SWE2301: INTRODUCTION TO SOFTWARE ENGINEERING

Lecture 06 : Software Design (1)
Venue: CIT Theater
Time: 12-1pm
Presented by M. I. Mukhtar

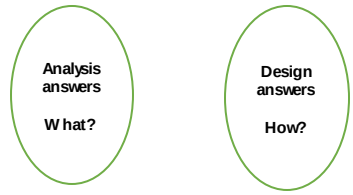
Software Design

- Software design is the process of transforming all the requirements gathered through system analysis into some suitable form that describes the complete structure of the system.
- The major goal of the design process is to describe the system in a diagrammatic form that programmers can easily implement through the writing of code into a working system.

Software Design..

- Software Design is more creative process than analysis because it answers "How?" the system will be developed.
- Requirement specification document produced at the end of the analysis phase which tells "what" a system does, becomes an input to the design process, which tells "how" a software system works.

Relationship between Analysis and Design



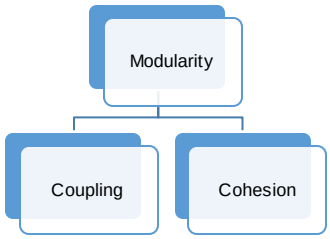
Software Design Objectives

- The design needs to be:
  - correct
  - complete
  - understandable
  - maintainable

Software Design Consideration

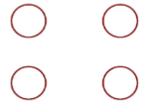
- Modularity is the single attribute of software that allows a program to be intellectually manageable.
- A system is considered modular if:
  - it consists of discreet components so that each component can be implemented separately,
  - a change to one component has minimal impact on other components.

Modularity



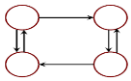
Coupling

- Coupling is the measure of the degree of interdependence between modules.
- "**Uncoupled"** systems are made up of modules which are not dependent at all on each other.

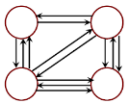


Coupling

- "**Loosely coupled"** systems are made up of modules which are relatively independent.



- "**Highly coupled"** systems are made up of modules which share a great deal of dependencies.



Cohesion

- Cohesion is a measure of the degree to which the elements of a module are functionally related.

Types of Cohesion

Functional Cohesion	Best (high)
Sequential Cohesion	
Communicational Cohesion	
Procedural Cohesion	
Temporal Cohesion	
Logical Cohesion	
Coincidental Cohesion	Worst (low)

Cohesion Contd.

- **Functional Cohesion:** A and B are part of a single functional task.
- **Sequential Cohesion:** Module A outputs some data which forms the input to B.
- **Communicational Cohesion**: A and B both operate on the same input data or contribute towards the same output data
- **Procedural Cohesion:** This occurs in modules that accomplish different tasks but follow same specific order in which the tasks are to be completed.

Cohesion Contd.

- **Temporal Cohesion:** This occurs in modules that are grouped by when they are processed. The parts are processed at a particular time in program execution.
- **Logical Cohesion:** This occurs in modules that contain instructions that appear to be related because they fall into the same logical class of functions.
- **Coincidental Cohesion:** This occurs in modules that contain instructions that have little or no relationship to one another.

Relationship btw Coupling and Cohesion

- The essence of the design process is that the system is decomposed into parts to facilitate understanding and modifying a system
- If the software is not properly modularized, trivial enhancement or changes will result into death of the project.
- Therefore, a software engineer must design the modules with goal of **low coupling** and **high cohesion**.

Summary

- Software Design answers "How?" the system will be developed.
- A system is considered modular if
  - it consists of discreet components so that each component can be implemented separately and
  - a change to one component has minimal impact on other components.

Questions ??