

# Strategies for Software Design

Jan 16 , 2018

[Share 0](#)[Tweet](#)[Share](#)[\(https://www.pinterest.com/pin/create/button/\)](https://www.pinterest.com/pin/create/button/)

It is a known fact that software development is a task that involves several complex and time consuming activities, which are combined together to create and design a software that is unique and different from its counterparts, as well as has exceptional features and quality. Moreover, before initiating the process of software development, the team of developers, testers, managers, and other stakeholders connected to the project, go through various stages of planning, designing, programming, and more. It is only after achieving success at each of these stages that the process of software development is initiated. These various stages of Software Development Life Cycle (SDLC) are extremely significant and carry a lot of importance.

Software engineers put in their great efforts to plan the whole development process of the project and ensure that no step is overlooked or considered unnecessary. Similarly, software designing too is a very significant stage of Software Development Life Cycle (SDLC) and is treated with utmost care and importance. Software engineers, before starting the development and testing process, design a basic structure of the software, which determines what techniques,

methodologies, tools and more will be used to complete the development process. Furthermore, it also determines the types of tests that will be executed on the software. Hence, here is a discussion on software design and its strategies to help you understand its significance in Software Development Life Cycle (SDLC).

## **What is Software Design?**

The first step in Software Development Life Cycle (SDLC), Software Design, is a process of implementing software solutions to one or more sets of problems. It transforms users and clients requirements into some suitable form, which helps programmer in coding and implementation. Software design usually involves problem solving and planning a software solution. This includes both low level component and algorithm design, as well as a high level architecture design. Moreover, software design moves the concentration from problem domain to solution drama. It tries to specify how to fulfil the requirements mentioned in Software Requirement Specifications (SRS).

Also, for assessing user requirements Software Requirement Specification (SRS) document is created, whereas for coding and implementation, there is a need of more specific and detailed requirements in software terms. The output of this process can directly be used into implementation in programming languages. Software design is both a process and model, wherein the former is a sequence of steps that enables the designer to describe all aspects of the software for building.

## **Defining Software Design Strategy:**

Software Design Strategy is a discipline that helps firms in deciding what to make, why to make it and how to innovate contextually, both immediately and over the long term. This process involves strategic design and interplay between design and business strategy. The software design strategy is mainly about organising design activities during the

course of design. The organisation of design activities, be it planned or ad-hoc reflects a designer's approach to creating a design. During the process of software development it is common that a software designer thinks about a list of design issues and tasks that need to be discussed and addressed as well as are just to be done in the ad-hoc way. Furthermore, design strategy can assist a designer in resolving several common problems such as:

- Promoting the adoption of technology.
- Identifying the most important questions that a company's products and services should address.
- Translating insights into actionable solutions.
- Prioritizing the order in which a portfolio of a products and services should be launched.
- Connecting design efforts to an organisation's strategies.
- Integrating design as a fundamental aspect of strategic brand intent.

### **Strategies Used For Software Design:**

To implement software design, software engineers use a variety of strategies that help them define their respective tasks and assists them with the designing process. The way these design strategies are intermixed and used can influence the outcome and the effectiveness of the final design. Moreover, these strategies may not be required every time, but they are immensely helpful in getting expected results and in mitigating the risks of any course of action. With the help of

these strategies one can design a proper plan of software development and ensure that all the demands, requirements and request of the clients are taken care of. Therefore, here are few of these strategies used by software engineers for designing accurate as well as bug free software product.

- a. **Structured Design:** This is a conceptualisation of problems into several well organised elements of solutions. It is mainly concerned about the solution design. The biggest advantage of structured design is that it gives better understanding of how the problem is being solved. Structured design also makes it simpler for the designer to concentrate on the problem more accurately. Additionally, a well-structured design follows some rules for communication among multiple modules, like cohesion and coupling, where cohesion is the grouping of all functionally related elements and coupling is the communication between different modules. In short, a well-structured design has high cohesion and low coupling arrangement.
- b. **Function-Oriented Design:** This is one of the classical methods of software design, where decomposition centres on identifying the major software function and then elaborating and refining them in a top-down manner. In Function oriented design, the system is comprised of many smaller sub-systems known as function. These functions are capable of performing significant tasks in the system. Also, in this the system is considered as top view of all functions. Furthermore, function oriented design inherits some qualities and properties of Structured Design, where divide and conquer methodology is used. This type of design works well where the system state does not matter and the program or function works on input rather than on state.
- c. **Object-Oriented Design:** Object oriented design works with around the entities and characteristics rather than with the function involved in the software system. The whole concept of software solution revolves around the

engaged entities. The important concepts of object oriented design are:

- **Objects:** These are all the entities involved in the solution design. Each of these entities have some attributes associated with it and has some methods to perform on the attributes.
- **Classes:** It is a generalised description of an object. It defines all the attributes, which an object can have as well as the methods that define the functionality of the object.
- **Encapsulation:** The attributes and methods when bundled together are known as encapsulation. The encapsulation not only bundles the important information about an object together, but also restricts the access of data from the outside world.
- **Inheritance:** In inheritance, the similar classes are stacked up together in a hierarchical manner, where the lower or sub classes can import, implement, and re-use the variables and methods from their immediate super classes.
- **Polymorphism:** Object oriented design provides a mechanism where methods performing similar tasks are assigned a same name, even though they vary in argument. This is known as polymorphism, which allows a single interface to perform tasks for different types.

**Conclusion:**

Software Development Life Cycle (SDLC) is a culmination of numerous activities, each of which play a significant role in developing a product that has exceptional functionality, scalability, performance, quality and more. To achieve great success in creating a perfect software product, one needs to follow each and every stage and step of Software Development Life Cycle (SDLC), which will not only assist a software engineer in developing, testing, programming, and designing the product, but will also allow them to validate whether the software is being developed as per the demands of the client or the user. Software design is one such important step of SDLC, which helps in planning the structure as well as the process of development. From deciding the team efforts, testing techniques, and designing strategies to tools and other important factors affecting the process of software development, all takes place in this stage. Software designing also ensures that the requirements of the client are taken care of and are implemented in the software properly. Hence, it is with the assistance of software design that one can ensure that the whole process goes on smoothly and as per the decided plan.

Give your feedback!

## Subscribe Pro QA

Enter your email

Submit

Data Flow Testing (data-flow-testing)

Test Process (test-process)

Smoke vs Sanity Testing (smoke-testing-vs-sanity-testing)

Sanity Testing (sanity-testing)

Complete Selenium Guide (selenium)

Smoke vs Regression Testing (smoke-vs-regression-testing)

Regression Testing (types-of-defects-in-software-testing)

Regression vs Retesting (regression-testing-vs-retesting)

Content Testing (content-testing)

Test Case Prioritization (test-prioritization)

Monkey Testing (monkey-testing)

Software Review (software-review)

Regression Testing (regression-testing)

Beta Testing Guide (beta-testing)

Test Closure (test-closure)

Active & Passive Testing (active-passive-testing)

Entry and Exit Criteria (entry-and-exit-criteria)

Stubs and Drivers (stubs-and-drivers)

---

## Related Posts:

---

## Follow Us:



(<https://www.facebook.com/qa.professional>)



(<https://twitter.com/ProfessionalQA>)



(<https://www.linkedin.com/in/professionalqa>)



(<https://plus.google.com/u/0/109878639147270322851>)



(<https://www.pinterest.com/professionalqa/>)



(<https://www.instagram.com/professionalqa16/>)



(<http://feeds.feedburner.com/professionalqa>)

We deliver. Get the best of  
Professional QA (/) in your inbox.

You can opt-out at any time. See our privacy policy ([privacy-policy](#)).

---

[About Us \(about-us\)](#)

[Privacy Policy \(privacy-policy\)](#)



[Terms And Conditions \(terms-and-conditions\)](#)

[Disclaimer \(disclaimer\)](#)

[Sitemap \(sitemap.xml\)](#)

[Contact Us \(contact-us\)](#)

**© 2019 ProfessionalQA.com - All rights reserved.**