



Strategy to Implement Project by Effective Software Design and Programming

Vikram Singh*

Bachelor of Engineering, Programmer Analyst, SCJP, SAFe Agilist, IEEE- Computer Society, India

***Corresponding Author:** Vikram Singh, Bachelor of Engineering, Programmer Analyst, SCJP, SAFe Agilist, IEEE- Computer Society, India.

Received: May 16, 2019; **Published:** November 06, 2019

Abstract

Software Design is very important phase for any Software Development Projects. During the design phase the programmers will discuss and analyze the business requirements with Product owners or stake holders and chalk out the strategy for the implementation. During the design phase programmers will design the flow chart or blue print and identify possible roadblocks and dependencies. During the design phase programmers will determine what software tools and technologies will be used and provide the timelines for coding, testing and deployments. In this article I will discuss on how to implement the software with effective design and programming skills.

Keywords: Software Design; Programming; Software Reliability; Software Maintenance; Software Reusability; Software Testing; Software Deployment

Introduction

The Software Design and Programming article is to provide the information which will be used to aid in Software Development projects by giving details on how the software implementation should be done.

Software design

Software design is the process of conceptualizing the software requirements and then transform into software programming & implementation. Software design is one of the important phases in software development life cycle process. There are two categories in Software Design, as follows

1. **High Level Design:** In this category the software architects and programmers will discuss what all modules are involved, is there any dependencies between the modules, what software tools and technologies can be used and what are the potential road blocks and issues that needs to be identified.
2. **Low Level Design:** In this category the architect's and programmers will determine system and subsystem in design, typically the project or business specific implementation will be discussed such as flow charts, user interface view, attributes, database connectivity, performance issues, security aspects will be discussed and documented.

The important characteristics of programming language is as follows-

- **Software Compatibility:** During the design phase architects and programmers should verify that the software is interoperability with another product.
- **Software Extensibility:** The effective software design should be able to add the new capabilities to the software without major changes to the existing architecture.
- **Modul Design:** The components could be implemented and tested in isolation before being integrated to form a desired software system. Hence the during the design phase it must be identified that the modules can be programmed and tested independently.
- **Software Maintenance:** This is an important part in SDLC life cycle where a project team need to measure how easily bug are fixed or functional modifications can be done without major impact to the system.
- **Software Licensing:** The Architects and Programmers of the software project team should make sure that the software is able to perform a required function under stated conditions for a specified period of time.
- **Software Application Security:** The Software infrastructure and security analyst should make sure that the software is able to withstand and resist hostile acts and influences.

- **Software Performance:** The system architects and performance engineers should make sure that the software performs its tasks within a time-frame that is acceptable as per the industry standards, and does not require too much memory.
- **Software Portability:** The project team during the design phase should determine that the software is usable across a number of different scenarios and environments.
- **Software Scalability:** During the software design phase the architects should determine that the developed software shall adapt well to increasing data or number of users.

Software programming

Software Programming is the next important phase of SDLC life cycle. After the software design is approved, the software programmers will come in to play. During the programming phase the software programmer will determine the feature for the module which is allocated to him/her. Based on the task allocated to programmers, they will start analyzing their task and break down the task into a minimum marketable product. Once the Analysis and task are completed, programmers will install all the required tools and software into their work station as per the company license and standards so that they can program or code in the programming language [1-5].

The important characteristics of programming language is as follows-

- **Easy to Implement:** The software programmer should make sure that the programming language that they are implementing it must be simple, easy to implement and have good readability.
- **Abstraction of Software Program:** The Abstraction is highly required for a programming language in which ability to define the complex structure and then its degree of usability comes.
- **Portability:** The Software Programmer should determine that the software program is portable.
- **Programming Efficiency:** A Programmer should determine that the language's efficiency must be high so that it can be easily converted into a system code and consumes little space in memory.
- **Indentation and Documentation:** An efficient programmer should always structure the program and indent it should it is reliable for the application development.
- **Software Tools:** An efficient programmer should verify that all the necessary tools for development, debugging, testing, maintenance of a program are available for a programming language.

- **Integrated Development Environment:** A project team of programmer should be provided with a single environment known as Integrated Development Environment (IDE).
- **Syntax and Semantics:** A project team of programmers should maintain the consistency in terms of syntax and semantics.

Conclusion

The effective Software Design and Programming has continuously transformed the industry into a next level technology in recent years. The robust software design and programming will make significant developments in the areas of Cloud Computing, Machine Learning, Internet of Things, Artificial Intelligence and Robotics. As a result, new methodologies and technologies are developed which are ideal for software industry.

Bibliography

1. BW Boehm. "The High Cost of Software in Practical Strategies for Developing Large Software System". MA, Reading: Addison-Wesley (1975).
2. KN McKay and J Laube. "Search and rescue: A case study of design flexibility". Simulation Conference Proceedings 1988 Winter (1988): 381-388.
3. TL Roberts., *et al.* "Factors that impact implementing a system development methodology". Software Engineering IEEE Transactions on 24.8 (1998): 640-649.
4. GA Ludgate., *et al.* "The Use of Structured Analysis and Design in the Engineering of the TRIUMF Data Acquisition and Analysis System". Nuclear Science IEEE Transactions 34.1 (1987): 157-161.
5. Xiping Song., *et al.* "Experience with an approach to comparing software design methodologies". Software Engineering IEEE Transactions 20.5 (1994): 364-384.

Volume 1 Issue 1 November 2019

© All rights are reserved by Vikram Singh.