Slide 06 : Software Design

Made by Md. Mehedi Hasan Rafy

Software Design

Software Design is the process of transforming user requirements (from SRS) into a detailed form suitable for coding and implementation.

- Shift focus from *problem domain* to *solution domain*.
- Output can be directly used in programming.
- It tries to specify how to fulfill the requirements mentioned in SRS.

Levels of Software Design

1. ***Architectural Design***
    - Highest abstraction level.
    - Represents system as a set of interacting components.
    - Gives overall structure of the solution domain.
2. ***High-level Design***
    - Breaks ‘single entity-multiple component’ architecture into less abstracted view of ‘sub-systems and modules’.
    - Shows module structure and their interactions.
    - Focuses on implementable module design.
    - Recognizes modular structure of each sub-system and their relation and interaction among each other.
3. ***Detailed Design***
    - Specifies logic, internal structure, and interfaces of each module.
    - Ready for coding.

Modularization

Modularization is a technique to divide a software system into multiple discrete and independent modules, which are expected to be capable of carrying out tasks independently.

- Modules can be compiled or executed separately.
- Follows *divide-and-conquer* approach.
- Easier development, debugging, maintenance, and re-usability.

Concurrency

Concurrency in the context of software engineering means to split the software into multiple independent units of execution, like modules and executing them in parallel.

- Provides the capability to execute more than one part of code in parallel to each other.
- It is necessary for the programmers and designer to recognize those modules, which can be executed in parallel.

Measures of the quality of a design of modules

1. Cohesion (↑)
2. Coupling (↓)

Cohesion

Cohesion is a measure that defines the degree of intra-dependability within elements of a module.

The greater the Cohesion, the better is the program design.

There are ***seven*** types of cohesion. They are –

1. ***Coincidental (worst)*** – Random, unplanned modularization, poor design. 2. ***Logical*** – logically categorized elements are put together into a module. 3. ***Temporal*** – Elements of module are processed at the same time. 4. ***Procedural*** – Elements of module executed in sequence to perform a task.	5. ***Communicational*** – Elements of module executed sequentially and operate on the same data. 6. ***Sequential*** – Output of one element is input to another and so on. 7. ***Functional (best)*** – All elements contribute to a single well-defined function (best). It is reusable.

Coupling

Coupling is a measure that defines the level of inter-dependability among modules of a program. It tells at what level the modules interfere and interact with each other.

There are five levels of coupling. They are –

1. **Content** – One module directly accesses or modifies another's content.
2. **Common** – Multiple modules share read and write access to some global data.
3. **Control** – One module controls the function or flow of execution of another.
4. **Stamp** – Modules share a common data structure but work on different parts.
5. **Data** – Modules interact by means of passing data as parameter. If a module passes a data structure as parameter, then the receiving module should use all its components.

Design Verification

- The output of software design process is design documentation, pseudo codes, detailed logic diagrams, process diagrams, and detailed description of all functional or non-functional requirements.
- Implementation of software depends on the outputs of this stage.
- Detects defects early, improving quality and reducing cost.
- A structured verification approach is used by reviewers to detect defects that might be caused by overlooking some conditions.