**Verification and Validation**

**In Software Testing**

Verification and Validation (V&V) in software testing are crucial processes to ensure that software meets both its specified requirements and user expectations. Here's a breakdown of each:

**Verification Testing:** Verification testing focuses on evaluating software artifacts (requirements, design, code, etc.) to ensure they adhere to specified requirements and standards. It involves activities such as code reviews, walkthroughs, inspections, and analyzing design and specification documents. The main objectives of verification testing are to confirm that the software is built correctly according to the intended design and that it aligns with organizational and team-specific standards. Verification testing is performed at every stage of development before implementing any features.

**Advantages of Verification Testing:**

* Early detection of bugs and defects.
* Provides insight into potential development needs.
* Helps in estimating and preparing for emerging issues.
* Keeps software aligned with customer and business requirements.
* Reduces unnecessary work in later stages of development.

**When to Use Verification Testing:** Verification tests must be conducted at every stage of development before implementing any features. For example, before creating a feature like an "Add to Cart" button, verification tests would review requirements and design documents to ensure compliance with specifications.

**Validation Testing:** Validation testing occurs at the end of a specific module or after the software has been entirely built. Its primary purpose is to ensure that the final product meets user needs and expectations. Validation tests include various forms of testing such as unit testing, integration testing, system testing, and user acceptance testing. The goal is to verify that the software functions as intended and satisfies the user's requirements under different conditions.

**Advantages of Validation Testing:**

* Detects bugs missed during verification.
* Reveals inadequacies in specifications.
* Ensures software meets customer demands and expectations.
* Validates software for cross-browser compatibility.

**When to Use Validation Testing:** Validation tests should be conducted after every feature or step in the development process is completed. For example, unit tests are run after creating each unit of code, while integration tests are performed after combining multiple modules.

**Difference Between Verification and Validation:**

* Verification is static and focuses on studying and verifying requirements, while validation is dynamic and tests the final product.
* Verification doesn't require executing code, while validation always requires executing code.
* Verification involves human verification of assets, while validation involves both human and machine-based checking.
* Verification detects bugs at the beginning of each development phase, while validation detects unnoticed bugs at the verification stage.
* Verification targets documents and design artifacts, while validation targets the final product.

**Key Takeaways:**

* V&V are essential for ensuring software quality and meeting user expectations.
* Thorough V&V activities help filter out bugs and improve user experience.
* Validation testing should include testing on real device-browser-OS combinations to improve accuracy and performance.