ONE PAGE SUMMARY

IT & Software › Software Testing › Exploratory Testing

Introduction to Exploratory Testing

Unlocking software quality through exploratory testing.

DEFINITION

> Exploratory testing is an approach where testers actively explore the software to identify issues. It combines learning, test design, and execution in real-time.

KEY CONCEPTS

Charters

Charters are brief descriptions of what to test, guiding the exploratory session.

Session-Based Testing

This technique involves time-boxed sessions where testers focus on specific areas of the application.

Documentation

While exploratory testing is less formal, documenting findings and insights is crucial for future reference.

Critical Thinking

Testers use critical thinking skills to hypothesize about potential issues and explore the software accordingly.

Feedback Loop

Exploratory testing encourages continuous feedback between testers and developers to improve software quality.

EXAMPLES

→ Testing a new feature by navigating through the UI without a predefined script.

→ Identifying usability issues by observing how real users interact with the application.

→ Exploring edge cases by intentionally inputting unexpected data.

MEMORY TIPS

★ Think of 'explore' as 'discover' to remember the goal of exploratory testing.

★ Use the acronym 'CSD' for Charters, Session-Based, and Documentation.

★ Visualize a treasure map to recall the exploratory journey through software.

COMMON MISTAKES

✗ Failing to document findings during testing sessions.

✗ Not having a clear charter leading to aimless exploration.

✗ Overlooking critical thinking, relying solely on scripted tests.

QUICK RECAP

Exploratory testing is a dynamic approach that emphasizes real-time exploration and critical thinking. It involves creating charters, conducting session-based testing, and documenting findings to enhance software quality.