

Nama : Muhammad Rifqi Daffa Ulhaq

NPM : 1204045

Kelas : 3B

Perihal : Materi: Pemahaman 7 prinsip testing, Functional testing, Non Functional testing, Structural testing dan Testing related to Change

- **Pemahaman 7 Prinsip Testing**

Prinsip-prinsip testing adalah seperangkat pedoman atau aturan yang diikuti untuk melakukan testing yang efektif dan efisien. Berikut adalah 7 prinsip testing:

1. Testing mengungkapkan keberadaan bug: Tujuan utama dari testing adalah untuk menemukan bug atau kelemahan dalam perangkat lunak. Jadi, testing dilakukan untuk memvalidasi bahwa perangkat lunak tersebut berfungsi dengan benar dan memenuhi persyaratan pengguna.
2. Testing tidak bisa membuktikan keberadaan ketiadaan bug: Meskipun testing dapat membantu menemukan bug, tidak mungkin untuk membuktikan bahwa perangkat lunak tidak memiliki bug sama sekali. Ini karena kompleksitas dan variasi dalam lingkungan penggunaan dan situasi yang berbeda-beda.
3. Testing harus dimulai sejak awal: Testing harus dimulai sejak tahap perancangan perangkat lunak untuk memastikan bahwa persyaratan pengguna dipahami dan diimplementasikan dengan benar. Ini membantu menghindari masalah yang lebih besar di kemudian hari.
4. Testing harus dilakukan secara terus-menerus: Testing harus dilakukan secara terus-menerus selama siklus hidup pengembangan perangkat lunak. Ini membantu menemukan bug dan masalah lebih awal sehingga dapat diperbaiki dengan cepat dan biaya yang lebih rendah.
5. Testing harus dilakukan oleh orang yang berbeda dari pengembang: Testing harus dilakukan oleh orang yang berbeda dari pengembang perangkat lunak. Hal ini membantu memastikan bahwa perangkat lunak diuji secara objektif dan bahwa semua fungsionalitas diuji.

6. Testing harus mengikuti rencana: Testing harus dilakukan sesuai dengan rencana testing yang telah disusun sebelumnya. Ini membantu memastikan bahwa semua fungsionalitas diuji dan bahwa perangkat lunak siap untuk digunakan.
7. Testing harus dihentikan ketika tujuan telah tercapai: Testing harus dihentikan ketika semua tujuan testing telah tercapai dan perangkat lunak telah terbukti siap untuk digunakan.

- Functional Testing

Functional testing adalah jenis testing yang dilakukan untuk memastikan bahwa perangkat lunak memenuhi persyaratan fungsional atau fitur yang telah ditentukan. Ini melibatkan menguji fungsionalitas perangkat lunak, seperti menjalankan fungsi, mengisi formulir, dan memverifikasi output yang dihasilkan.

Ada beberapa jenis functional testing yang biasanya dilakukan, termasuk:

1. Unit testing: Menguji setiap unit kode secara terpisah.
2. Integration testing: Menguji bagaimana unit-unit tersebut saling berinteraksi dan bekerja bersama-sama.
3. System testing: Menguji fungsionalitas sistem secara keseluruhan.
4. Acceptance testing: Menguji apakah perangkat lunak telah memenuhi persyaratan pengguna.

- Non-Functional Testing

Non-functional testing adalah jenis testing yang dilakukan untuk memastikan bahwa perangkat lunak memenuhi persyaratan non-fungsional. Beberapa contoh persyaratan non-fungsional yang harus dipenuhi oleh perangkat lunak adalah keamanan, kinerja, ketersediaan, skalabilitas, dan usabilitas.

Berikut adalah beberapa jenis non-functional testing:

1. Performance testing: Memeriksa kinerja perangkat lunak dalam berbagai situasi beban.
2. Security testing: Memeriksa keamanan perangkat lunak dengan menguji rentang kontrol akses, kerentanan, dan pengaturan keamanan lainnya.

3. Compatibility testing: Memeriksa apakah perangkat lunak kompatibel dengan berbagai sistem operasi, perangkat keras, dan perangkat lunak lainnya.
4. Usability testing: Memeriksa kemudahan penggunaan perangkat lunak dan sejauh mana user interface memenuhi kebutuhan pengguna.
5. Scalability testing: Memeriksa kemampuan perangkat lunak untuk mengatasi peningkatan jumlah pengguna atau volume data.
6. Recovery testing: Memeriksa kemampuan perangkat lunak untuk memulihkan diri dari kegagalan atau situasi darurat.

- Structural Testing

Structural testing adalah jenis testing yang dilakukan untuk memastikan bahwa struktur kode perangkat lunak telah diuji dengan benar. Tujuannya adalah untuk menemukan kesalahan atau kelemahan dalam kode dan memastikan bahwa setiap jalur kode telah diuji secara menyeluruh.

Berikut adalah beberapa jenis structural testing:

1. Statement coverage: Memeriksa apakah setiap baris kode telah diuji.
2. Branch coverage: Memeriksa apakah setiap cabang dalam kode telah diuji.
3. Condition coverage: Memeriksa apakah setiap kondisi dalam kode telah diuji.
4. Path coverage: Memeriksa apakah setiap jalur dalam kode telah diuji.

- Testing Related to Change

Testing related to change adalah jenis testing yang dilakukan ketika ada perubahan pada perangkat lunak, seperti penambahan fitur atau perbaikan bug. Tujuannya adalah untuk memastikan bahwa perangkat lunak masih berfungsi dengan baik dan bahwa perubahan tersebut tidak mempengaruhi fungsionalitas yang sudah ada sebelumnya.

Beberapa jenis testing yang dilakukan dalam testing related to change adalah:

1. Regression testing: Menguji perangkat lunak untuk memastikan bahwa perubahan baru tidak mempengaruhi fungsionalitas yang sudah ada sebelumnya.
2. Integration testing: Menguji bagaimana unit-unit dalam perangkat lunak saling berinteraksi setelah terjadi perubahan.

3. Acceptance testing: Menguji apakah perubahan telah memenuhi persyaratan pengguna sebelum diimplementasikan secara penuh.
4. Smoke testing: Menguji bahwa perangkat lunak masih dapat berfungsi dengan baik setelah perubahan dilakukan.