one. In gray box testing, software developers are provided with a limited knowledge of the internal structure of the application.

Acceptance Testing

Acceptance testing refers to an important stage where a system is checked to ensure whether it meets the required specifications and business objectives before its deployment to production.

Integration Testing

Testing the interface between the modules.

Unit Testing

Unit testing done by the developer. A small unit of module is an application is tested

Sanity Testing

Sanity testing, a subset of regression testing, is a quick, focused testing effort to determine whether a particular section of the application is still functioning after a minor change or a bug fix.

Smoke Testing

It is a process of validating the basic functionality in a build is working or not. If the smoke testing is pass, then the build is approved for performing the remaining testing. If the smoke testing is a fail, the mail will be sent to the Development Lead and Manager. Smoke test are generally positive test cases and include general validations

Regression Testing

Regression testing is a process of confirming that the code changes done for the CR and IR is not negatively impacting the existing functionalities. By executing all the test cases of other functionalities on the new build, regression testing is done. If all the test cases are a pass, it means that the code changes are not impacting the existing other functionalities. If any one of the test case fails, it means that the existing functionality gets affected due to this code change.

User Interface (UI) Testing

The process of ui testing includes where the software testers can know if the application is functioning correctly. It can be performed manually by a software tester, as well as can be performed automatically with the help of using a software program. This software testing type ensures that the application meets the specifications, functions correctly and has a satisfactory user interface & experience.

Boundary value analysis

Boundary value analysis is used for developing test cases to test the boundaries which separates the continuous range of inputs. The boundaries are identified as part of this technique and then the test cases are written. For any scenario

Lower boundary cases (using values just below the boundary specified)

Upper Boundary cases (Using values just above the boundary specified)

On boundary cases (Using the boundary values)

Formula for finding the lower, upper and on boundary cases: min-1, min, min+1, max, max-1, max+1

Decision table testing

Decision table is one of the testing design techniques which will help us test the system's behavior with various set of inputs. This technique can also be called as Cause and Effect table, as it captures the driving factor (cause, input data) and the captures the effect (output) in the tabular format.

The future of manual testing in the age of AI

The future of manual testing in the age of AI is an evolving topic. While AI and automation tools are rapidly transforming the landscape of software testing, manual testing is unlikely to disappear entirely. Here's how it's shaping up:

AI-Powered Testing Augments Manual Testing

- AI tools can assist manual testers by identifying high-risk areas that require human intervention, based on patterns they learn from large datasets.

Manual Testing

- While AI excels at automating routine tasks, manual testers will still be needed for tasks that require creativity, intuition, and understanding of user experience.
- Human testers are crucial in understanding how users interact with an app in real-world scenarios, especially for things like usability and accessibility testing.

Increased Collaboration Between Humans and Machines

- AI tools will act as assistants, streamlining repetitive tasks and freeing up human testers to focus on the most complex issues. The collaboration could look like a hybrid testing approach, where AI handles repetitive checks, while humans take over tasks that involve critical thinking and complex decision-making.

Improved Test Coverage with AI

- AI can help generate better test coverage by analysing past defects, user interactions, and code quality. With its pattern recognition abilities, AI tools can suggest areas that might have been overlooked in manual testing.

Shift in Skillsets for Manual Testers

- With AI and automation tools handling more of the basic testing tasks, manual testers will need to evolve their skills. Future manual testers may need to develop an understanding of AI, machine learning, and how these tools work to effectively collaborate with them.
- Manual testers may shift toward roles in **test design**, **test strategy**, and **test optimisation** rather than executing basic tests.

AI Challenges

- Despite its promise, AI isn't perfect for all scenarios. Manual testers may still be needed to fill gaps in areas where AI struggles, such as handling ambiguous or unpredictable behaviour or understanding nuanced design issues that AI might not fully grasp.
- Testers will still be required to verify AI-generated test results, ensuring that the tool's behaviour aligns with real-world conditions.

Cost and Efficiency Benefits

- Over time, automation and AI can drive down the cost of manual testing by reducing the number of routine tasks and increasing testing speed. However, this might also mean fewer opportunities for traditional manual testing roles unless testers adapt to the new tools.

1. **Common manual testing techniques:** <https://medium.com/@balakumarjeeva/common-manual-testing-techniques-eff7e2e41d9b>
2. **Boundary value analysis:** <https://medium.com/@balakumarjeeva/boundary-value-analysis-boundary-value-analysis-is-used-for-developing-test-cases-to-test-the-293d592dbbed>
3. **Decision table testing:** <https://medium.com/@balakumarjeeva/decision-table-testing-decision-table-is-one-of-the-testing-design-techniques-which-will-help-us-dcc332b81897>
4. **The future of manual testing in the age of AI:** <https://dev.to/balakumar/the-future-of-manual-testing-in-the-age-of-ai-10cm>