No: 1

Date 29 Dec 2021

Nama : EKO Saputra
NIM : 20192001

Kelas : IP3A

MK: Rekayasa Perangkat Lunak
Tugas 8

Rekayasa Desain

Desain dan kualitas

- Desain harus mengimplementasikan semua kebutuhan eksplisit yang ada dalam model analisis, dan harus mengkomendasikan semua kebutuhan implisit yang diinginkan oleh konsumen.
- Desain harus dapat berupa panduan yang dapat dibaca dan dipahami oleh orang yang akan membuat kode, dan yang menguji dan mendukung PL tersebut
- Desain harus menyediakan gambaran utuhdari PL, menggambarkan domain data fungsional, dan penilaian dari perspektif implementasi.

Panduan kualitas

- Sebuah desain harus menampilkan arsitektur
- Sebuah desain harus berbentuk modular
- Sebuah desain harus berisi represtasi yang berbeda
- Sebuah desain harus meyadi komponen yang menunjukan karakteristik fungsional yang independen.
- Sebuah desain harus dipresentasikan menggunakan yang secara efektif

Prinsip-prinsip Desain

- Proses desain tidak boleh berjalan dengan "kaca mata kuda"
- Proses desain tidak boleh mengabang penemuan-penemuan dasar.
- Desain harus menampakan keseragaman dan integrasi
- Desain bukan coding dan coding bukan bukan desain
- Desain harus di review untuk meminimalkan kesalahan semantik.

Konsep dasar

- Abstraksi
- Arsitektur
- Patterns / Pola
- Modularitas
- Menyembunyikan
- Independensi fungsi
- Perincmen
- Refactoring

No.

Date

<-

Arsitektur

" Struktur kesatuan dari PL dan cara dimana struktur menyediakan integritas konseptual bagi sebuah sistem

- Properti struktual, Aspek representasi desain arsitektur ini merupakan komponen sebuah sistem, dan pola dari komponen tersebut dipaket dan berinteraksi satu dengan yang lain.

- Properti extra-fungsional. Deskripsi desain arsitektur harus menggambarkan arsitektur mencapai sebuah kinerja yang baik.

- Keluarga atau sistem yang berhubungan. Desain arsitektur harus dapat menggambar pola-pola yang diulang.

Patterns / Pola

Desain Pattern

- Nama Pattern
- Also knows as
- Mastuation
- Aplikabilities
- Struktur
- Participants
- Golaborations
- Konsekuensi
- Recatool Pattens

Refactoring

- Fowler menderfinisikan refactoring spb:
  - Refactoring adalah proses mengubah sistem PL dimana mengubah kode (desain) sehingga meningkatkan struktur internal

L>

←

- Ketika PL refactored, desain akan tetap terhadap :
  - Redundancy
  - Elemen yang tak berguna
  - Algoritma yang tak perlu
  - Struktur data yang tidak tepat
  - Atau kesalahan desain lain untuk diperbaiki

Konsep desain OO

- Desain class
- Interface
- Messages
- Polymorphism

- Desain class
  - Analisis Kelas disempurnakan untuk menjadi class-class entitas
  - Class -class di kembangkan untuk membuat interface
  - Class -class kontrol untuk mengelola
    - Pembuatan objek entitas
    - Komunikasi komplek objek

- Interface

  Pilihan Desain :
  - Class dapat didesain dan dibangun dari nol.
  - Hiperki class berguna untuk mencari kemungkinan jika kelas yang lebih tinggi membutuhkan operasi yang berguna
  - Karakteristik dari klass yang sudah ada dapat di Override

- Polymorphisme

Semua graphs menjadi subclass dari class umum yang disebut graph menggunakan konsep overloading. Setiap class mendefinisikan operasi

↳

No.

Date

Design Patterns

- Desain terbagi disegala bidang tetap mempunyai keterbatasan untuk melihat pola yang memecahkan sebuah masalah.
- Sebuah deskripsi dari design Pattern dapat juga dilihat sebagai sekumpulan desain forces.
  - Desain forces menjelaskan kebutuhan non fungsional yang dibutuhkan dengan PL dimana Patterns diaplimentasikan
  - karakteristik Pattern (class, tanggung jawab, dan kolaborasi)

Frameworks

- Sebuah framework bukan merupakan patterns arsitektur, namun lebih merupakan kerangka dengan sekumpulan "Plug Points" yang memungkinkan untuk beradaptasi dengan domain permasalahan tertentu.