

Design process

- Software design is an iterative process through which requirements are translated into a “blueprint” for constructing the software.
- Initially, the design is represented at a high level of abstraction.
- As design iterations occur, the design is represented at much lower levels of abstraction.

Software Quality Guidelines and Attributes

Three characteristics that serve as a guide for the evaluation of a good design:

- All of the explicit requirements contained in the requirements model and all of the implicit requirements desired by stakeholders must be implemented during the design.
- The design must be a readable, understandable guide for developers those who generate code and testers those who test and subsequently support the software.
- The design should provide a complete picture of the software from an implementation perspective by addressing the data, functional, and behavioural domains.

Software quality guidelines

1. A design should create an architecture using the recognizable architectural styles and composed of components with good design characteristic and it is implemented in evolutionary manner for testing.
2. A design should be modular, that is the software must be logically partitioned into elements.
3. A design should represent data, architecture, interface and components in well-defined manner.
4. A design should lead to proper data structure.
5. A design should lead to components with independent functional characteristic.
6. A design should lead to simple interface with minimum connections between the components.
7. A design should be derived iteratively with the obtained information from software requirement analysis.
8. The notations used in design representation should effectively communicates its meaning.

Quality attributes

The attributes of design given in the acronym 'FURPS' are listed as follows:

Functionality:

It is assessed by evaluating the feature set and capabilities of the program, functionality delivered and security of overall system.

Usability:

It is accessed by considering the human factors, overall aesthetics, consistency and documentation.

Reliability:

It is evaluated by measuring parameters like frequency and severity of failure, accuracy of output result, the mean-time-to-failure(MTTF), recovery from failure and the program predictability.

Performance:

It is measured by considering processing speed, response time, resource consumption, throughput and efficiency.

Supportability:

- It combines the ability to extend the program, adaptability, serviceability. These three factors represent common term maintainability.
- Testability, compatibility and configurability are the terms used with which a system can be easily installed and found the problem easily.
- Supportability also consists of more attributes such as compatibility, extensibility, fault tolerance, modularity, reusability, robustness, security, portability, scalability.

The Evolution of Software Design

The evolution of software design is a continuing process that has now spanned almost six decades. Early design focus on development of modular programs in top down manner, that is called as structured programming.

Newer design approaches proposed an object-oriented approach to design that is bottom up manner.

Each software design method introduces unique approach.

All of the above-mentioned design methods have common characteristics

1. A mechanism for the translation of the requirements model into a design representation
2. A notation for representing functional components and their interfaces
3. Heuristics for refinement and partitioning.
4. Guidelines for quality assessment.

Text Books:

1. Roger Pressman S., "Software Engineering: A Practitioner's Approach", 7th Edition, McGraw Hill, 2010.
2. Sommerville, "Software Engineering", Eighth Edition, Pearson Education, 2007

Web Links

1. <https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-355j-software-engineering-concepts-fall-2005/lecture-notes/cnotes4.pdf>
2. <https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-355j-software-engineering-concepts-fall-2005/lecture-notes/cnotes5.pdf>.
3. https://cdn.shopify.com/s/files/1/0457/4009/7694/files/software_engineering_pdf_pressman_7th_edition.pdf