



Catatan Kuliah

Rekayasa Perangkat Lunak (Software Engineering)

Bagian 2

Software Engineering: A Practitioner's Approach, 6/e

Chapter 9 Rekayasa Desain



Arsitektur

“Struktur keseluruhan dari PL dan cara dimana struktur menyediakan integritas konseptual bagi sebuah sistem” [SHA95a]

Properti Struktural. Aspek representasi desain arsitektur ini menentukan komponen-komponen sebuah sistem (mis : modul, objek, filter), dan pola komponen-komponen tersebut dipaket dan berinteraksi satu dengan yang lain. Sebagai contoh : objek dipaket untuk enkapsulasi baik data dan proses yang memanipulasi data dan berinteraksi dengan invokasi method

Properti extra-fungsional. Deskripsi desain arsitektur harus menggambarkan bagaimana arsitektur mencapai kebutuhan kinerja, kapasitas, reliabilitas, keamanan, adaptabilitas, dan karakteristik sistem yang lain.

Keluarga atau sistem-sistem yang berhubungan. Desain arsitektur harus dapat menggambar pola-pola yang diulang, yang secara umum ditemukan dalam disain keluarga atau sistem yang mirip. Esensinya, desain harus mempunyai kemampuan untuk menggunakan kembali blok-blok bangunan arsitektur



Patterns/Pola

Design Pattern Template

Nama Pattern—menggambarkan esensi pattern dalam nama yang singkat tapi ekspresif

Intent/Tujuan—menjelaskan pattern dan apa yang dilakukan

Juga dikenal sebagai/Also-known-as—Daftar sinonim untuk pattern terkait

Motivation/Motivasi—menyediakan contoh masalah

Aplikabilitas—menjelaskan situasi desain spesifik dimana pattern dapat diterapkan

Struktur—menggambarkan class yang dibutuhkan untuk implementasi pattern

Participants—menggambarkan tanggungjawab class-class yang diperlukan untuk mengimplementasikan pattern

Collaborations—menggambarkan bagaimana partcipian berkolaborasi untuk memikul tanggung jawabnya.

Konsekuensi—menggambarkan pengaruh desain terhadap pattern dan potensi masalah yang harus diperhatikan ketika pattern diimplementasi.

Related patterns—relasi referensi silang design patterns

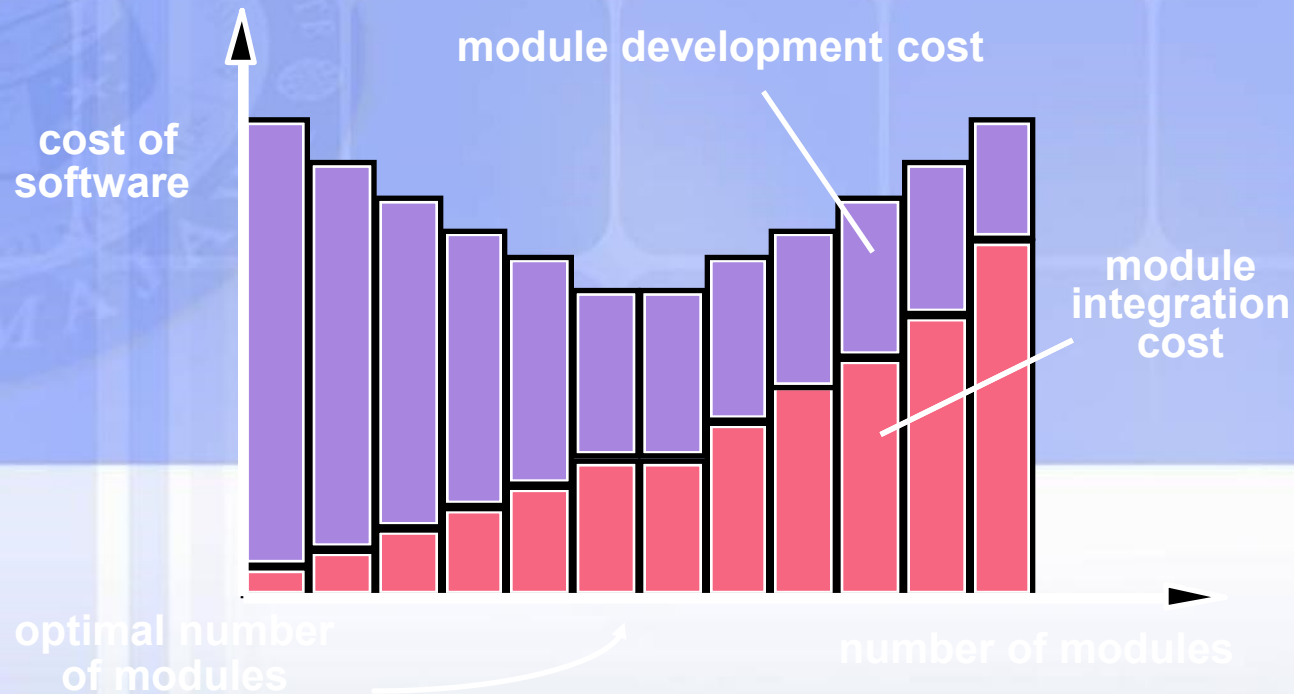
Desain Modular

easier to build, easier to change, easier to fix ...

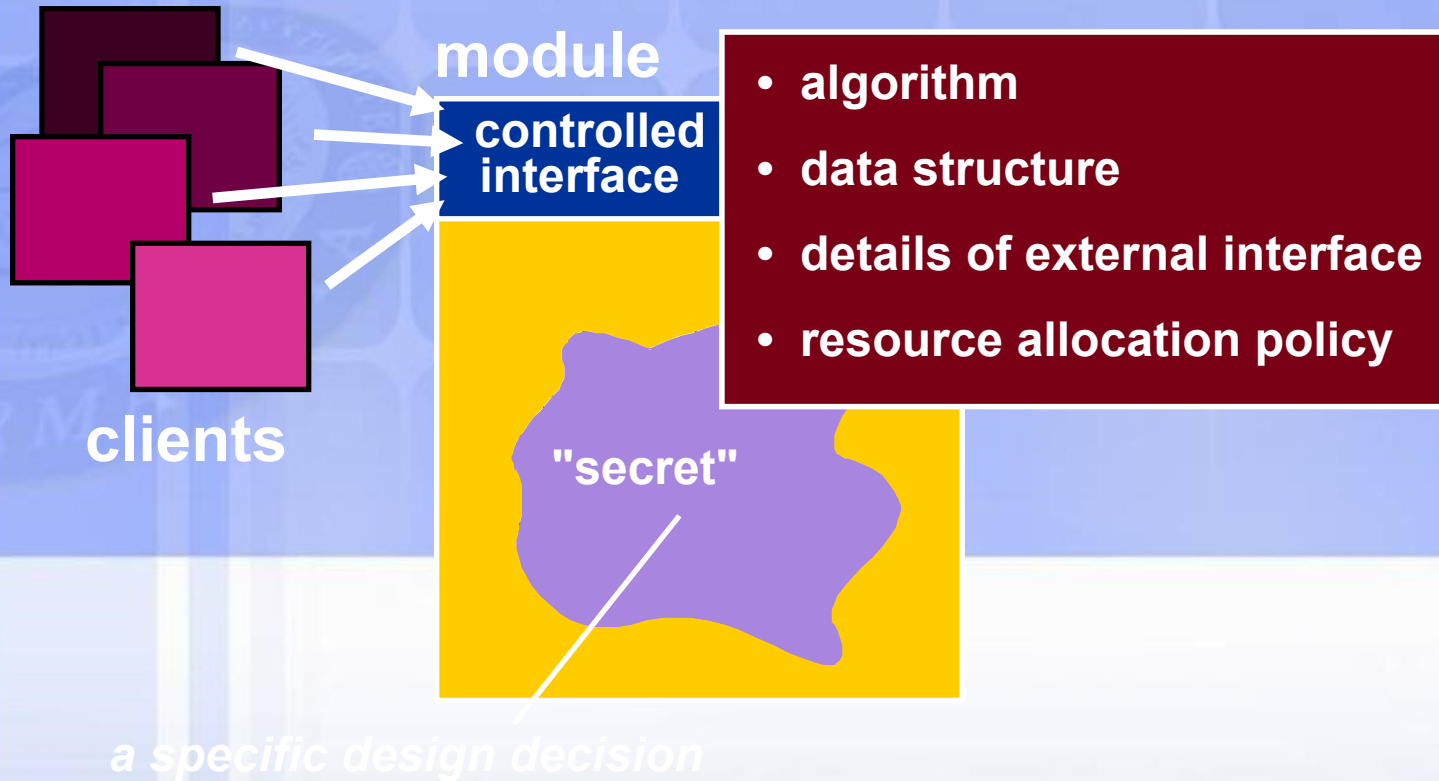


Modularitas: Permasalahan

*Berapakah jumlah modul yang pas
Untuk desain PL tertentu ?*



Penyembunyian Informasi





Mengapa Informasi Disembunyikan?

- Mengurangi “efek samping”
- Membatasi pengaruh global dari keputusan desain lokal
- Menekankan komunikasi melalui interface yang terkendali
- Mengurangi penggunaan data global
- Merujuk pada enkapsulasi—sebuah atribut dari desain kualitas tinggi
- Menghasilkan PL dengan kualitas tinggi

Langkah-langkah Refinement

open

walk to door;
reach for knob;

open door;

walk through;
close door.

repeat until door opens
turn knob clockwise;
if knob doesn't turn, then
take key out;
find correct key;
insert in lock;
endif
pull/push door
move out of way;
end repeat

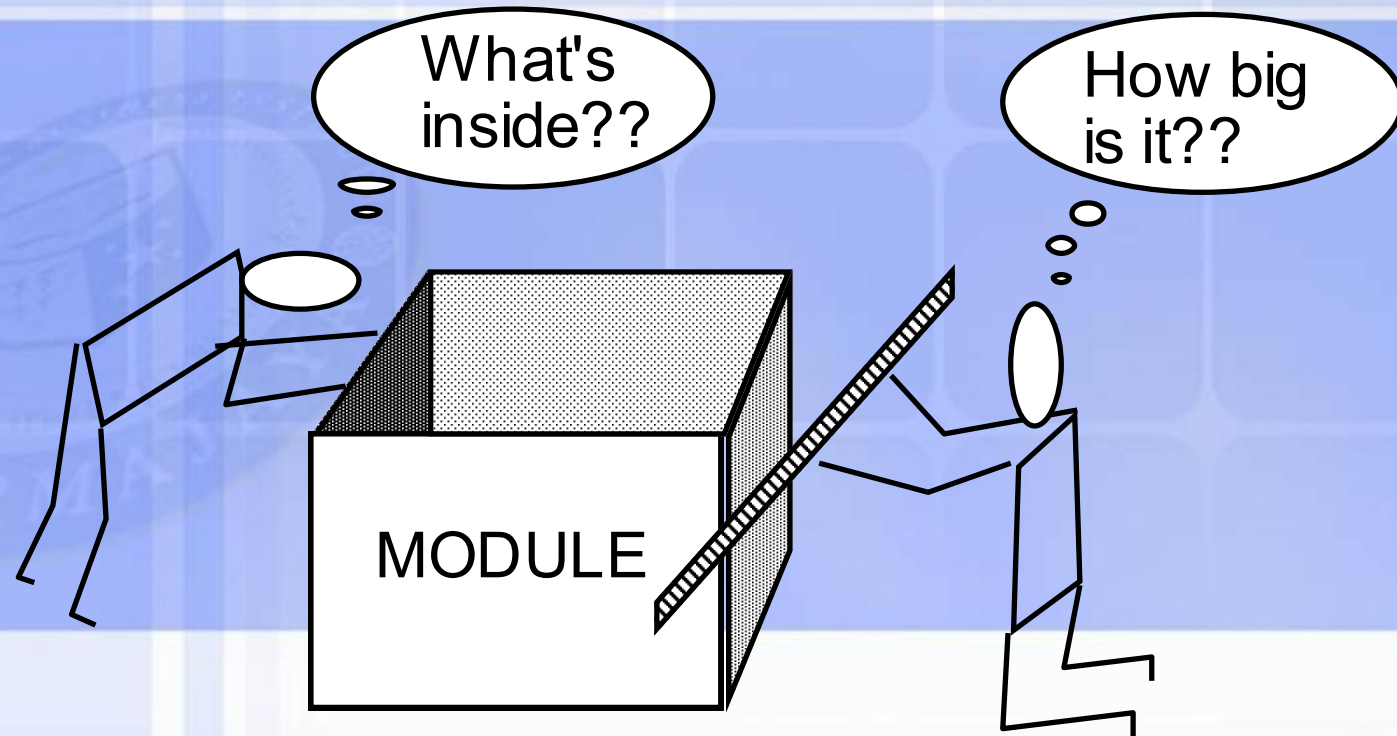


Independensi Fungsi

COHESION - the degree to which a module performs one and only one function.

COUPLING - the degree to which a module is "connected" to other modules in the system.

Mengukur modul: dua sudut pandang



Refactoring

- Fowler [FOW99] mendefinisikan refactoring sbb :
 - "Refactoring adalah proses mengubah sistem PL dimana dia tidak mengubah perilaku eksternal dari kode (desain) sehingga meningkatkan struktur internal."
- Ketika PL refactored, desain akan diperiksa terhadap :
 - redundancy
 - Elemen desain yang tidak berguna
 - Algoritma yang tidak efisien atau tidak perlu
 - Struktur data dengan konstruksi yang buruk atau tidak tepat
 - Atau kesalahan desain lain yang dapat diperiksa untuk menghasilkan desain yang lebih baik.