*NOTES:*

A list of some common types of software testing, organized by their purpose:

**Verifying functionality:**

- **Unit testing**: verifies the functionality of individual units or components of the software.

- **Integration testing**: verifies that different modules or services used by the software work well together.

- **Functional testing**: verifies that the software functions as intended and meets the specified requirements.

- **Acceptance testing**: verifies that the software meets the acceptance criteria and is ready for release to the customer.

- **Smoke testing**: verifies that the main features of the software work properly and exposes simple failures severe enough to reject a prospective software release.

- **Regression testing**: verifies that changes to the software do not adversely affect existing functionality.

**Evaluating performance and reliability:**

- **Performance testing**: measures how well the software performs under various conditions, such as high load or stress.

- **Load testing**: measures the performance of the software under heavy load or high user traffic.

- **Stress testing**: measures the stability and reliability of the software under extreme conditions, such as low memory or high CPU usage.

- **Recovery testing**: verifies that the software can recover from failures or errors and resume normal operation.

**Ensuring usability and accessibility:**

- **Usability testing**: evaluates how easy it is for users to use the software and accomplish their tasks.

- **Accessibility testing**: verifies that the software is accessible to users with disabilities and complies with accessibility standards.

**Verifying compatibility and installation:**

- **Compatibility testing**: verifies that the software is compatible with different operating systems, browsers, and devices.

- **Installation testing**: verifies that the software can be installed and uninstalled correctly on the target system.

**Exploring potential issues or areas for improvement:**

- **Exploratory testing**: involves exploring the software to identify potential issues or areas for improvement.

- **Ad-hoc testing**: involves randomly testing the software without any formal test plan or test cases.

**Verifying security and compliance:**

- **Security testing**: verifies that the software is secure and protects against unauthorized access or attacks.