Iqra National University Peshawar Pakistan

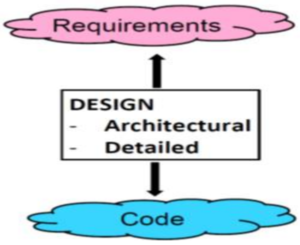
Department of Computer Science

Spring Semester, Mid-Assignment, April 2020

Subject:	**Software Design**	Issue Date:	**13/April/2020**
Program:	**MS (CS & SE)**	Submission Date:	**17/April/2020**
Teacher Name:	**Dr. Fazal-e-Malik**		
Student Name	**Rooh Ullah Jan**	ID	**6611**

Q.No:1(a) is the Software Design needed in the Software Development Projects? If yes then explain how it is important?

Software design is the most important phase of the software development cycle. Software design is the first step in SDLC (Software Design Life Cycle), which moves the concentration from problem domain to solution domain. It tries to specify how to fulfill the requirements mentioned in SRS (Software Requirements Specification) .Without a proper design we risk building an unstable system means without a proper design our system has no stability.



Importance of software design

Abstraction: The design produces the best abstract of software. It allows understanding and meeting the requirements in the best possible way.

Prevents Redundancy: The software design helps in reducing redundancy in software by identifying the modules having same characteristics and functionality.

Reusability: The software design enables us to re-use different modules/functions/codes/components having the same functionality instead of performing same task multiple times.

Reduces Risk: It enables us to reduce risk in the software by identifying problems in the design phase which may lead into the failure of software. The components are identified and mitigated at this stage.

User Friendly Interface: The software design must have a user friendly interface as it allows both owner and developer to realize how the software will look and how its functions will be performed. It must combine functionality, in formativeness and simplicity as it will provide us with the initial vision of the software product we want to receive.

Provides Scalability: The design enables software to be scalable. If any changes are required or any additional module needs to be incorporated, it gives room to add new modules or made changes to the existing modules with affecting other components.

Ease in Testing: The software design provides ease by testing the software as a whole or testing individual component without affecting other modules or overall functionality of the software.

Q.No:1(b) as one of the most complex man-made artifacts (objects), computer software is very difficult to design. Explain what are the main challenges in software design?

Some of the major challenges are given below.

1: The Requirements Volatility

A major reason for the complexity of software projects is the constant change of requirements. When designed properly, software can be modified or extended easily; however, when designed poorly. Modifying software can become overwhelming and lead to all sorts of complex problems. Requirements volatility is challenging because they impact future or already going development efforts.

2: The Process

In the design phase, software processes involve a broad set of activities and tasks that bridge the gap between requirements and construction while adhering to a set of project-specific (or company-specific) constraints. These activities include common ones, such as architectural and detailed design, as well as other supporting management activities. These supporting activities include establishing a design review process, defining design quality evaluation criteria, evaluating design reuse, establishing design change management and version control procedures, design tool adoption, resource allocation, and others. In many cases, a company’s design process is not well established, poorly understood, or approached with minimalistic expectations that ignore aspects that are essential to executing a successful design phase. Focusing design efforts on creating independent software products, such as a simple class diagram or user interface, while ignoring other design activities may create complexities later on during system's test and maintenance. The design process is challenging because essential design process activities are often overlooked, done in an ad-hoc manner, or simply not done at all.

3: The Technology

Besides the operating environment, the technology for designing and implementing today's software systems continues to evolve to provide improved capabilities. Examples of these include modeling languages and tools, programming languages, development environments, design patterns, and design strategies. In some cases, emerging technologies do not completely replace old ones. Some software systems are required to interoperate with old legacy systems designed with older design methodologies. This results in software designers employing a myriad of design methodologies and technologies, all on the same software system. In other cases, design models need to be derived from existing code, modified, and made interoperable with newer technologies. This technology-driven aspect of the design phase creates a demand for capable software designers that can assimilate new technology quickly and effectively in order to succeed at designing software. The technology aspect of software design is challenging because it is fast and ever-changing, therefore designers must keep abreast of the latest advances and become proficient in the application of these advancements while maintaining rooted in legacy technology.

4: The Ethical and Professional Practices

Software designers must exhibit strong ethical and professional practices to ensure that the systems they build are of highest quality and that all design considerations are properly evaluated. In many cases, this requires designers to exert strong leadership skills to influence and negotiate with stakeholders, motivate the design team, and lead the design process to accomplish the project's goals.

5: Managing Design Influences

Designs are shaped by many different influences from stakeholders, the development organization, and other factors. These influences can have cyclical effects between the system and its external influences, such that external factors affect the development of the system and the system

Affects its external factors. Managing these influences is essential for maximizing the quality of systems and their related influence on future business opportunities.

Q.No:2 the literature on design methods began to appear in the 1950s and 60s. Since then, design methodology has become an independent discipline of scientific study.

a) What are the essential characteristics of design?

Some essential characteristics of design are given below.

For good quality software to be produced, the software design must also be of good quality. Now, the matter of concern is how the quality of good software design is measured? This is done by observing certain factors in software design. These factors are:

1. Correctness
2. Understandability
3. Efficiency
4. Maintainability

Now, let us define each of them in detail.

1) **Correctness**

First of all, the design of any software is evaluated for its correctness. The evaluators check the software for every kind of input and action and observe the results that the software will produce according to the proposed design. If the results are correct for every input, the design is accepted and is considered that the software produced according to this design will function correctly.

2) **Understandability**

The software design should be understandable so that the developers do not find any difficulty to understand it. Good software design should be self-explanatory. This is because there are hundreds and thousands of developers that develop different modules of the software, and it would be very time consuming to explain each design to each developer. So, if the design is easy and self- explanatory, it would be easy for the developers to implement it and build the same software that is represented in the design.

3) **Efficiency**

The software design must be efficient. The efficiency of the software can be estimated from the design phase itself, because if the design is describing software that is not efficient and useful, then the developed software would also stand on the same level of efficiency. Hence, for efficient and good quality software to be developed, care must be taken in the designing phase itself.

4) **Maintainability**

The software design must be in such a way that modifications can be easily made in it. This is because every software needs time to time modifications and maintenance. So, the design of the software must also be able to bear such changes. It should not be the case that after making some modifications the other features of the software start misbehaving. Any change made in the software design must not affect the other available features, and if the features are getting affected, then they must be handled properly.

Q.No:2(b) what activities of Software Design Process are used by designers?

The design process activities used by designer are the followings:

1. **Architectural design**: The sub-systems of system and their relationships are identified based on the main functional requirements of software.
2. **Abstract specification:** For each sub-system, an abstract specification of its services and the constraints under which it must operate is defined. The expression of specification may be in text in a natural language, in a specification language, in a formal mathematical language and diagrammatic technique is also used to represent or express specification. Also characteristics of good specification are that it should be complete, verifiable, modifiable consistent, etc.

3. **Interface design**: Interfaces allow the sub-system' services to be used by other sub-systems. The representation of interface should be hidden. In this activity the interface is designed and documented for each sub-system. The specification of interface must be unambiguous.
4. **Component design**: Services are allocated to components and the interfaces of these components are designed. Where you take each system component and design how it will operate.
5. **Data structure design:** The data structures used in the system implementation are designed in detail and specified. Also the designer

designs the system data structures and how these are to be represented in a database.

6. **Algorithm design**: In this activity the algorithms used to provide services are designed in detail and specified.

As shown in the below picture.

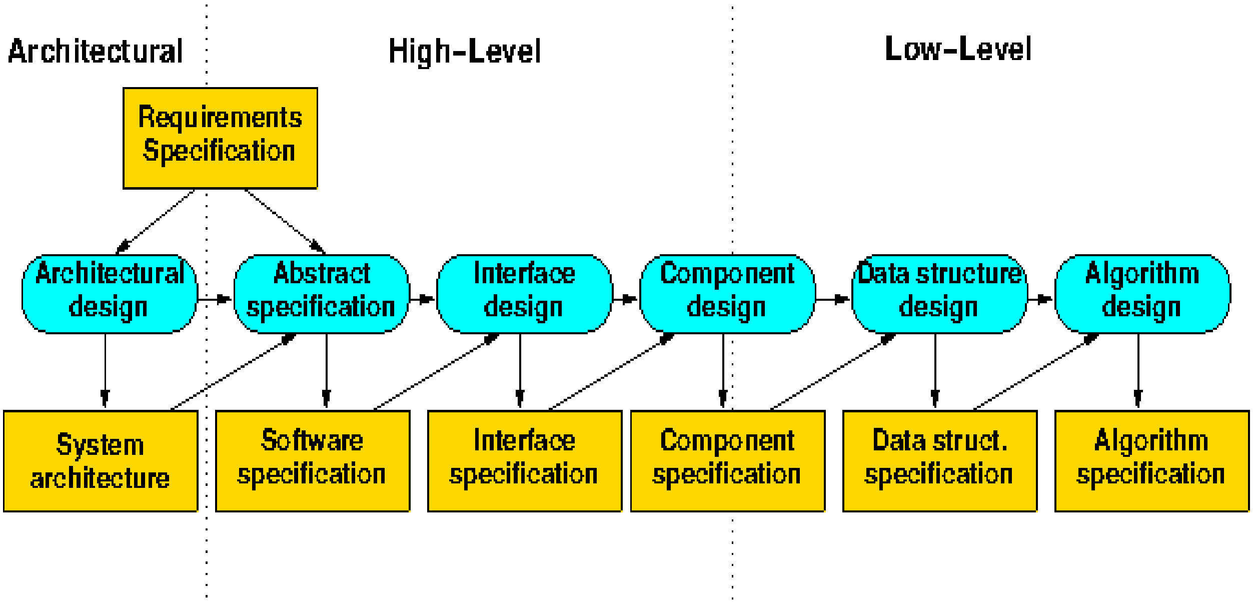


Fig for software design process.

The above fig shows all the design activities.

Q.No:3 Study and explain Models other than McCall' Model of attributes. Does the software Design affect these Models?

- **Boehm Model -** This model describes how easily and reliably a software product can be used. This model actually elaborates the aspects of McCall model in detail. It begins with the characteristics that resort to higher level requirements. The model's general utility is divided into various factors - portability, efficiency and human engineering, which are the refinement of factors like portability and utility. Further maintainability is refined into testability, understandability and modifiability.

- **FURPS Model** -This model categories requirements into functional and non-functional requirements. The term FURPS is an acronym for Functional requirement(F) which relies on expected input and output, and in nonfunctional requirements (U) stands for Usability which includes human factors, aesthetic, documentation of user material of training, (R) stands for reliability(frequency and severity of failure, time among failure), (P) stands for Performance that includes functional requirements, and finally (S) stands for supportability that includes backup, requirement of design and implementation etc.

- **Ghezzi Model** -This model states that the internal qualities of a software help the software developers to attain a collaborative result both in terms of external and internal qualities of a software. The overall qualities can be accuracy, flexibility, integrity, maintainability, portability, reliability, re-usability and usability.

- **IEEE Model** -It is a standard which defines various specifications for software maintenance, thus providing a quality model. This model gives a variety of measurement techniques for various qualitative factors like efficiency, functionality, maintainability, portability, reliability and usability.

- **Dromey's Quality Model** -This model emphasizes on evaluating one software's quality with another. It helps to find out defects if any, and also to point out the factors that caused such defects. This model is designed on the basis of the relationship that exist between software properties and its quality attributes.

- **SATC's Model** -SATC is an acronym for Software Assurance Technology Centre. Its objective is to improve software quality by defining metrics program which helps to meet the basic needs with least expenditure. This model tests a quality model by evaluating the results of the metrics used, and also on the basis of discussions based on the project. This model defines set of goals and process attributes based on the structure of ISO 9126-1 quality model.

- **ISO 9126-1 Quality Model** -This model has two primary categories – internal and external quality attributes and quality in use attributes. The internal quality attributes are the properties of the system the evaluation of which can be done without executing it. Whereas the external quality attributes are those that are evaluated by observing the system during execution.

- **Capability Maturity Model -**One of the most important quality models of software quality maintenance. The model lays down a very simple approach to define the quality standards. It has five levels namely – initial, repeatable, defined, managed, optimizing.

Effect of Software Design on Quality Modals: The Software design (architecture design, interface design and detailed design) affect the following key quality attributes of software.

- **Efficiency:** It describes the response of the system in some interval of time. The time a system takes to perform an activity between different components (individual Components)/Parallel Computation by components. A bad software design would lead to degradation of the performance of the system.
- **Correctness:** Correctness is to implement the requirements specified by user in such a way so that the software becomes error free and accurate. Incorrect implementation at any level of design, leads to incorrect implementation of specified requirements.
- **Reliability:_** The software to perform as per the user requirements and consistently well at a specified environment in a given period of time. It largely depends on the amount of faults (i.e. defects) in the software system which could arise due to faults at design structure, errors in algorithm and data structure design, flaws in coding and implementation. Quality design decomposes a complicated design problem into simpler sub-problems, to reduce the probability of errors.
- **Portability:** The ability to move software from one software/hardware platform to another.
- **Maintainability:_**Characteristic of design to restore a failed equipment, machine, or software system to its normal operable state in a given timeframe. Software design provides ease in both corrective and Adaptive maintenance. Maintainability of software mostly depends on Architectural design.
- **Reusability:_** To reuse components of software in the development of other software systems.
- **Interoperability:_**The ability of independently-developed components to interact and cooperate with each other. Architectural mismatch could lead Software's lack of interoperability due to bad design.