1. Pada fase apa saja QE berperan pada proses SDLC? Sebutkan dan jelaskan peran seperti apa yang QE dapat lakukan pada setiap fase tersebut!
2. Menurut Anda, apa tantangan ketika menerapkan agile testing? Berikan penjelasan pada masing-masing 5 poin agile testing manifesto!

Jawaban :

1. SDLC adalah proses yang digunakan untuk mengembangkan sistem informasi SDLC

membagi semua proses pengembangan sistem informasi menjadi beberapa fase, seperti pemilihan dan perencanaan proyek, analisis, desain, serta pengujian dan implementasi. Tujuan penyelesaian setiap proses di SDLC adalah untuk memudahkan pengembang dan pengguna meninjau atau meningkatkan langkah-langkah tertentu. SDLC adalah proses yang lebih menitikberatkan pada bagian fungsi dari sistem. Contoh model pengembangan SDLC adalah model air terjun, pengembangan iteratif dan inkremental, pengembangan tangkas, kode dan perbaikan, model spiral. SDLC sendiri dibagi menjadi empat fase, yaitu: (1) desain dan pemilihan, (2) analisis sistem, (3) desain sistem, dan (4) implementasi dan operasi sistem. Model SDLC yang berfokus pada fungsionalitas sistem daripada kebutuhan pengguna, merupakan salah satu kesulitan bagi suatu organisasi untuk memahami sistem yang digunakan.

Seorang insinyur yang berkualitas harus memiliki pengetahuan menyeluruh tentang proses. Karena apa yang dilakukan, terutama pada saat proses pengujian, baik manual maupun otomatis, diharapkan berdampak signifikan terhadap SDLC. QE terlibat dalam fase perencanaan dan implementasi sistem. Perancangan sistem (Systems Design), dimana fungsi dan fungsi sistem dijelaskan secara rinci. Fungsi yang berlaku adalah:

• Analisis interaksi objek dan fungsi dalam sistem.

• Menganalisis data dan membuat skema database.

• Desain antarmuka pengguna. Implementasi sistem (Systems Implementation), yaitu. melaksanakan perencanaan langkah-langkah sebelumnya dan melakukan percobaan. Dalam aplikasinya, QE melakukan fungsi-fungsi berikut:

• Pembuatan database sesuai dengan skema desain

• Pengembangan aplikasi berdasarkan desain sistem.

• Menguji dan meningkatkan aplikasi (debugging).

1. Agile adalah kemampuan untuk menciptakan dan merespons perubahan dalam

lingkungan yang tidak pasti. Metode Pengembangan Agile adalah kerangka kerja pengembangan perangkat lunak berdasarkan nilai dan prinsip yang dijelaskan dalam Agile Development Manifesto. Pendekatan tangkas untuk pengembangan perangkat lunak relatif baru. Banyak organisasi terkemuka telah mengadopsi Agile di semua atau beberapa proyek mereka. Faktor utama tantangan implementasi agile adalah budaya perusahaan terhadap konsep agile, kurangnya keterampilan atau pengalaman dalam implementasi agile, kurangnya dukungan manajemen dan masalah hierarki dalam organisasi.

The 5-Point Agile Testing Manifesto

1. Pengujian adalah aktivitas, bukan fase. >> Di Agile, pengujian bukanlah langkah, tetapi aktivitas yang harus dilakukan bersama dengan pengkodean, dokumentasi, dan lainnya.

2. Mencegah kesalahan daripada mencari kesalahan >> Sebagian besar kesalahan muncul pada tahap posting. Biasanya terlihat seperti ini: orang membuat asumsi tentang persyaratan mereka dan menerapkannya sebelum mengklarifikasinya. Asumsi menjadi jelas hanya setelah menguji perangkat lunak dan mengidentifikasi kesalahan. Cara terbaik untuk menghindari kesalahan adalah dengan mengajukan pertanyaan.

3. Membangun sistem OVER merusak sistem >> Tujuan utama dari proses pengembangan adalah untuk menciptakan produk yang berkualitas. Untuk melakukan ini, QE harus memposisikan dirinya sebagai pengguna, mengevaluasi produk dari sudut pandang pengguna, dan dapat memberi tahu pengembang.

4. Memahami pengujian daripada memeriksa fungsionalitas >> Agile sering kali tidak disukai oleh penguji yang menganggap tugas mereka adalah memastikan produk memenuhi persyaratan. Namun secara resmi, mereka hanya memantau seberapa dekat pengembang mengikuti spesifikasi. Itu tidak benar-benar mengatakan apa-apa tentang kualitas produk atau apakah itu sesuai dengan tujuannya. Saat tugas penguji seperti ini, semua tugas pengujian dapat diotomatisasi dan tidak ada input manusia dalam prosesnya. Pengujian tangkas harus mengotomatiskan pemeriksaan sederhana sehingga penguji dapat melakukan tugas yang tidak dapat dilakukan oleh komputer atau pengujian kegunaan.

5. Tanggung jawab tim kualitas OVER kepada tanggung jawab penguji >> Mempertimbangkan tanggung jawab kualitas seluruh tim adalah prinsip dasar ketangkasan. Beban kualitas produk yang dibangun tidak boleh dipikul oleh satu anggota saja. Misalnya, jika bug baru ditemukan selama rilis produk, QE tidak bertanggung jawab untuk itu. QE tidak serta merta peduli dengan bug atau memberatkan developer. Namun, setiap anggota tim bertanggung jawab atas kepemilikan bersama atas produk yang sedang dibuat.