1. Jelaskan apa yang dimaksud dengan software.

Software secara umum dapat diartikan sebagai sekumpulan data-data elektronik yang tersimpan dan diatur oleh komputer yang berupa program atau instruksi untuk menjalankan dan mengeksekusi suatu perintah.

1. Apakah yang dimaksud dengan software engineering?

Software engineering adalah aplikasi rekayasa untuk pengembangan perangkat lunak dalam metode yang sistematis

1. Apa perbedaan antara software engineering dan web engineering?

Software engineering satu bidang yang mendalami cara-cara pengembangan perangkat lunak

Web engineering (rekayasa web) adalah suatu proses yang digunakan untuk menciptakan suatu sistem aplikasi berbasis web dengan menggunakan ilmu rekayasa

|  |  |  |
| --- | --- | --- |
|  | **Aplikasi Dekstop** | **Aplikasi Web** |
| **Grafis** | Tingkat grafis ditentukan oleh kebutuhan grafis aplikasi desktop tersebut. Butuh spec yang tinggi. | Grafis yang digunakan tidak begitu tinggi. |
| **Interaksi penggunaan** | Tidak semua aplikasi desktop dapat berjalan di semua Sistem Operasi. | Dapat berjalan disemua Sistem Operasi, yang penting ada web browser dan koneksi internet. |
| **Penggunaan jaringan** | Tidak bergantung pada internet (dapat berjalan secara offline) | Untuk menjalankannya dibutuhkan koneksi internet (tapi juga dapat berjalan secara offline) |
| **Akses** | Terbatas pada laptop atau PC | Dapat diakses dimana saja (mobile, tablet atau PC) asal terdapat web browsernya |
| **Fungsionalitas perbaikan** | Dapat dengan mudah memodifikasi settingannya | Tidak memerlukan lisensi ketika menggunakan Aplikasi Web, sebab lisensi itu telah menjadi tanggung jawab dari web penyedia aplikasi. |
| **Popularitas** | Aplikasi desktop banyak digunakan sebelum adanya smartphone, namun sekarang popularitasnya mulai meredup. |  |

1. Dalam proses pengembangan software, terdapat biaya yang harus dikeluarkan. Biaya apa sajakah itu?

* Biaya perencaaan, dokumentasi dan desain
* Biaya dukungan operasional
* Biaya pembuatan

1. Terdapat dua jenis produk software, yaitu Generic dan Customized. Jelaskan perbedaan dari keduanya.

**1. Perbedaan Tujuan**

Software generik atau software umum adalah jenis software yang dibuat untuk dipasarkan ke berbagai bidang penggunaan komputer. Maksudnya, software ini diharapkan dapat digunakan di bidang bisnis, arsitektur, pemasaran, kuliah, dan lain sebagainya. Sementara itu, software spesifik dibuat dengan memiliki tujuan khusus yang sangat spesifik dan hanya dipasarkan pada kalangan tertentu yang sangat terbatas.

**2. Perbedaan Fungsi**

Oleh karena ditujukan untuk dipasarkan pada banyak pasar, maka software generik memiliki fungsi-fungsi yang bersifat umum dan *applicable*di banyak tempat. Misalnya software untuk mengetik, memiliki fungsi untuk melakukan pembuatan dan editing dokumen dalam tingkat yang umum seperti misalnya menambah header atau footer. Sementara itu, software spesifik memiliki jenis-jenis fungsi yang sangat spesifik dan tidak dapat digunakan untuk menjalankan pekerjaan selain yang telah ditujukan dari pabrik. Misalnya software untuk melihat atau mengamati detak jantung. Fungsi software atau aplikasi ini terbatas dalam menerjemahkan detak jantung dan membuatnya ke dalam sebuah bentuk grafik yang dapat dimengerti oleh manusia atau dokter pada umumnya.

**3. Perbedaan Pengguna**

Software generik menyasar pengguna yang lebih banyak dan variatif. Contohnya adalah aplikasi editing dokumen seperti Microsoft Offfice atau LibreOffice, yang menyasar dari anak sekolah, pelaku bisnis, guru, akuntan, dan lain sebagainya. Oleh karena itu aplikasi ini menjadi salah satu aplikasi *wajib*bagi hampir semua orang yang perlu untuk membuat dokumen. Sementara itu, software spesifik lebih cenderung menyasar pengguna-pengguna khusus yang memiliki kebutuhan dan kemampuan khusus untuk mengaplikasikan software tersebut. Misalnya adalah software kasir di berbagai jenis pasar modern seperti *Indomaret*atau *Alfamart.*Aplikasi yang ada di kedua pasar tersebut bisa jadi telah dimodifikasi sedemikian rupa hingga hanya bisa digunakan atau dimengerti oleh kasir-kasir yang ada di dua pasar tersebut.

**4. Perbedaan Jumlah Pengguna**

Oleh karena software generik dibuat untuk memenuhi kebutuhan banyak orang atau pasar secara umum, maka jumlah pengguna pada aplikasi ini cenderung lebih banyak apabila dibandingkan dengan software spesifik. Hal ini wajar mengingat fungsi dari software generik sendiri diperlukan oleh banyak orang. Sementara itu, software spesifik tidak harus memiliki kuota atau syarat jumlah pengguna tertentu sehingga dia dikatakan sebagai software spesifik. Andaikan ada satu orang pengguna saja, aplikasi ini sudah dapat dikatakan sebagai aplikasi atau software spesifik karena sudah memenuhi syarat bahwa aplikasi tersebut dibuat untuk memenuhi kebutuhan pengguna tertentu yang sangat spesifik.

**5. Perbedaan Pembuat**

Software generik biasanya dibuat oleh perusahaan atau developer tertentu yang memiliki resoures atau sumber daya yang cukup besar. Hal itu karena dalam pemasaran software generik, selain dibutuhkan kemampuan coding yang cukup rumit dan juga waktu yang relatif dituntut untuk singkat, maka biaya yang dibutuhkan cukup besar. Akan tetapi bukan berarti mereka harus benar-benar menjalankan bisnis tertentu. Beberapa developer misalnya menggunakan donasi dari pengguna softwarenya untuk mengembangkan aplikasi yang ia buat.

Sementara itu, software spesifik boleh jadi tidak harus dibuat oleh orang yang memiliki sumber daya sangat besar. Semua orang, yang memiliki kemampuan untuk melakukan *programming*, hampir dapat dipastikan mampu atau dapat menjadi seorang pengembang software spesifik. Akan tetapi tidak menutup kemungkinan bahwa software spesifik yang ingin dibuat membutuhkan biaya yang cukup besar. Sebut saja software spesifik yang digunakan dalam dunia pertambangan.

1. Seperti apakah software yang dapat dikatakan sebagai software yang berkualitas

* **High Performance**

Software yang di buat mempunyai performance yang tinggi,walaupun digunakan oleh beberapa user

* **Mudah digunakan**

Software yang dibuat mempunyai sifat “***Easy to Use***” yakni mudah digunakan, sehingga tidak membutuhkan proses yang lama untuk seorang user dalam mempelajarinya

* **Penampilan yang baik**

Software yang dibuat harus memiliki ***Antar Muka/Interface***yang menarik,sehigga user tidak merasa jenuh

* **Reability**

Kehandalan sejauh mana suatu software dapat diharapkan untuk melakukan fungsinya sesuai dengan ketelitian yang diperlukan

* **Mampu beradaptasi**

Sejauh mana software yang dibuat mampu beradaptasi dengan perubahan –perubahan teknologi yang ada

* **Interopability**

Software yang dibuat haruslah mampu berinteraksi dengan aplikasi yang lain. Biasanya dapat dilihat dari adanya fasilitas Export/ Import data dari aplikasi yang lain

* **Mobility**

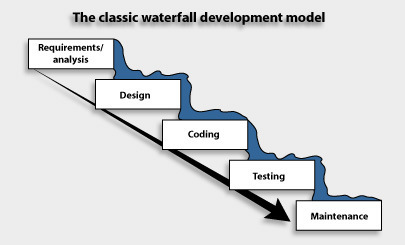
Software yang dibuat dapat berjalan pada bermacam-macam ***Sistem Operasi***

1. Apa saja tanggung jawab profesional dari seorang software engineer?

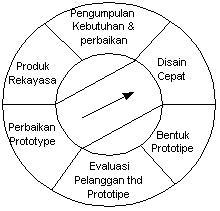
* Kerahasiaan
  + Para engineers biasanya harus menghormati kerahasiaan majikan mereka atau klien terlepas dari apakah perjanjian kerahasiaan formal telah atau tidak telah ditandatangani.
* Kompetensi
  + Engineers tidak boleh salah menggambarkan tingkat kompetensinya. Mereka harus tidak secara sadar menerima pekerjaan yang melebihi kompetensinya.
* Hak kekayaan intelektual
  + Engineers harus mengetahui hukum setempat yang mengatur penggunaan intelektual properti seperti paten, hak cipta, dll. Mereka harus berhati-hati untuk memastikan bahwa kekayaan intelektual pengusaha dan klien dilindungi.
* Penyalahgunaan komputer
  + Engineers perangkat lunak tidak boleh menggunakan keterampilan teknis mereka untuk menyalahgunakan orang lain komputer orang. Penyalahgunaan komputer berkisar dari yang relatif sepele (game bermain di mesin majikan, misalnya) hingga sangat serius (diseminasi virus).

1. Aktivitas apa sajakah yang diperlukan untuk mengembangkan sebuah software?

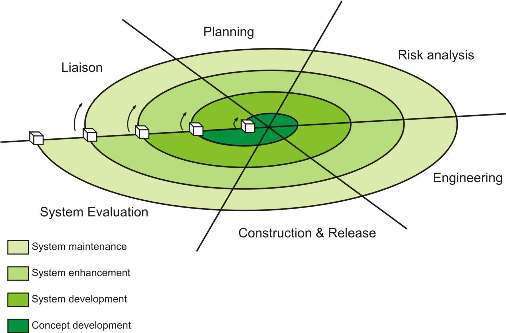
* Berdasarkan waterfall development



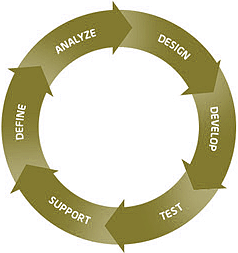
* + **Tahap analisis:** pada tahap ini berlangsung proses pengumpulan kebutuhan secara lengkap untuk dianalisis dan didefinisikan kebutuhan apa saja yang harus dipenuhi oleh program yang akan dibuat, seperti memahami domain permasalahan, tingkah laku, unjuk kerja dan interface (antar muka).
  + **Tahap desain:** proses ini melibatkan empat atribut sebuah program yaitu struktur data, arsitektur, perangkat lunak, representasi interface, dan detail (algoritma) prosedural.
  + **Tahap pengkodean:** proses penterjemahan desain ke dalam bentuk bahasa mesin yang dapat dilakukan secara mekanis.
  + **Tahap pengujian:** proses ini dikerjakan setelah kode dirancang dan difokuskan pada fungsi dan jumlah kesalahan untuk diperbaiki.
  + **Tahap pemeliharaan:** meliputi penyesuaian atau perubahan yang berkembang seiring dengan adaptasi perangkat lunak dengan kondisi atau situasi sebenarnya setelah disampaikan kepada konsumen atau pelanggan.
* Berdasarkan prototype development



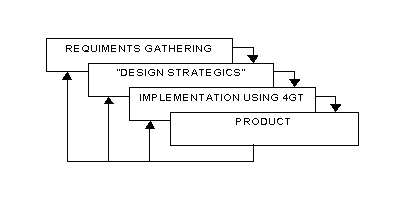
* + **Tahap Pengumpulan kebutuhan:** pada tahap ini, pelanggan dan pengembang saling bantu dalam mendefinisikan format seluruh perangkat lunak, menentukan keperluan dan garis besar sistem yang akan dirancang.
  + **Tahap Quick design:** membangun rancangan global sebagai contoh bagi user
  + **Tahap Pembangunan Prototipe:** proses perancangan sementara yang fokusnya kepada penyajian kepada pelanggan, termasuk pengujian dan penyempurnaan.
  + **Tahap Evaluasi Pelanggan:** di mana pelanggan melakukan pengujian terhadap prototipe yang ada dan pengembang memperhalus analisis kebutuhan pemakai.
  + **Tahap Pembuatan dan Implementasi:** tahap ini termasuk proses desain (rancang), pengkodean dan testing.
* **Berdasarkan spiral modeling**



* + **Tahap Liason:** pada tahap ini dibangun komunikasi yang baik dengan calon pengguna/pemakai
  + **Tahap Planning (perencanaan):** pada tahap ini ditentukan sumber-sumber informasi, batas waktu dan informasi-informasi yang dapat menjelaskan proyek.
  + **Tahap Analisis Resiko:** mendefinisikan resiko, menentukan apa saja yang menjadi resiko baik teknis maupun manajemen.
  + **Tahap Rekayasa (engineering):** pembuatan prototipe
  + **Tahap Konstruksi dan Pelepasan (release):** pada tahap ini dilakukan pembangunan perangkat lunak yang dimaksud, diuji, diinstal dan diberikan sokongan-sokongan tambahan untuk keberhasilan proyek.
  + **Tahap Evaluasi:** Pelanggan/pemakai/pengguna biasanya memberikan masukan berdasarkan hasil yang didapat dari tahap engineering dan instalasi.
* Berdasarkan RAD Model



* + **Tahap Pemodelan Bisnis:** dibuat agar dapat menjawab pertanyaan-pertanyaan berikut: informasi apa yang mengontrol proses bisnis? Informasi apa yang didapat? Siapa yang mendapatkannya? Untuk siapa informasi itu ditujukan? Siapa yang akan memprosesnya?
  + **Tahap Pemodelan Data:** informasi-informasi yang dipadu dari pemodelan bisnis dipilah-pilah ke menjadi sekumpulan objek data yang masing-masing objek diidentifikasikan dan ditentukan hubungan antara objek-objek tersebut.
  + **Tahap Pemodelan Proses:** aliran informasi yang didapat dalam proses pemodelan data diolah sedemikian untuk dapat menopang fungsi-fungsi bisnis. Prosesnya dikreasikan untuk menambah, memodifikasi, menghapus dan atau mendapatkan kembali sebuah objek data.
  + **Tahap Pembuatan Aplikasi:** RAD dapat saja memakai kembali komponen program yang sudah ada bila dimungkinkan, atau membuat komponen yang dapat digunakan lagi bila diperlukan di masa mendatang. RAD juga diasumsikan menggunakan teknik generasi keempat (4GT).
  + **Tahap Pengujian dan Pergantian:** Proses RAD menekankan pada pemakaian kembali yang memungkinkan berkurangnya keseluruhan waktu pengujian, namun komponen harus diuji dan harus dilatih secara penuh dan terintegrasi.
  + **Model RAD sangat tepat diterapkan untuk** sistem yang telah jelas dan lengkap kebutuhannya, di mana terdapat komponen-komponen yang dapat dipakai kembali dalam proyek yang berskala kecil dengan waktu pengembangan perangkat lunak yang singkat.
* **Berdasarkan Teknik 4GT**



* + **Tahap Pengumpulan Kebutuhan:** tahap ini dimulai dengan mengumpulkan serangkaian kebutuhan yang nantinya akan diterjemahkan ke dalam prototipe. Namun, apabila pelanggan tidak yakin dengan apa yang diperlukan dan fakta-fakta tidak jelas diketahui maka prototipe tidak dapat dikerjakan oleh peralatan 4GT.
  + **Tahap Merancang Strategi:** tahap ini dibutuhkan untuk proyek besar yakni dengan menterjemahkan kebutuhan menjadi prototipe operasional  agar tidak timbul masalah yang sama jika dibuat dengan model konvensional. Namun, untuk proyek skala kecil tahap ini dapat dihilangkan dengan  langsung melakukan implementasi dengan menggunakan bahasa generasi keempat (4GT).
  + **Tahap Implementasi Menggunakan Bahasa Keempat:** untuk skala kecil tahap ini dapat langsung dilakukan ketika kebutuhan telah jelas, dan untuk proyek besar tahapan ini dijalankan setelah dirancang prototipe operasional. Implementasi yang menggunakan 4GT memudahkan pengembang software untuk menjelaskan hasil yang diharapkan yang nantinya akan diterjemahkan ke dalam bentuk kode sumber dan kode objek.
  + **Tahap Produksi:** Tahap ini merupakan langkah terakhir yakni mengubah implementasi  4GT ke dalam hasil akhir berupa produk.

1. Jelaskan apa yang dimaksud dengan software validation.

Validation mengacu kepada sekumpulan aktifitas yang berbeda yang memastikan bahwa software telah dikembangkan dapat memenuhi ekspektasi/harapan pengguna sistem. Melibatkan checking dan review proses serta ujicoba sistem

1. Jelaskan apa yang dimaksud dengan Requirements Engineering.

Requirements engineering adalah fase terdepan dari proses rekayasa perangkat lunak (software engineering), dimana software requirements (kebutuhan) dari user (pengguna) dan customer (pelanggan) dikumpulkan, dipahami dan ditetapkan

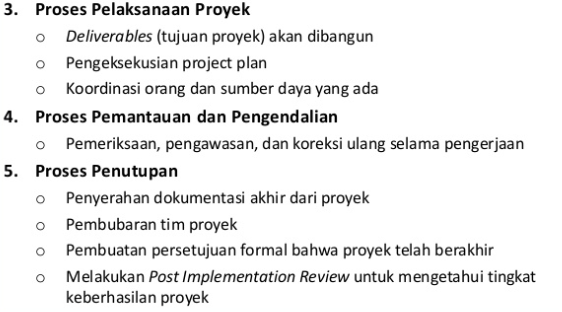
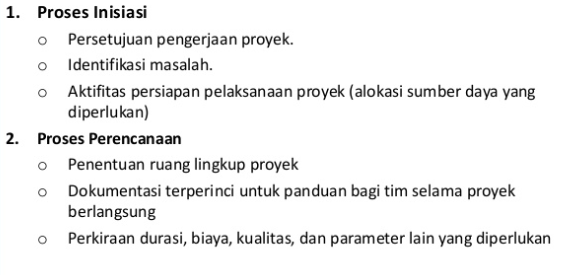
1. Jelaskan apa yang dimaksud dengan Software Project Management?

Software Project Management dalah software yang dibuat untuk mengatur manajemen tugas, waktu pengerjaan dan kerjasama dalam sebuah tim. Maka dari itu Software Manajement Project sangat cocok untuk suatu lingkungan bisnis yang menuntut kemampuan akuntansi, fleksibilitas, inovasi, kecepatan dan perbaikan yang berkelanjutan.

1. Apa saja kriteria dari susksesnya sebuah Proyek Software?

* Menentukan definisi tujuan (goal definition) yang jelas, maksudnya seberapa besar proyek yang akan dilaksanakan serta kebutuhan apa yang diperlukan oleh semua orang yang terlibat dalam pembuatan proyek.
* Hasil dari proyek tersebut dapat diterima oleh pelanggan, deadline yang tepat, serta sesuai anggaran atau tidak melebihi budget.
* Komitmen yang kuat pada suatu proyek, maksudnya proyek yang berhasil adalah proyek yang dapat memiliki komitmen dalam hal manajemen dan organisasi dalam sebuah proyek. Sesuai yang direncanakan maksudnya tidak mengambil jalan pintas dalam sebuah proyek. Terlihat dari harapan-harapan yang membangun di sebuah tim yang menangani proyek.
* Cakupan (Scope) proyek yang digarap sewajarnya, biasanya proyek yang berhasill memliki cakupan (scope) yang jelas, tidak serakah dan hasilnya pun sempurna.
* Biaya yang dikeluarkan ketika proyrk terselesaikan tidak jauh dari rencana awal, maksudnya jangan sampai biaya yang dikeluarkan sudah besar, akan tetapi kualitas dari hasil sebuah proyek mengecewakan. Atau biaya yang dikeluarkan sudah banyak hasil proyeknya telat waktu.
* Kualitas yang baik, maksudnya ketika dilakukan proses pengujian hasil proyek sesuai dengan apa yang diharapkan. Jangan sampai hasil dari sebuah proyek cepat, tapi kualitasnya dikorbankan.
* Ketrampilan sumber daya manusia, maksudnya diperlukan SDM yang mempunyai kompetensi yang unggul atau ahli didalam bidangnya. SDM yang mempunyai jiwa disiplin tepat waktu, dapat membuat lingkungan kerja yang kondusif, serta pekerja yang dapat diatur oleh manajer.
* Komunikasi yng baik, maksudnya begini ketika tim proyek menjalankan sebuah proyek ada baiknya menjalin hubungan secara terus menerus kepada pemilik dan pengguna. Dan tidak menutup kemungkinan proyek yang berhasil adalah tim yang dapat menjalin komunikasi sesama tim
* Resiko yang ditimbulkandari sebuah proyek kecil, sebisa mungkin proyek yang dijalankan tidak menimbulkan resiko. Diharapkan seminimal mungkin resiko terjadi dalam sebuah proyek.
* Yang terakhir hasil dari sebuah proyek diharapkan tidak menimbulkan suatu permasalahan baru diperusahaan dalam artikata malah menyulitkan perusahaan dalam menjalankan kegiatan operasional perusahaan.

1. Apa saja aktivitas dari Software Project Management?



1. Apa yang dimaksud dengan Manajemen Resiko?

Manajemen risiko adalah suatu pendekatan terstruktur/metodologi dalam mengelola ketidakpastian yang berkaitan dengan ancaman; suatu rangkaian aktivitas manusia termasuk: Penilaian risiko, pengembangan strategi untuk mengelolanya dan mitigasi risiko dengan menggunakan pemberdayaan/pengelolaan sumberdaya.

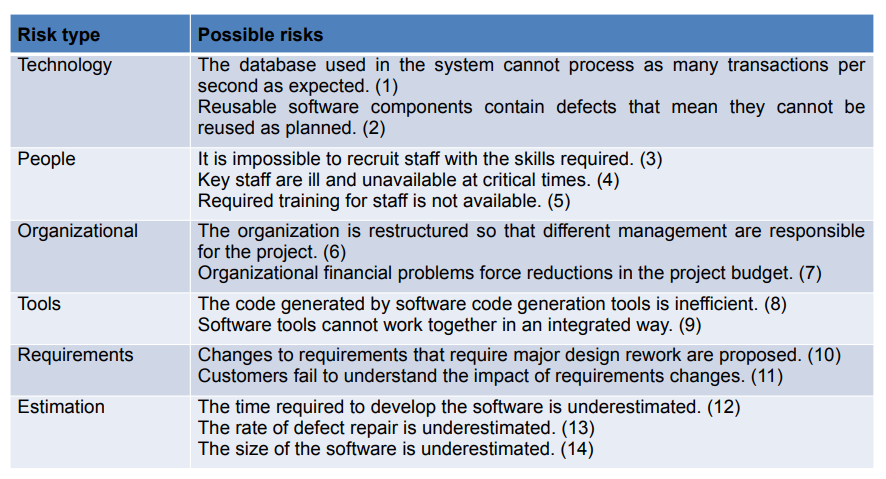
1. Apa saja proses dalam Manajemen Resiko?

* Komunikasi dan konsultasi
* Pertimbangan konteks
* Identifikasi resiko
* Penanganan resiko
* Monitoring dan review
* Pencatatan proses menajemen resiko

------------------

* Risk identification
  + Identify project, product and business risks;
* Risk analysis
  + Assess the likelihood and consequences of these risks;
* Risk planning
  + Draw up plans to avoid or minimise the effects of the risk;
* Risk monitoring
  + Monitor the risks throughout the project;

1. Apa saja jenis resiko yang ada dalam Proyek Software dan berikan contoh resiko dari masing- masing jenis.



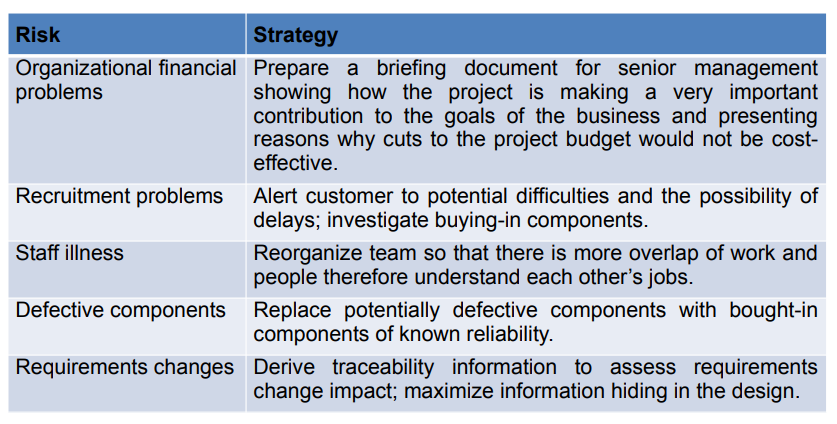
1. Apa yang dimaksud dengan Analisis Resiko?

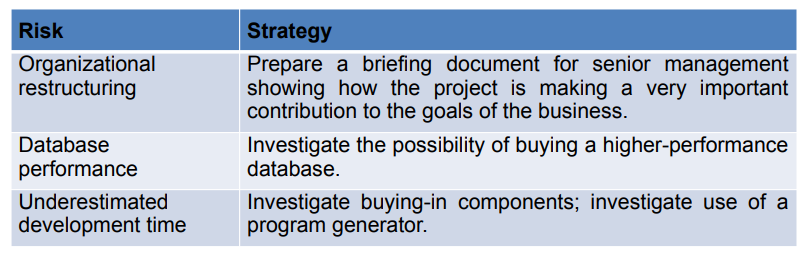
* Nilai probabilitas dan keseriusan setiap risiko.
* Probabilitas mungkin sangat rendah, rendah, sedang, tinggi atau sangat tinggi.
* Konsekuensi risiko dapat menjadi bencana, serius, ditoleransi atau tidak signifikan.

1. Bagaimana caranya melakukan Perencanaan Resiko?

* Pertimbangkan setiap risiko dan kembangkan strategi untuk mengelola risiko itu.
  + Strategi penghindaran
    - Probabilitas bahwa risiko akan muncul berkurang;
  + Strategies Strategi minimalisasi
    - Dampak risiko pada proyek atau produk akan berkurang;
  + Rencana darurat
    - Jika risiko muncul, rencana darurat adalah rencana untuk menghadapinya risiko;

1. Berikan contoh resiko yang mungkin terjadi dalam proyek software dan bagaimana menanganinya?

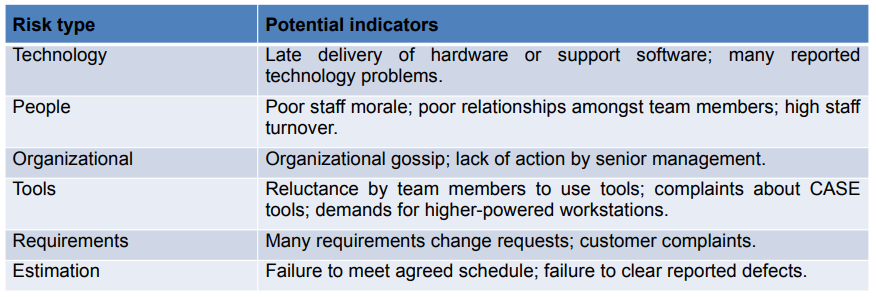




1. Apa yang dimaksud dengan monitoring resiko?

* Menilai setiap risiko yang teridentifikasi secara berkala untuk memutuskan apakah atau tidak itu menjadi kurang atau lebih mungkin.
* Juga menilai apakah dampak dari risiko tersebut berubah.
* Setiap risiko utama harus didiskusikan di manajemen rapat kemajuan

1. Pada soal 16 disebutkan jenis-jenis Resiko, sebutkan contoh indikator terjadinya resiko tersebut.



1. Apa peranan manajemen SDM dalam Manajemen Resiko?

* Konsistensi
  + Semua anggota tim harus diperlakukan dengan cara yang sebanding tanpa favorit atau diskriminasi.
* Hormat
  + Anggota tim yang berbeda memiliki keterampilan yang berbeda dan ini perbedaan harus dihormati.
* Inklusi
  + Libatkan semua anggota tim dan pastikan bahwa orang-orang melihat dipertimbangkan.
* Kejujuran
  + Anda harus selalu jujur ​​tentang apa yang berjalan baik dan apa berjalan buruk dalam suatu proyek.

1. Apa yang dimaksud dengan manajemen perubahan (Change Management)?

Manajemen perubahan dimaksudkan untuk memastikan sistem itu evolusi adalah proses yang dikelola dan prioritas itu diberikan untuk perubahan yang paling mendesak dan hemat biaya.

1. Kenapa diperlukan Manajemen perubahan dalam proyek software?

Kebutuhan dan persyaratan organisasi berubah selama seumur hidup suatu sistem, bug harus diperbaiki dan sistem harus beradaptasi dengan perubahan di lingkungan mereka.

1. Apa yang dimaksud dengan Version Management?

Manajemen versi (VM) adalah proses menjaga melacak berbagai versi komponen perangkat lunak atau item konfigurasi dan sistem di mana ini komponen digunakan.

1. Apa kaitan antara Version Management dan Change Management?
2. Apa yang dimaksud dengan Release Management?
3. Jelaskan faktor-faktor apa sajakah yang mempengaruhi System Release Planning?
4. Jelaskan apa perbedaan antara kebutuhan fungsional dan non-fungsional.
5. Jelaskan dan berikan contoh perbedaan antara kebutuhan pengguna (user requirements) dan kebutuhan sistem (system requirements)

* Kebutuhan pengguna(user requirements) dimaksudkan untuk menggambarkan fungsi dan fitur sistem dari perspektif pengguna agar sistem sesuai dengan keinginan pengguna. Pengembang software yang terlibat dalam proses harus mampu memahami perspektif/sudut pandang pengguna.
* Kebutuhan sistem(system requirements)ini merupakan hal yang lebih rinci lagi daripada kebutuhan pengguna dan juga dapat dimasukan ke dalam kontrak sistem. Kebutuhan sistem dikembangkan setelah kebutuhan pengguna telah ditetapkan dimana semua fitur dan kemampuan sistem dijelaskan.

1. Jelaskan apa yang dimaksud dengan System Modeling.

Pemodelan Sistem adalah suatu bentuk penyederhanaan dari sebuah elemen dan komponen yang sangat komplek untuk memudahkan pemahaman dari informasi yang dibutuhkan.

1. Apa itu Unified Modeling Language (UML)?

UML merupakan singkatan dari Unified Modeling Language. UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

1. Jelaskan apa yang dimaksud dengan Software Architecture.

Adalah proses yg mendefinisikan solusi yg terstruktur yang memenuhi kebutuhan teknis dan operasional, disisi lain mengoptimasi quality dari sebuah aplikasi yg meliputi: performance, security, dan manageability.

1. Berikan contoh dari Software Architectural Conflicts.
2. Apa yang dimaksud dengan Architecture Reuse
3. Apa yang dimaksud dengan Architectural Models.
4. Jelaskan apa itu Object Oriented Development?

Object Oriented Development adalah suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan abstraksi objek-objek yang ada di dunia nyata

1. Apa saja karakteristik dari Object Oeriented Development?
2. Apa fungsi dari Use-Case Model?

Use case diagram menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara sederhana, diagram use case digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.

1. Apa yang dimaksud dengan Rapid Software Development?

Rapid application development (RAD) atau rapid prototyping adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik incremental (bertingkat). RAD menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini.

1. Apa karakteristik dari Rapid Software Development?
2. Jelaskan manfaat prototyping.

* Menghasilkan syarat yang lebih baik dari produksi yang dihasilkan oleh metode ‘spesifikasi tulisan’.
* User dapat mempertimbangkan sedikit perubahan selama masih bentuk prototipe.
* Memberikan hasil yang lebih akurat dari pada perkiraan sebelumnya, karena fungsi yang diinginkan dan kerumitannya sudah dapat diketahui dengan baik.
* User merasa puas. Pertama, user dapat mengenal melalui komputer. Dengan melakukan prototipe (dengan analisis yang sudah ada), user belajar mengenai komputer dan aplikasi yang akan dibuatkan untuknya. Kedua, user terlibat langsung dari awal dan memotivasi semangat untuk mendukung analisis selama proyek berlangsung.

1. Apa itu extreme programming?

Extreme Programming adalah suatu model yang termasuk dalam pendekatan agile yang diperkenalkan oleh Kent Back. Menurut penjelasannya, definisi XP adalah sebagai berikut: “Extreme Programming (XP) adalah metode pengembangan software yang cepat, efisien, beresiko rendah, fleksibel, terprediksi, scientific, dan menyenangkan.“. Model ini cenderung menggunakan pendekatan Object-Oriented. Tahapan-tahapan yang harus dilalui antara lain: Planning, Design, Coding, dan Testing.

1. Jelaskan apa itu Framework MVC (Model-View-Controller).

Model-View-Controller atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller)

1. Jelaskan apa manfaat penggunakan Framework MVC.

* Mempercepat proses pembuatan aplikasi baik itu aplikasi berbasis desktop, mobile ataupun web.
* Membantu para developer dalam perencanaan, pembuatan dan pemeliharaan sebuah aplikasi.
* Aplikasi yang dihasilkan menjadi lebih stabil dan handal, hal ini dikarenakan Framework sudah melalui proses uji baik itu stabilitas dan juga kehandalannya.
* Memudahkan para developer dalam membaca code program dan lebih mudah dalam mencari bugs.
* Memiliki tingkat keamanan yang lebih, hal ini dikarenakan Framework telah mengantisipasi cela - cela keamanan yang mungkin timbul.
* Mempermudah developer dalam mendokumentasikan aplikasi - aplikasi yang sedang dibangun.

1. Jelaskan apa perbedaan antara Component Testing dan System Testing pada software testing process.
2. Sebutkan dan jelaskan 3 (tiga) jenis software testing yang Anda ketahui.

**1. Performance Testing**

Performance test adalah integration dan usability test yang menentukan apakah system or subsystem dapat memenuhi kriteria kinerja berbasis waktu seperti response time atau throughput. Response time menentukan batas waktu maksimum yang diijinkan dari respon software untuk query dan update. Throughput menentukan jumlah minimum query dan transaksi yang harus diproses per menit atau per jam.

**2. System Testing**

System test adalah integration test dari behavior seluruh sistem atau independent subsystem. System testing biasanya dilakukan pertama kali oleh pengembang atau personil pengujian untuk memastikan bahwa keseluruhan sistem tidak berfungsi dan bahwa sistem telah memenuhi persyaratan pengguna (user requirement). System testing biasanya dilakukan di akhir setiap iterasi untuk mengidentifikasi isu -- isu penting, seperti masalah performance yang perlu ditangani di iterasi berikutnya. Biasanya test ini harus dilakukan sesering mungkin.

**3. Unit Testing**

Unit testing adalah proses metode pengujian individual, class, atau komponen sebelum mereka terintegrasi dengan perangkat lunak lainnya. Tujuan dari unit testing adalah untuk mengidentifikasi dan memperbaiki kesalahan sebanyak mungkin sebelum modul -- modul digabungkan menjadi unit perangkat lunak yang lebih besar, seperti program, class dan subsistem. Kesalahan menjadi lebih sulit dan mahal untuk ditemukan dan diperbaiki ketika banyak unit telah digabungkan.

Unit testing memerlukan implementasi dari driver dan/atau stub. Stub adalah class atau method dummy yang dapat dipanggil namun biasanya tidak melakukan apapun kecuali mengembalikan tipe yang diperlukan. Modul driver adalah program yang menjalankan method atau fungsi dari class yang akan ditest.

*Berikut adalah langkah yang harus dilakukan :*

* + Menentukan nilai dari parameter input
  + Memanggil unit yang dites, melewatkannya dengan parameter input
  + Menerima parameter kembalian dari unit yang dites dan mencetaknya, menampilkannya, atau mengetes hasilnya terhadap hasil yang diharapkan.

**4. Integration Testing**

Integration test adalah mengevaluasi behavior dari kelompok method atau class. Tujuan dari integration test adalah untuk mengidentifikasi kesalahan yang tidak dapat dideteksi oleh unit testing. Kesalahan tersebut mungkin disebabkan oleh beberapa masalah, diantaranya :

Interface incompatibility, misalnya sebuah method melewatkan parameter dengan tipe data yang salah ke method lainnya

Parameter values, misalnya sebuah method mengembalikan nilai yang tidak terduga seperti nomor negatif untuk harga

Run-time exceptions, misalnya method menyebabkan kesalahan seperti out of memory atau file already in use karena ada konflik kebutuhan sumber daya

Unexpected state interactions, misalnya state dari dua atau lebih objek yang berinteraksi menyebabkan kesalahan yang kompleks seperti ketika method class Order menjalankan satu kesalahan dari semua kemungkinan state objek Customer.

Beberapa masalah di atas merupakan kesalahan paling umum yang sering ditemui dalam integration testing, tetapi sebenarnya masih banyak masalah lainnya yang dapat menjadi penyebab kesalahan (error).

**5. Usability Testing**

Usability test adalah test untuk menentukan apakah method, class, subsistem, atau sistem telah memenuhi persyaratan pengguna. Oleh karena banyaknya tipe persyaratan sistem baik yang fungsional maupun non-fungsional, maka banyak tipe dari usability test yang harus dilakukan di waktu yang berbeda. Umumnya usability test mengevaluasi persyaratan fungsional dan kualitas dari user interface.

User berinteraksi dengan sistem untuk menentukan apakah fungsi telah seperti yang diharapkan dan apakah user interface membuat sistem dapat mudah digunakan. Pengujian ini sering dilakukan untuk mendapatkan feedback yang cepat dalam meningkatkan interface dan mengkoreksi kesalahan dalam komponen perangkat lunak.

**6. Smoke Testing**

Smoke testing adalah system test yang biasanya dilakukan setiap hari atau beberapa kali per minggu. Build and smoke test sangat penting karena menyediakan feedback yang cepat dalam masalah yang signifikan.

**7. Stress Testing**

Stress Testing adalah pengujian yang biasanya dilakukan dalam membuat sebuah website, dimana stress testing dilakukan untuk mengetahui sekuat apa server website kita menampung visitor dalam website tersebut, dengan cara melakukan hit dummy ke website menggunakan tools.

**8. User Acceptence Test (UAT)**

User acceptance test digunakan untuk menentukan apakah sistem yang dikembangkan telah memenuhi kebutuhan pengguna. Dalam beberapa proyek, acceptance testing dilakukan pada putaran terakhir proses pengujian yaitu sebelum sistem diserahkan kepada user.

Acceptance Testing biasanya dilakukan setelah rangkaian testing seperti Unit Testing, Integration Testing, dan System Testing selesai dan menggunakan metode Black Box Testing, dengan mengunakan dokumen test case untuk di presentasikan di akhir ke user / client.

1. Apa yang dimaksud dengan Security Requirments?

Security Requirments adalah fitur keamanan yang diperlukan oleh pengguna sistem atau kualitas yang harus dimiliki sistem untuk meningkatkan kepercayaan pengguna pada sistem yang mereka gunakan. Secara umum, persyaratan keamanan dianggap sebagai persyaratan non-fungsional

1. Berikan contoh Security Requirements yang ada pada aplikasi proyek akhir kelompok Anda.
2. Sebutkan dan Jelaskan tahapan-tahapan analisis Survivability.



* Pertahanan / Resistance
  + Menghindari masalah dengan membangun kemampuan ke dalam sistem tahan serangan
* Recognition / Penemuan
  + Mendeteksi masalah dengan membangun kemampuan ke dalam sistem mendeteksi serangan dan kegagalan dan menilai kerusakan yang terjadi
* Pemulihan / Recovery
  + Menoleransi masalah dengan membangun kemampuan ke dalam sistem untuk memberikan layanan saat diserang

\*\*\* Selamat Mengerjakan \*\*\*