**Ringkasan UTS Software Engineering**

1. Software 🡪 sekumpulan perintah / fungsi yang disatukan menjadi program yang akan dijalankan oleh komputer. Terdiri dari 3 hal :

* Instruction 🡪 yang memiliki fitur, function, performance.
* Data structures 🡪 membuat program untuk memanipulasi informasi.
* Dokumentasi 🡪 menjelaskan operasi dan kegunaan program.

1. Komponen yang diperhatikan pada software :

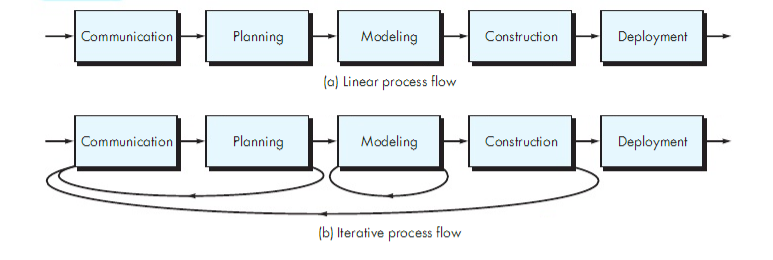
* Waktu
* Biaya
* Kinerja

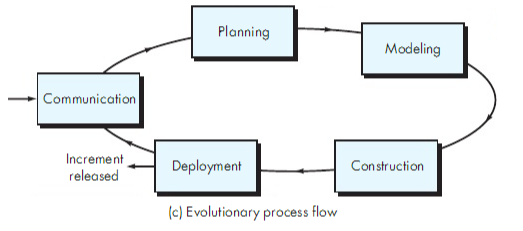
1. Software engineering 🡪 ilmu yang membahas produksi sebuah software mulai dari requirements, design, coding, hingga maintenance.
2. Software process 🡪 tahapan membangun sebuah software :

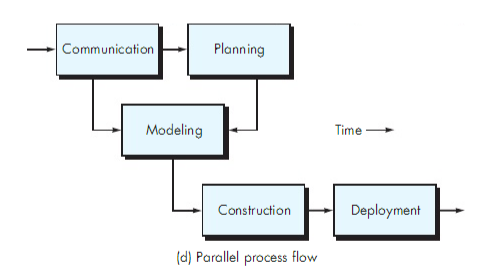
* Spesifikasi 🡪 apa yang harus dilakukan (requirements, analisis).
* Development 🡪 proses pembuatan / develop software (design, coding).
* Validasi 🡪 mengecek software.
* Evolusi 🡪 maintain software.

1. Process flow :

* Linear
* Iterative
* Evolutionary
* Parallel







1. Process pattern :

* Stage pattern 🡪 mendefinisikan masalah terkait aktivitas framework untuk proses.
* Task pattern 🡪 mendefinisikan masalah terkait aktivitas software engineering agar sukses.
* Phase pattern 🡪 menentukan urutan aktivitas framework yang terjadi dalam proses.

1. Pattern analysis :

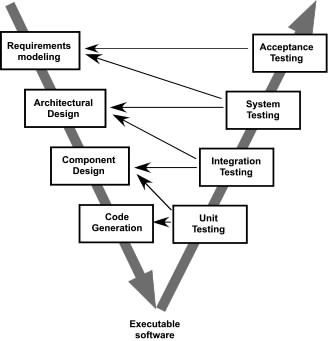
* Nama pattern 🡪 judul yang mewakili pattern.
* Intent 🡪 menjelaskan apa yang pattern selesaikan / kerjakan.
* Motivation 🡪 skenario yang menjelaskan bagaimana pattern digunakan.
* Konteks 🡪 penjelasan isu external yang berefek pada pattern.
* Solusi 🡪 penjelasan aplikasi / cara kerja pattern.
* Konsekuensi 🡪 apa isu yang terjadi ketika suatu pattern diaplikasikan.
* Design 🡪 menganalisa design pattern (penggunaannya) agar tercapai tujuan.
* Penggunaan 🡪 implementasi pada sistem.
* Pattern berkaitan 🡪 variasi / hubungan dengan pattern lain.

1. Jenis – jenis process model / metode pengembangan software :

* Waterfall



* V – Model



* Incremental model
* Evolutionary model
* Prototyping



* Spiral



* Concurrent



1. Pengeluaran biaya pada process model :

* Waterfall 🡪 biaya lebih banyak di integrasi / testing.
* Iterative development 🡪 biaya lebih banyak di development.
* Component – Based 🡪 biaya lebih banyak di integrasi / testing.

1. Specialized process model :

* Component – Based Development 🡪 proses untuk apply ketika development objective nya adalah reuse.
* Formal Methods 🡪 menekankan requirement spesifikasi secara matematis.
* AOSD 🡪 menyediakan pendekatan proses dan metodologi untuk defining, specifying, designing, constructing aspects.
* Unified Process 🡪 software process yang melibatkan UML.

1. Sifat – sifat agile :

* Efektif (respon cepat) terhadap perubahan.
* Komunikasi efektif pada stakeholder.
* Penggambaran customer ke team.
* Organisir team sehingga bekerja dalam kontrol.
* Rapid, incremental delivey.

1. Agile development :

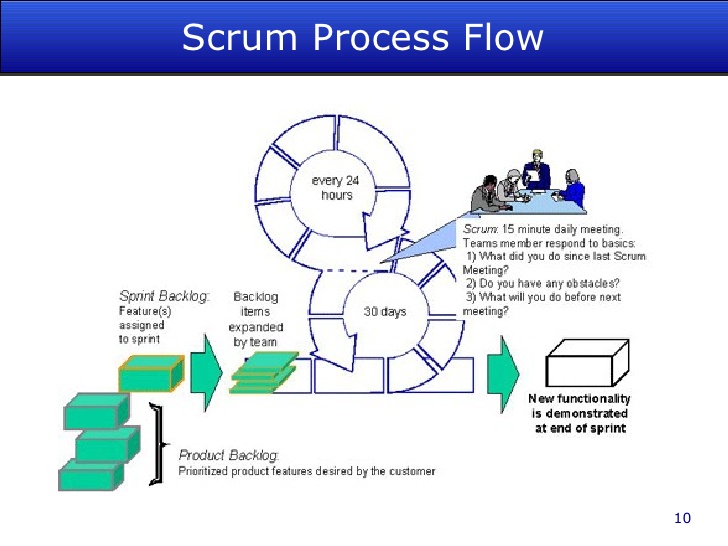
* XP (Xtreme Programming)
* Planning 🡪 perencanaan dari user stories.
* Design 🡪 prinsip KIS, CRC cards, refactoring.
* Coding 🡪 membentuk unit test sebelum coding, pair programming.
* Testing 🡪 dilakukan setiap hari, acceptance test dilakukan user
* Adaptive Software Development



* Dynamic System Development



* Scrum



* Crystal 🡪 memiliki manuverability berdasarkan karakteristik masalah.
* Feature Driven Development 🡪 menekankan pada mendifinisikan feature.



1. CASE (Computer Aided Software Engineering) 🡪 tools untuk support / mendukung dalam mendevelop software :

* Upper – CASE 🡪 yang dilakukan sebelum early process (requirements, design).
* Lower – CASE 🡪 yang dilakukan saat sudah masuk programming, debugging, testing.

1. Atribut software yang baik :

* Maintability 🡪 bisa di maintain / di update.
* Dependability 🡪 bukan bajakan / harus trustworthy, mengikuti perubahan zaman.
* Efisiensi 🡪 hemat sumber daya sistem.
* Acceptability 🡪 dapat diterima oleh user.

1. Tantangan dalam dunia software engineering :

* Heterogenity 🡪 dapat dijalankan di berbagai platform.
* Delivery 🡪 selesai tepat waktu.
* Trust 🡪 saat testing, dapat dipercaya oleh user.

1. Isu tanggung jawab software engineer :

* Confidentiality 🡪 tidak menyebar source code.
* Competence 🡪 kompeten dengan software engineer yang lain.
* Intelektual property rights 🡪 ada hak paten yang tidak boleh digunakan sembarangan.
* Penyalahgunaan komputer 🡪 tidak boleh menyalahgunakan skill computer engineer lain.

1. Software application domain :

* System software 🡪 contoh : windows, sistem operasi.
* Application software 🡪 contoh : browser.
* Engineering / scientific software 🡪 contoh : calculator.
* Embedded software 🡪 contoh : BIOS.
* Product – line software
* Web / mobile application 🡪 contoh : browser, application software.
* Artificial Intelligence software 🡪 contoh : Traffic Light Management Control.

1. 4 kategori software di bidang industri :

* Web Apps
* Mobile Application
* Cloud Computing
* Product – Line software

1. Layered technology :

* Tools
* Methods
* Process model
* Quality focus

1. Software engineering practice :

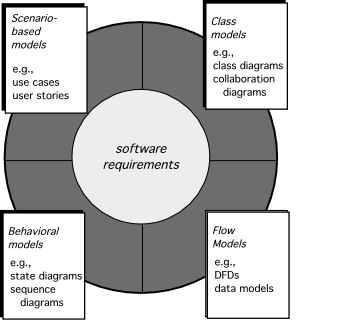
* Memahami masalah
* Merencanakan solusi
* Lakukan rencana
* Uji hasil untuk akurasi

1. Requirement engineering terdiri dari :

* Inception 🡪 komunikasi awal user dan developer. Terdapat hal – hal : memahami masalah, orang yang ingin solusi, lingkungan solusi, efektifitas komunikasi dan kolaborasi customer – developer.
* Elicitation 🡪 meminta requirement dari stakeholder.
* Elaboration 🡪 mengembangkan requirement dengan analysis model yang mengidentifikasi data, function, behavioral requirements.
* Negotiation 🡪 persetujuan user dan developer tentang sistem. Developer harus mampu menciptakan solusi.
* Specification 🡪 spesifikasi yang dibutuhkan oleh user dan membuat lebih spesifik requirement yang didapat. Bisa berupa : dokumen tertulis, kumpulan model, formal mathematical, kumpulan scenario (use case), prototype.
* Validation 🡪 validasi error, kalrifikasi, missing information, inconsistency, melihat apakah sudah sesuai requirement atau masih ada yang konflik.
* Requirement management 🡪 setelah selesai semua tahap, didokumentasikan.

1. Element analysis model (Software requirement modelling):

* Scenario – Based elements :
* Functional 🡪 berupa narasi / user stories.
* Use Case 🡪 menggambarkan interaksi aktor dengan sistem.
* Class – Based elements 🡪 class diagram dan CRC.
* Behavior elements 🡪 state diagram (kondisi objek), sequence diagram.
* Flow – oriented elements 🡪 data flow diagram (diagram yang menggambarkan aliran data pada suatu sistem yang terstruktur).

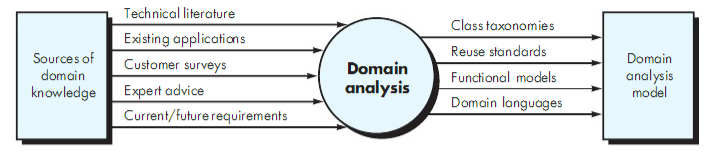


1. Analysis model pada scenario based :

* A Bridge



* Domain analysis



1. Langkah pada scenario based :

* Membuat use case awal
* Refining / menyaring use case
* Membuat formal use case

1. Class – based 🡪 merepresentasikan objek / entitas yang dibutuhkan. Ciri – cirinya :

* Ada atribut
* Memegang nilai sifat / keadaan class
* Ada 4 operasi :
* Memanipulasi data
* Menanyakan tentang
* Melakukan perhitungan
* Memantau objek

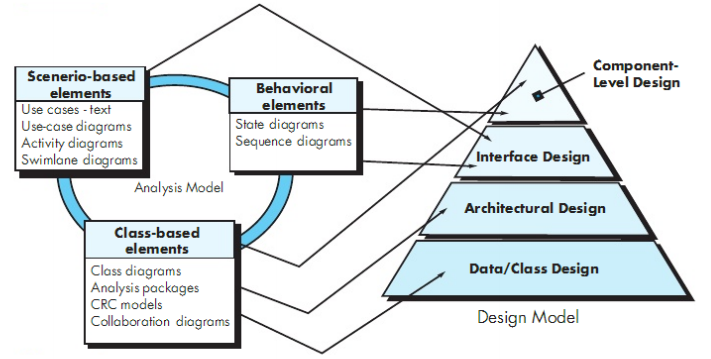
1. Behavioral element 🡪 indikasi bagaimana software respon ke external event.

* Mengevaluasi use case untuk memahami urutan interaksi.
* Identifikasi event pada interaksi.
* Buat sequence setiap use case
* Buat state diagram sistem
* Review untuk akurasi dan konsistensi

1. Desain software yang baik :

* Firmness 🡪 tidak ada bug.
* Comodity 🡪 sesuai dengan tujuannya.
* Delight 🡪 experience penggunaan program harus baik.

1. Design concept 🡪 mentranslasikan requirement model ke design model.



1. Design model ada 4 :

* Component – Level Design 🡪 dari class based dan behavioral.
* Interface design 🡪 dari scenario dan behavioral.
* Architectural design 🡪 dari class based.
* Data / class design 🡪 dari class based.

1. Kualitas dan desain :

* Harus di implement di semua requirement.
* Harus bisa dipelajari
* Menyediakan gambaran lengkap software, addressing data, functional, dan behavioral design.

1. Quality guideliness 🡪 memperkirakan seperti pa yang client inginkan (melihat / menempatkan diri dari sisi client).
2. Architectural design :

* Analisis efektifitas desain
* Mempertimbangkan alternatif arsitektural ketika ada perubahan
* Mengurangi resiko yang berkaitan dengan pembuatan software

1. Cara analisis architectural design :

* Kumpul skenario
* Kumpul requirement, constraint, environment description
* Deskripsikan pola arsitektural
* Evaluasi kualitas atribut
* Identifikasi sensitivitas kualitas atribut
* Kritik candidate architecture menggunakan analisis sensitivitas

1. Component – level design :

* Component 🡪 penamaan komponen dengan benar.
* Interface 🡪 menampilkan informasi tentang komunikasi.
* Dependecies dan inheritance 🡪 left – right dan bottom – up approach.

1. Interface design 🡪 Easy to learn, use, understand. Golden rules :

* User ada dalam control
* Mengurangi user memory load
* Membuat interface konsisten

1. Komponen desain pada Web App :

* Cohesive function
* Cohesive package content & functionality

1. WebApps design



1. Mobile app design :

* Formulation
* Planning
* Analysis
* Engineering
* Testing dan implementasi
* Evaluasi

1. Market research 🡪 proses penelitian di bidang pemasaran yang dilakukan secara sistematis, mulai dari perumusan masalah sampai mendapat hasil. Mencakup hal berikut :

* Siapa yang menggunakan aplikasi ?
* Kapan aplikasi ini digunakan ?
* Dimana bisa digunakan ? (tempat yang akan digunakan)
* Apa yang membuat aplikasi ini ada ?
* Mengapa menggunakan aplikasi ini ? (dari sisi user) dan Mengapa membuat aplikasi ini ? (dari sisi developer).

1. Tujuan market research :

* Mendefinisikan market oppurtunities dan masalah.
* Evaluasi market.

1. Informasi yang di research :

* Market 🡪 mengetahui orang yang akan mendapat benefit dari produk kita, diciptakan untuk orang yang tepat.
* Industry 🡪 mengetahui kompetitor agar bisa menciptakan produk yang lebih baik.
* Macro – Environment (politik, sosial, ekonomi, teknologi) 🡪 agar tidak mengganggu dari sisi politik, ekonomi, sosial, serta dari sisi teknologi yang digunakan adalah teknologi baru.

1. Design process yang baik :

* Memenuhi tujuan 🡪 sesuai dengan tujuan utama aplikasi.
* Cost effective 🡪 mengeluarkan biaya yang efektif / tidak mahal.
* Sukses 🡪 aplikasi harus sukses, bisa digunakan, sesuai prosedur.
* Iconic 🡪 aplikasi memiliki keunikan tersendiri.
* Long – Lasting 🡪 bisa digunakan terus – menerus.

1. Faktor yang mempengaruhi kinerja desain :

* Cost effective
* Memenuhi / memuaskan service requirement
* Dapat dibuat – achievable

1. Dari sisi software, faktor ini mempengaruhi karena memperhitungkan orang yang bekerja / developer aplikasi. Dapat memuaskan dari sisi requirement, desain, dan implementasi. Dan tercapainya tujuan utama aplikasi.
2. Desain process, tahapannya :

* Problem recognition :
* Brief 🡪 identifikasi masalah, kebutuhan, situasi.
* Investigation 🡪 riset, analisis topik, spesifikasi requirement, mengumpulkan informasi.
* Creative Phase :
* Ide 🡪 yang memenuhi brief dan functional requirement.
* Evaluasi 🡪 evaluasi ide dengan spesifikasi.
* Developing 🡪 sketching, modelling.
* Implementasi 🡪 apakah telah sesuai tujuan / bisa digunakan?
* Planning 🡪 merencanakan produk yang bisa direalisasikan.
* Manufacturing 🡪 membuat product design.
* Mengkaji design original 🡪 design yang baik akan bertahan sepanjang waktu.

1. New product development 🡪 proses pengembangan produk baru dan membawanya masuk ke market.
2. Tantangan pada market development :

* Sumber daya
* Waktu
* Biaya
* Spesifikasi produk

1. Overcoming tantangan yang ada dengan melibatkan semua fungsi untuk mengatasi tantangan yang ada.
2. Strategi new product development :

* Memiliki ide 🡪 produk baru, peningkatan yang ada dari produk.
* Bekerja sama dengan perusahaan, paten, izin / license.

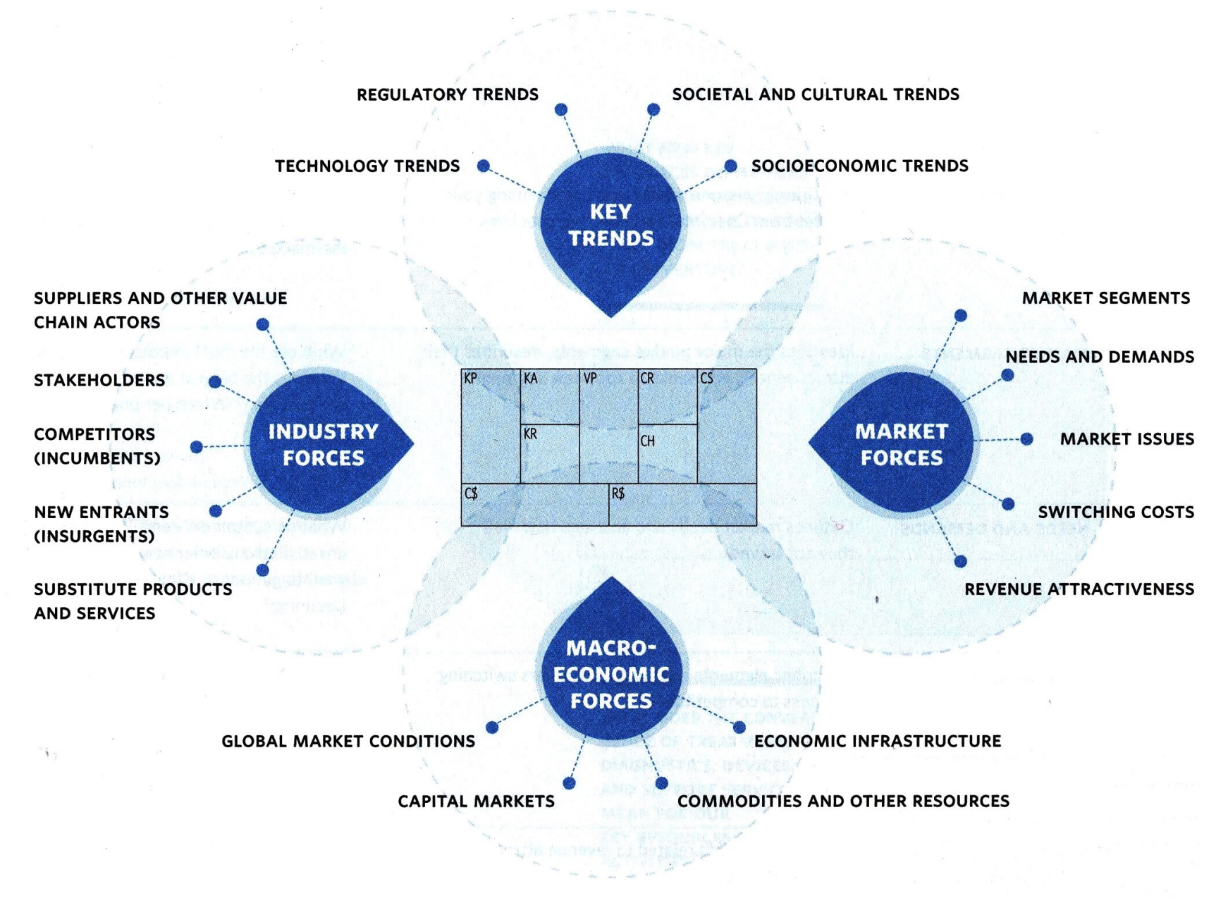
1. Kegagalan produk dapat disebabkan oleh hal – hal berikut :

* Tidak memenuhi keinginan
* Masalah design yang berefek pada fungsionalitas
* Cara promosi, harga, positioning yang kurang tepat
* Respons kompetitif
* Development yang over budget

1. Strategi pada development produk :

* Idea generation
* Idea screening
* Concept development & testing
* Marketing strategies
* Business analysis
* Product Development
* Test Marketing
* Commercialization

1. Business Model Environment :



1. Market forces :

* Market Segments 🡪 segmen yang dituju.
* Kebutuhan dan permintaan 🡪 membuat daftar kebutuhan dan menganalisanya.
* Market issues 🡪 issue pada market, customer, dan perspektif.
* Switching costs 🡪 elemen yang menyebabkan customer berpindah produk.
* Revenue 🡪 elemen berkaitan dengan revenue dan pricing power.

1. Industry forces :

* Kompetitor 🡪 mengidentifikasi siapa kompetitor kita dan kekuatan mereka.
* New entrants 🡪 menentukan siapa pemain baru dan bagaimana agar bisa tetap bersaing.
* Produk pengganti 🡪 produk apa yang berpotensi menggantikan produk kita.
* Supplier 🡪 menjelaskan key value dalam market.
* Stakeholder 🡪 menjelaskan actor yang berperan dalam bisnis.

1. Key Trends :

* Technology 🡪 identifikasi teknologi yang digunakan untuk menunjang bisnis.
* Regulatory 🡪 identifikasi aturan – aturan yang berlaku yang mempengaruhi bisnis.
* Social dan budaya 🡪 identifikasi tren sosial dan budaya yang menunjang bisnis.
* Sosioekonomik 🡪 menentukan tren sosioekonomik yang relevan ke bisnis.

1. Macroeconomics forces :

* Kondisi global market 🡪 mewakilkan keseluruhan perspektif makroekonomik.
* Capital market 🡪 menjelaskan kondisi capital market berkaitan dengan kebutuhan kita.
* Sumber daya 🡪 harga saat ini dan tren sumber daya untuk bisnis kita.
* Infrastruktur ekonomi 🡪 menjelaskan infrastruktur ekonomi pada market pada bisnis.