TUGAS KAPITA SELEKTA

Ditujukan sebagai salah satu syarat

Untuk memperoleh nilai pada kegiatan matakuliah Kapita Selekta

Program Studi DIV Teknik Informatika

1204014 – ANITA ALFI SYAHRA

PROGRAM STUDI D IV TEKNIK INFORMATIKA

UNIVERSITAS LOGISTIK DAN BISNIS INTERNATIONAL

BANDUNG

2023

3 Alasan dunia tidak butuh software testing (Haris Dermawan)

- **Alasan 1 Waktu dan Biaya Pengembangan Menjadi Lebih Lama**

a. Seven Testing Principles

- Testing shows presence of defects = tidak dapat membuktikan bahwa tidak ada cacat. pengujian harus dirancang untuk menemukan cacat sebanyak mungkin.
- Exhaustive testing is impossible = menguji semuanya (semua kondisi masukan dan prasyarat) tidak dapat dilakukan kecuali untuk kasus sepele. alih-alih pengujian menyeluruh, analisis risiko dan prioritas harus digunakan untuk memfokuskan upaya pengujian.
- Early testing = Untuk menemukan cacat lebih awal, kegiatan pengujian harus dimulai sedini mungkin dalam siklus hidup pengembangan perangkat lunak atau sistem, dan harus terfokus atau menetapkan tujuan.
- Defect clustering = Sejumlah kecil modul biasanya berisi sebagian besar cacat yang ditemukan selama pengujian prarilis, atau bertanggung jawab atas sebagian besar kegagalan operasional.
- Pesticide paradox = Jika pengujian yang sama diulang terus menerus, pada akhirnya kumpulan kasus pengujian yang sama tidak akan lagi menemukan cacat baru. Kasus pengujian perlu ditinjau dan direvisi secara berkala.
- Testing is context dependent = ID pengujian dilakukan secara berbeda dalam konteks yang berbeda. Risiko dapat menjadi faktor besar dalam menentukan jenis pengujian yang diperlukan.
- Absence-of-errors fallacy = Menemukan dan memperbaiki cacat tidak membantu jika sistem yang dibangun tidak dapat digunakan dan tidak memenuhi kebutuhan dan harapan pengguna.

b. Software Testing Life Cycle (STLC)

- Test planning = melibatkan kegiatan yang meentukan tujuan pengujian dan pendekatan untuk memenuhi tujuan pengujian dalam batasan yang ditentukan oleh konteks.
- Test monitoring and control = melibatkan perbandingan kemajuan actual yang sedang berlangsung terhadap rencana pengujian menggunakan apa pun metric pemantauan pengujian yang ditentukan dalam rencana pengujian.
- Test analysis = basis pengujian dianalisis untuk mengidentifikasi fitur yang dapat diuji dan menentukan kondisi pengujian terkait.
- Test design = merancang dan memprioritaskan test case dan set test case. Mengindentifikasi data pengujian yang diperlukan untuk mendukung kondisi pengujian

dari kasus.

- Text implementation = perangkat pengujian yang diperlukan untuk pelaksanaan pengujian dibuat dan di selesaikan. Termasuk mengurutkan kasus uji ke dalam prosedur uji.
- Test execution = mencatat id dan versi item uji atau objek uji, alat uji, dan perangkat uji. Menjalankan test secara manual atau dengan menggunakan alat eksekusi test.
- Test completion = memriksa apakah semua laporan cacar telat ditutup, memasukan permintaan perubahan atau item simpanan produk untuk setiap cacat yang masih belum terselesaikan diakhir pelaksanaan pengujian.

- **Alasan 2 Keyakinan Berlebihan pada Pengalaman Proyek Sebelumnya**

a. Solusi Keterbatasan Sumber Daya yang Dimiliki :

- Menentukan Prioritas: Fokus pada pengujian pada area yang palingpenting dan vital dalam perangkat lunak. Tentukan fitur atau fungsimana yang paling mempengaruhi keseluruhan kinerja sistem dan fokus pengujian pada area tersebut.
- Automatisasi Pengujian: Otomatisasi pengujian perangkat lunak dapat membantu perusahaan menghemat waktu dan sumber daya yang diperlukan. Beberapa jenis pengujian, seperti pengujian regresi, dapat diotomatisasi dengan alat pengujian perangkat lunak.
- Outsourcing Pengujian: Mempekerjakan layanan pengujian perangkat lunak pihak ketiga atau mengontrak pengujian ke perusahaan yang spesialis dalam pengujian perangkat lunak dapatmembantu perusahaan menghemat waktu dan sumber daya yang diperlukan.
- Menggunakan Metode Pengujian yang Efisien: Memilih metodepengujian yang efisien dan efektif dapat membantu perusahaan mengoptimalkan penggunaan sumber daya yang tersedia. Misalnya, pengujian exploratory dapat membantu perusahaan menemukan kesalahan dengan cepat dan efisien.
- Menggunakan Tools Gratis atau Open-Source: Terdapat beberapa alat pengujian perangkat lunak gratis atau open-source yang dapatmembantu perusahaan melakukan pengujian dengan biaya yang rendah.

b. Black Box Testing :

- Test condition, test case, dan data uji berasal dari test basis yangdapat mencakup persyaratan perangkat lunak, spesifikasi, kasus penggunaan, dan user story
- Test case dapat digunakan untuk mendeteksi kesenjangan antara persyaratan dan implementasi persyaratan, serta penyimpangan dari persyaratan
- Cakupan diukur berdasarkan item yang diuji dalam dasar pengujian dan teknik yang diterapkan pada dasar pengujian

- **Alasan 3 Keterbatasan Sumber Daya yang Dimiliki**

  - Menentukan Prioritas: Fokus pada pengujian pada area yang paling penting dan vital dalam perangkat lunak. Tentukan fitur atau fungsi mana yang paling mempengaruhi keseluruhan kinerja system dan focus pengujian pada area tersebut.
  - AutomatisasiPengujian: Otomatisasi pengujian perangkat lunak dapat membantu perusahaan menghemat waktu dan sumber daya yang diperlukan. Beberapa jenis pengujian, seperti pengujian regresi, dapat diotomatisasi dengan alat pengujian perangkat lunak.
  - Outsourcing Pengujian: Mempekerjakan layanan pengujian perangkat lunak pihak ketiga atau mengontrak pengujian keperusahaan yang spesialis dalam pengujian perangkat lunak dapat membantu perusahaan menghemat waktu dan sumber daya yang diperlukan
  - Menggunakan Metode Pengujian yang Efisien: Memilih metode pengujian yang efisien dan efektif dapat membantu perusahaan mengoptimalkan penggunaan sumber daya yang tersedia. Misalnya, pengujian exploratory dapat membantu perusahaan menemukan kesalahan dengan cepat dan efisien.
  - Menggunakan Tools Gratis atau Open-Source: Terdapat beberapa alat pengujian perangkat lunak gratis atau open-source yang dapat membantu perusahaan melakukan pengujian dengan biaya yang rendah.

Functional testing

- Kebutuhan tentang fungsi software secara menyeluruh
- Pemodelan dengan UML, ataupun penjelasan fiturfitur dalam bentukpernyataan masalah, adalah termasuk dalam Persyaratan Fungsional
- Diagram:
  - Gunakan Diagram Kasus
  - Diagram Aktivitas
- Pernyataan Masalah:
  - Harus mencari inventaris
  - Harus melakukan perhitungan ini

- Harus menghasilkan laporan khusus

Testing related to change

- Pengujian konfirmasi atau pengujian ulang, setelah cacat terdeteksi dan diperbaiki, perangkat lunak harus diuji ulang untuk memastikan bahwa cacat asli telah berhasil dihilangkan
- Pengujian regresi, adalah pengujian berulang dari program yang sudah diuji, setelah modifikasi, untuk menemukan kesalahan cacat diperkenalkan atau terungkap sebagai akibat dari perubahan

- Functional testing adalah jenis pengujian perangkat lunak yang menguji apakah fungsionalitas atau fitur-fitur dari perangkat lunak sudah berjalan dengan baik dan sesuai dengan spesifikasi. Contoh dari jenis ini antara lain:
    - Pengujian fungsionalitas tombol "submit" di halaman web.
    - Pengujian fungsionalitas pencarian produk di situs e-commerce.
    - Pengujian fungsionalitas login pada sebuah aplikasi.

• **Non-functional testing**
- Operational – Physical/technical environment
- Performance – Speed and reliability
- Security – Who can use the system
- Cultural & Political – Company policies, legal issues
- Non-Functional testing adalah jenis pengujian perangkat lunak yang menguji aspek-aspek non-fungsional dari perangkat lunak, seperti performa, keamanan, dan skalabilitas. Contoh dari jenis ini antara lain:
    - Pengujian performa situs web saat ada banyak pengunjung yang membukanya.
    - Pengujian keamanan aplikasi web untuk memastikan tidak ada celah keamanan yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab.
    - Pengujian skalabilitas sebuah sistem untuk memastikan sistem tersebut dapat menangani jumlah pengguna yang besar.

• **Structural testing**
- Cakupan adalah sejauh mana struktur telah dilaksanakan oleh pengujian, dinyatakan sebagai persentase dari item yang tercakup
- Jika cakupannya tidak 100%, maka pengujian lebih lanjut dapat dirancang untuk menguji item-item yang terlewat untuk meningkatkan cakupan
- Non-Functional testing adalah jenis pengujian perangkat lunak yang menguji aspek-aspek non-fungsional dari perangkat lunak, seperti performa, keamanan, dan skalabilitas. Contoh dari jenis ini antara lain:
    - Pengujian performa situs web saat ada banyak pengunjung yang membukanya.

- Pengujian keamanan aplikasi web untuk memastikan tidak ada celah keamanan yang dapat dimanfaatkan oleh pihak yang tidakbertanggung jawab.
- Pengujian skalabilitas sebuah sistem untuk memastikan sistemtersebut dapat menangani jumlah pengguna yang besar.

• **Testing related to Change**

- Pengujian konfirmasi atau pengujian ulang, setelah cacat terdeteksi dan diperbaiki, perangkat lunak harus diuji ulang untuk memastikanbahwa cacat asli telah berhasil dihilangkan
- Pengujian regresi, adalah pengujian berulang dari program yang sudahdiuji, setelah dimodifikasi, untuk menemukan cacat yang diperkenalkan atau ditemukan sebagai akibat dari perubahan
- Testing related to change adalah jenis pengujian perangkat lunak yangdilakukan saat terjadi perubahan pada kode program atau konfigurasi sistem. Contoh dari jenis ini antara lain:
    - Pengujian regresi setelah perubahan kode program untuk memastikan perubahan tersebut tidak merusak fungsionalitasyang sudah ada sebelumnya.
    - Pengujian integrasi setelah perubahan konfigurasi sistem untukmemastikan bahwa semua sistem terintegrasi dengan baik setelah perubahan tersebut.

Pengujian smoke testing setelah perubahan kecil untukmemastikan perubahan tersebut tidak mengganggu fungsionalitas utama dari perangka