

Test scenario adalah tahapan-tahapan atau urutan-urutan untuk menguji fitur-fitur yang ada di dalam system/aplikasi. Test scenario mempunyai karakteristik lebih umum atau luas, tidak spesifik secara detail, dan tidak menyebutkan urutan langkah-langkah dalam melakukan pengujian, maka dari itu hasil yang diharapkan akan memiliki pemahaman yang berbeda. Test scenario merupakan uji coba secara end to end atau langsung ke tujuan yang ingin dicapai tanpa harus melakukan langkah-langkah pengujian secara detail. Secara garis besar test scenario berfokus pada “kondisi apa yang akan diuji” daripada “cara atau urutan urutan apa saja untuk menguji”.

Fitur adalah sejumlah kelebihan atau menu unggulan yang ditawarkan dari system/aplikasi tersebut kepada pengguna untuk mempermudah ataupun memberikan pengalaman yang berbeda dari system/aplikasi lain yang sejenis, tentu saja diharapkan dapat memenuhi ekspektasi pengguna ataupun memberikan rasa kepuasan kepada pengguna sehingga dapat menjadi alat untuk daya Tarik terhadap system/aplikasi tersebut.

Positive scenario = adalah pengujian yang dilakukan hanya untuk menjalankan scenario test dengan hanya untuk memastikan data yang diinputkan merupakan data yang valid dan benar sesuai dengan requirements yang telah ditentukan sebelumnya. Contoh, jika ada suatu kotak isian yang diharuskan untuk memasukkan suatu bilangan, maka kita harus memasukkan sesuai persyaratan tersebut untuk memastikan system/aplikasi tersebut dapat berjalan dengan semestinya dan diharapkan dapat sesuai dengan kondisi yang ada.

Negative scenario = adalah pengujian yang dilakukan dengan menginputkan nilai yang tidak valid atau tidak tepat, lebih tepatnya berlawanan dari requirements yang telah ditentukan. Negative scenario juga sebaiknya dilakukan dengan berulang dan lebih banyak variasi agar lebih banyak menemukan permasalahan dan menguji apakah system/aplikasi tersebut tetap berjalan dengan stabil meskipun terdapat banyak kesalahan yang dilakukan oleh end user. Selain itu juga agar ditemukan lebih banyak permasalahan/bug sehingga dapat dilakukan improve oleh developer yang lebih baik.

Unit Testing

- Merupakan proses pengujian yang dilakukan untuk memeriksa bahwa kode program yang ditulis telah memberikan output yang benar dan sesuai berdasarkan nilai inputan yang diberikan. Pada tahap ini masing-masing dari code akan diuji untuk menentukan apakah fungsi dari code-code tersebut berjalan dengan semestinya atau tidak.

Integration Testing

- Merupakan proses pengujian dimana masing-masing unit/fungsi (yang berinteraksi dengan system/aplikasi) tersebut digabungkan dan diuji untuk memverifikasi apakah dapat berfungsi sebagaimana mestinya ketika terintegrasi. Integration testing juga berperan untuk mengidentifikasi kesalahan yang tidak dapat ditemukan dari unit testing.

Stress Testing

- Merupakan proses pengujian yang dilakukan untuk mengetahui sekuat apa aplikasi/system kita dalam menangani permintaan data ataupun pengunjung yang mengunjungi aplikasi/system kita tersebut. Pengujian ini biasanya dilakukan dalam menempatkan jumlah besar di luar kapasitas

penyimpanan, permintaan basis data yang kompleks, sekuat apa server dalam menangani visitor. Tujuan dari pengujian ini adalah untuk mengetahui bagaimana feedback dari system tersebut apabila mengalami kondisi tersebut.

UAT (User Acceptance Test)

- Pengujian ini dilakukan oleh end user atau klien untuk memastikan /melakukan verifikasi system, tujuannya adalah untuk memvalidasi perangkat lunak terhadap proses bisnis yang terjadi. UAT biasanya dilakukan pada tahapan proses pengujian akhir sebelum system diserahkan kepada user.

Usability Testing

- Proses pengujian yang dilakukan untuk memastikan apakah software tersebut sudah sesuai dengan requirements dari penggunaanya atau memenuhi ekspektasi yang diinginkan oleh user. Tujuan dari usability testing ini adalah untuk mencari permasalahan, mengumpulkan data kualitatif dan kuantitatif, serta mendapatkan feedback yang cukup cepat dalam meningkatkan user interface dan mengetahui masalah dalam system/aplikasi tersebut.

Menurut saya, pyramid testing lebih baik. Pyramid testing dapat membantu Developer maupun Q.A dalam membuat perangkat lunak yang berkualitas. Pyramid testing lebih baik karena konsep yang ditawarkan dapat mengurangi waktu yang diperlukan oleh tim pengembang untuk dapat mengidentifikasi apakah perubahan yang dilakukan dapat merusak code atau tidak. Selain itu pyramid testing mengusung konsep pengujian otomatisasi testing, dimana dapat menghasilkan feedback yang cepat untuk hasil testing.

Tidak seperti anti pattern ice cream cone yang mengusung konsep pengujian manual, yang mana dapat menyebabkan waktu lama untuk mendapatkan hasil atau bug. Hal ini membuat sulit untuk menguji fungsionalitas yang ada.

Automated testing adalah metode pengujian perangkat lunak yang menggunakan suatu software testing yang bertujuan untuk membandingkan atau mendapatkan hasil antara output yang di'ekspektasikan

dengan output yang dihasilkan oleh system. Automated testing sangat bergantung dengan script yang dijalankan, apabila script tersebut ditulis dengan benar maka akan dapat memindai software lebih detail, sehingga kemungkinan besar diharapkan dapat menemukan bug yang tidak ditemukan oleh developer.

White box Testing adalah tester melihat source code program dan menemukan bugs dari kode yang akan diuji. White box testing merupakan pengujian yang harus memiliki kemampuan pemrograman, untuk memahami source code yang ditulis. White box testing juga mempunyai konsep pengujian perangkat lunak dalam hal keamanan dan kinerja program (pengujian source code program, perencanaan implementasi, keamanan, aliran data, kegagalan system).

Black Box Testing adalah metode pengujian yang mengedepankan fungsionalitas yang ada tanpa memperdulikan atau mengetahui setiap source code yang ada. Pengujian dilakukan dengan cara

menguji tampilan antarmuka aplikasi, fungsi-fungsi didalam aplikasi, dan sesuai tidaknya dengan proses bisnis yang diinginkan oleh user.

Bug severity atau tingkat keparahan bug merupakan tingkat dari dampak bug pada aplikasi yang telah diuji. Hasil pengujian keseluruhan system/aplikasi yang mengalami kegagalan atau ketidak sesuaian dari apa yang telah di ekspektasikan akan dilakukan pengurutan kedalam beberapa kategori, dari yang mulai low sampai dengan critical. Hal ini bertujuan untuk dapat mengklasifikasi kegagalan dari mana dulu yang parah atau yang penting untuk segera dilakukan perbaikan sampai dengan kegagalan mana yang bisa ditunda dulu untuk bisa dilakukan perbaikan.

Yang harus disertakan dalam pelaporan dokumen QA perangkat lunak adalah, laporan hasil testing, dimana ini mencakup scenario test dan test case, tentu saja juga dengan dilampirkan hasil dari pengujian tersebut. Positive scenario dan negative scenario juga menurut saya perlu dilibatkan dalam pelaporan dokumen tersebut. Dan yang terakhir adalah UAT (User Acceptance Test) sebagai dokumen yang mempunyai insight dari sisi hasil pengujian dari user/client.

Hal yang dapat dilakukan penguji saat pengembang membangun aplikasi adalah, penguji dapat ikut serta membantu pengujian aplikasi yang sedang dikerjakan atau sedang tahap development, hal ini bisa dilakukan dengan pendekatan framework agile, dimana setiap tahapan development dapat dengan fleksibel untuk dilakukan pengujian dan perbaikan tanpa harus menunggu system/aplikasi tersebut selesai tahapan development. Selain itu penguji dapat juga berperan sebagai end user untuk dapat membantu memberikan usulan-usulan perbaikan dari segi usability dan tampilan user interface untuk membantu perbaikan dari system/aplikasi tersebut.