

PERTEMUAN KE-13

RENCANA PENGUJIAN

13.1 TUJUAN PEMBELAJARAN :

Adapun tujuan pembelajaran yang akan dicapai sebagai berikut:

- 13.1. Pembuatan rencana uji coba dan membuat studi kasus
- 13.2. Teknik uji coba Black-box dan White-box

13.2 URAIAN MATERI

Tujuan Pembelajaran 13.1:

Pembuatan rencana uji coba dan membuat studi kasus
--

Pendekatan langsung untuk pengujian yang sempurna akan merekomendasikan rencana pengujian perangkat lunak lengkap dan komprehensif yang melakukan uji unit untuk semua unit individu, uji integrasi untuk semua integrasi unit, dan uji sistem untuk menguji paket perangkat lunak secara keseluruhan. Menerapkan rencana secara “langsung” memang memastikan kualitas perangkat lunak yang tinggi tetapi juga memerlukan investasi sumber daya yang luas dan jadwal yang panjang.

Berkaitan dengan itu, pertanyaan-pertanyaan tertentu yang umum akan muncul, sebagai contoh:

1. Apakah dibenarkan untuk melakukan pengujian unit untuk sebuah modul yang terdiri dari 98% perangkat lunak yang reuse?
2. Apakah pengujian unit wajib dilakukan pada modul sederhana yang menampilkan modul versi 12 dari modul dasar yang digunakan berulang kali oleh tim pengembang selama tiga tahun terakhir?

Hanya dalam kasus yang jarang itu dibenarkan untuk menguji “segalanya”. Biasanya, kelayakan pengujian “segalanya” sangat terbatas. Terlepas dari kinerja dari daftar tes yang ditentukan dalam kontrak atau diperlukan oleh prosedur

pengembang (misalnya beban tes untuk sistem secara keseluruhan), beberapa pertimbangan untuk tes yang akan diterapkan, antara lain :

1. Unit modul mana yang harus diuji?
2. Integrasi mana yang harus diuji?
3. Prioritas menentukan pengalokasian sumber daya pengujian kepada sistem aplikasi perangkat lunak individu. Akibatnya, aplikasi yang mendapat prioritas rendah diuji dengan hanya beberapa jenis tes atau tidak termasuk dalam pengujian sistem sama sekali.

Dalam menentukan apa yang harus dimasukkan dan apa yang dikecualikan dari uji sistem, uji unit dan uji integrasi yang sudah direncanakan harus dipertimbangkan. Untuk kualitas perangkat lunak dari aplikasi dan modul yang tidak dicakup oleh uji unit, integrasi dan sistem, kami mengandalkan cek kode yang dilakukan oleh programmer dan pemimpin timnya serta inspeksi kode dan walkthrough diprakarsai oleh tim pengembangan.

Metode untuk unit rating (modul), integrasi dan aplikasi dalam menentukan prioritas rencana pengujian didasarkan pada dua faktor:

- Faktor A: Keparahan tingkat kerusakan. Seberapa parah tingkat yang dihasilkan jika suatu modul atau aplikasi gagal.
- Faktor B: Tingkat resiko perangkat lunak. Tingkat resiko merupakan probabilitas kegagalan. Dalam rangka untuk menentukan tingkat risiko unit, modul, integrasi atau aplikasi, isu-isu yang mempengaruhi risiko memerlukan pemeriksaan. Masalah ini dapat diklasifikasikan sebagai modul / masalah aplikasi dan programmer.

Isu yang mempengaruhi tingkat risiko perangkat lunak

1. Masalah Modul / aplikasi

- Magnitude
- Kompleksitas dan kesulitan
- Persentase dari perangkat lunak asli (vs persentase perangkat lunak yang digunakan kembali)

2. Masalah Programmer

- Kualifikasi profesional
- Pengalaman pada subjek modul tertentu
- Ketersediaan dukungan profesional (cadangan dari pengetahuan dan pengalaman)
- Kenalan dengan programmer dan kemampuan untuk mengevaluasi kemampuannya

Penting untuk dicatat bahwa ada masalah yang merupakan kasus khusus dari masalah umum dalam menentukan intensitas kegiatan jaminan mutu yang cukup. Oleh karena itu, ada logika yang mendasari dalam mencapai prioritas pengujian perangkat lunak yang lebih tinggi.

Para perencana harus mempertimbangkan dua sumber utama kasus uji – contoh kasus uji kehidupan nyata dan / atau uji kasus sintetik – yang paling tepat dengan kebutuhan mereka. Setiap komponen dari rencana pengujian, berurusan dengan pengujian unit, integrasi atau sistem, membutuhkan keputusan individu tentang uji kasus dan sumber mereka masing-masing :

- Penggunaan sumber tunggal atau gabungan kasus uji atau keduanya
- Berapa banyak uji kasus dari masing-masing sumber harus dipersiapkan
- Karakteristik uji kasus

Siapa yang akan melakukan berbagai tes ditentukan pada tahap perencanaan:

- Uji integrasi, terutama uji unit, umumnya dilakukan oleh tim pengembangan perangkat lunak. Dalam beberapa kasus itu adalah unit pengujian yang melakukan pengujian.
- Uji sistem biasanya dilakukan oleh tim uji independen (pengujian tim internal atau pengujian tim konsultan eksternal).
- Dalam kasus sistem perangkat lunak yang lebih besar, lebih dari satu tim pengujian dapat digunakan untuk melakukan pengujian sistem. Keputusan prasyarat yang akan dibuat pada kasus yang harus dipertimbangkan adalah alokasi tes sistem antara internal dan tim pengujian eksternal.

Tujuan Pembelajaran 13.2:

Teknik uji coba Black-box dan White-box

Dua macam pendekatan pengujian yaitu :

1. Black Box Testing

Test case ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemutakhirannya.

Teknik pengujian black-box berfokus pada domain informasi dari perangkat lunak, dengan melakukan test case dengan menpartisi domain input dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Metode pengujian graph-based mengeksplorasi hubungan antara dan tingkah laku objek-objek program. Partisi ekivalensi membagi domain input ke dalam kelas data yang mungkin untuk melakukan fungsi perangkat lunak tertentu. Analisis nilai batas memeriksa kemampuan program untuk menangani data pada batas yang dapat diterima.

Metode pengujian yang terspesialisasi meliputi sejumlah luas kemampuan perangkat lunak dan area aplikasi. GUI, arsitektur client/server, dokumentasi dan fasilitas help dan sistem real time masing-masing membutuhkan pedoman dan teknik khusus untuk pengujian perangkat lunak.

2. White Box Testing

Adalah meramalkan cara kerja perangkat lunak secara rinci, karenanya logikal path (jalur logika) perangkat lunak akan ditest dengan menyediakan test case yang akan mengerjakan kumpulan kondisi dan atau pengulangan secara spesifik. Secara sekilas dapat

diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

Pengujian white-box berfokus pada struktur control program. Test case dilakukan untuk memastikan bahwa semua statemen pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji. Pengujian basic path, tehnik pengujian white-box, menggunakan grafik (matriks grafiks) untuk melakukan serangkaian pengujian yang independent secara linear yang akan memastikan cakupan.

Pengujian aliran data dan kondisi lebih lanjut menggunakan logika program dan pengujian loop menyempurnakan tehnik white-box yang lain dengan memberikan sebuah prosedur untuk menguji loop dari tingkat kompleksitas yang bervariasi. Pengujian black-box didesain untuk mengungkap kesalahan pada persyaratan fungsional tanpa mengabaikan kerja internal dari suatu program.

13.3 LATIHAN SOAL/TUGAS

1. Buatlah studi kasus pengembangan perangkat lunak, kemudian buatlah rencana pengujiannya!

13.4 DAFTAR PUSTAKA

1. Pressman, R. S. (2015). Software Engineering. A Practitioner's Approach (8th ed.). New York: McGraw-Hill Education.
2. Sommerville, I. (2011). Software Engineering (9th ed.). Boston: Addison-Wesley.