ONE PAGE SUMMARY

IT & Software › Software Testing › Mutation Testing

Introduction to Mutation Testing

Mutation Testing: Strengthening Software Quality Through Intentional Errors

DEFINITION

> Mutation testing is a software testing technique that evaluates the effectiveness of tests by introducing small changes, or mutations, to the code. It helps identify weaknesses in test cases and ensures better software reliability.

KEY CONCEPTS

Mutants

Mutants are modified versions of the original program created by making small changes to the code.

Test Suite

A test suite is a collection of test cases designed to validate the functionality of the software.

Mutation Score

The mutation score measures the percentage of mutants that were detected by the test suite, indicating its effectiveness.

Equivalent Mutants

Equivalent mutants are changes that do not affect the program's behavior and should be ignored in testing.

Mutation Operators

Mutation operators are rules that define how to create mutants, such as changing arithmetic operators or altering variable values.

EXAMPLES

→ Changing '+' to '-' in a calculation function.

→ Replacing a conditional check from '==' to '!='.

→ Removing a return statement in a function.

MEMORY TIPS

★ Think of 'mutate' as 'modify' to remember the purpose of mutation testing.

★ Use the acronym MUTE: Mutants, Uncover, Test, Evaluate.

★ Visualize a 'mutation' as a small bug introduced to find hidden flaws.

COMMON MISTAKES

✗ Overlooking equivalent mutants, leading to inaccurate mutation scores.

✗ Not updating the test suite after adding new features.

✗ Focusing only on high mutation scores without analyzing test quality.

QUICK RECAP

Mutation testing enhances software testing by introducing intentional errors to assess test effectiveness. By analyzing how well tests detect these errors, developers can improve software reliability and quality.