



A study of SDLC to develop well engineered software

Mohit Kumar Sharma

Ph.D. Research Scholar (GNA University)

Head, Sr. Assistant Professor

Post Graduate Department of Computer Science

Jagdish Chandra D.A.V. College, Dasuya, Punjab, India

Abstract: Software Development Life Cycle (SDLC) is an important concept used in software engineering to describe a procedure for planning, creating, coding, testing and implementation of user requirement specification. Software development life cycle applies to a range of hardware and software configurations. SDLC is step by step process for creating quality software for users. It involves different phases that are followed one after one, that are essential for software engineers such as planning, analysis, design, coding, testing and implementation. In the early years, hardware was costly and software relatively cheap. In digital era, hardware is cheap and software is expensive. So the costs of hardware and software have been reversed due to increased demand of well engineered software products. This paper includes different phases of SDLC, software quality, qualities of well engineered software and factors affecting software quality.

Key words: SDLC, Phases, Software Quality, Factors

I. INTRODUCTION

In Computer language, a program is collection of instructions or statements or an executable code, which serves some specific purpose. Software is considered to be collection of executable programs, standard libraries and document specifications. Software is a collection of computer programs, procedure and documentation that are used for specific purpose.

In the early years, hardware was costly and software relatively cheap. In digital era, hardware is cheap and software is expensive. So the costs of hardware and software have been reversed due to increased demand of well engineered software products. Different software categories as system software, application software, scientific software, embedded software, web application software and artificial intelligence software are used for specific purposes. So, there should be requirement of well planned mechanism to create such good quality software.

describe a procedure for planning, creating, coding, testing and implementation of user requirement specification. The systems development life-cycle applies to a range of hardware and software configurations. SDLC is step by step process for creating quality software for users. [1] It involves different phases that are followed one after one, that are essential for software developers, such as planning, analysis, design, coding and implementation. It includes evaluation of present software, information gathering, and feasibility study and request approval.

II. OBJECTIVES OF THE STUDY

- A. To create awareness about the proper implementation of software development life cycle.
- B. To encourage software engineers to use SDLC for developing well engineered software.
- C. To create awareness about the qualities of well engineered software.
- D. To study factors affecting software quality.

III. PHASES OF SOFTWARE DEVELOPMENT LIFE CYCLE

Software development life cycle (SDLC) is a method by which quality software can be developed in the given time and according to the customer expectations. [5] SDLC is step by step process for creating quality software for users. It involves different phases that are followed one after one, that are essential for software engineers to follow such as-

1. **Requirement specification and gathering** - This phase starts by the software engineers to collect all user requirements to carry on the development of software. The software developing team holds discussions with the users for problem and tries to bring out as much information as possible on their requirements. The requirements are collected from user, system requirements and functional requirements. The requirements are collected by using studying the existing system and software, conducting

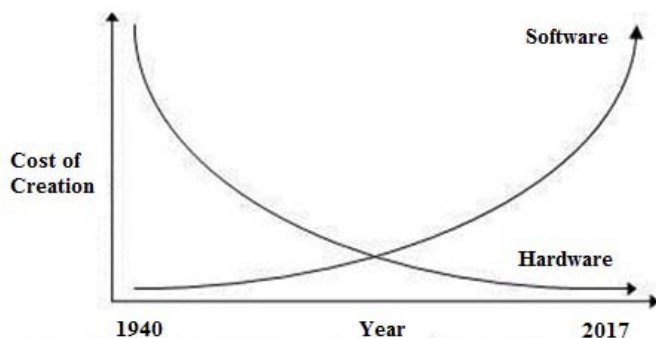


Fig 1. Hardware vs Software Costs

Software engineering is an engineering branch associated with development of quality software using well-defined scientific principles, methods and procedures. The output of software engineering is an efficient and reliable well engineered software product. Software Development Life Cycle (SDLC) is concept used in software engineering to

interviews of users, referring to the database or collecting answers by the questionnaires. This phase should be done very carefully, that means software quality depends on all correct information collected from user.

- Feasibility Study** – This is important phase after requirement specification and gathering, the software engineer team comes up with a next plan of software process to analyze if software can be designed to fulfill all requirements of the user, and if there is any possibility of software being no more useful. It is also analyzed if the software is financially, practically and technically feasible for the organization to take up. It should be financial feasible that means within budget. It should be practically and technically feasible, that means easily operated by users in future. There are many algorithms and procedures available, which help the software engineers to conclude the feasibility study.

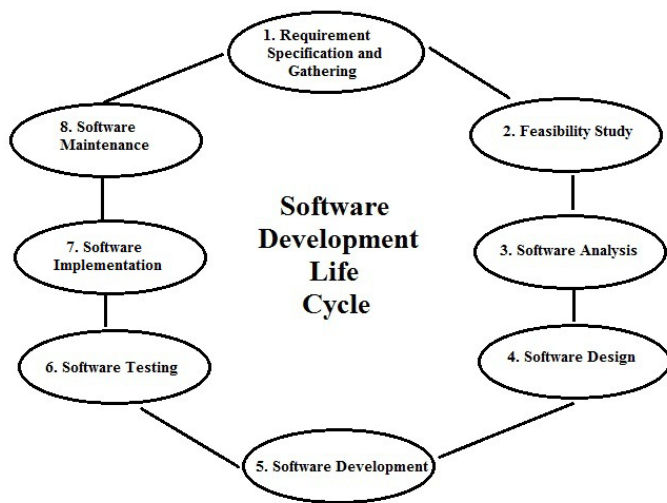


Fig 2. Phases of Software Development Life Cycle

- Software Analysis** - At this next phase, the software engineer team should analyze the software user requirements and remove anomalies, errors or not required information from the specifications, plan and try to bring up the best software model suitable for the project. Software analysis includes understanding of software product limitations, system related problems or changes to be done in existing systems. The software engineer team analyzes the scope of the software and plans the schedule and resources accordingly.
- Software Design** – At next phase is to bring down whole knowledge of requirements and analysis for design the software. Various designs like functional design and object oriented design are available for designing software product. Various tools like data flow diagram and Entity relationship diagram, data dictionaries can be used for designing. The inputs from users and information gathered in requirement gathering phase are the inputs of this step.
- Software Development or Coding** - This phase is also known as programming or coding phase. The practical development of software design starts in terms of writing program code in the suitable programming language and developing error free executable programs efficiently.

Software may need to be integrated with the libraries, databases, and other program. The software developer team should be expert in programming skills required to develop a good software product.

- Software Testing** - Software testing is done after creating software to remove errors or mistakes, bugs to make it error free good quality software product. While coding by the developers and testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing, object oriented testing, and testing the product at static and dynamic levels. Testing time consumes more time as compare to other phases of SDLC.
- Software Implementation** - This phase includes installing the software on user computer. At times, software needs post-installation configurations at user end. It includes all hardware and software requirements to run the developed and tested software. Software is tested for portability and adaptability is solved during implementation. It also includes training of software to the user for efficient working.
- Software Maintenance** - This phase is very important phase for maintenance of software products that means after implementation of software, software may be failing or any error can occurs or to add new features to currently software. The aim of software maintenance to remove errors or bugs, to change platforms requirements and add new features to existing software. There are corrective, perfective and adaptive maintenance for correcting, adding and change platform to existing software to make more reliable.

IV. SOFTWARE QUALITY

Software quality is a procedure that evaluates, assesses, and improves the performance of software. Software quality is defined as the degree to which software meets requirements for reliability, maintainability, portability as contrasted with functional, performance, and interface requirements that are satisfied as a result of software engineering.

Software quality involves user requirements and performance requirements, documentation and all well engineered qualities required for all professionally developed software.

Software quality assurance is defined as a planned and systematic approach to the evaluation of the quality, software product standards, processes and procedures. [2] The goal of quality assurance is to provide management with the data necessary to be informed about product quality. Quality of conformance is an issue focused primarily on implementation. If the implementation follows the design and the resulting system meets its requirements and performance goals, conformance quality is high quality. Software quality assurance plays an important role in software development life cycle. [4]

Quality assurance process starts with planning & conducting inspection and reviews. It is an ongoing process within the software development life cycle that routinely

checks the developed software to ensure it meets desired quality measures. [3]

V. QUALITIES OF WELL ENGINEERED SOFTWARE

Software development life cycle phases ensures these qualities if followed step by step to get well engineered software product as-

1. **Efficiency** – Software should be well developed and ensures efficiency. Computer resources should be utilized in such a way that software should not make wasteful use of computer system resources such as memory and processor cycles. Efficiency includes responsiveness, processing time, and memory utilization properly.
2. **Maintainability** – Software should be coded in such a way so that it can evolve to meet the changing needs of users. The aim of maintenance to remove errors or bugs, to change platforms requirements and add new features to existing software.
3. **Functionality** – Software should be function able properly as per user need. It is the process and procedures used to establish the suitability of a system that is, the ability of a system to meet the needs of a user.
4. **Accuracy** – It is correctness and function should be according to user or customer requirements. There should be error free software.
5. **Security** - Software security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources.
6. **Acceptability** - Software must be acceptable to the users for which it is designed as per user requirements. It must be understandable, usable, and compatible with other systems. Software should be developed on time for user and must be within budget.
7. **Reliability** – Software should be reliable that means it ensures error free software and run on compatible user systems. After testing of software, it makes software more reliable.

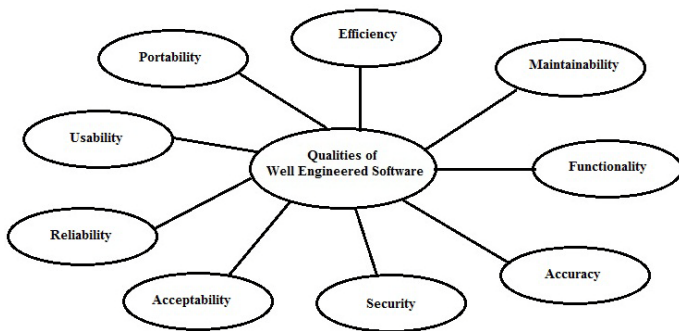


Fig 3. Qualities of Well-engineered software

8. **Usability** – Software must be easy to learn and operate by the user. It should be developed in such a way that makes

user friendly. Its learn ability to the user easy to operate and handle.

9. **Portability** – It is important quality required for software. Software must be the ability to be transferred from one environment to another. It should be easily installed on any other machine easily.

VI. FACTORS AFFECTING SOFTWARE QUALITY

The quality requirements encompass all levels of software production. The following are the factors affecting software quality as-

1. **Correctness** - Correctness is the extent to which a program satisfies its specifications. If software does not work correctly as required then it is wasteful.
2. **Reliability** - Reliability is the property that defines how well the software meets its requirements. If software is not error free then it is not reliable.

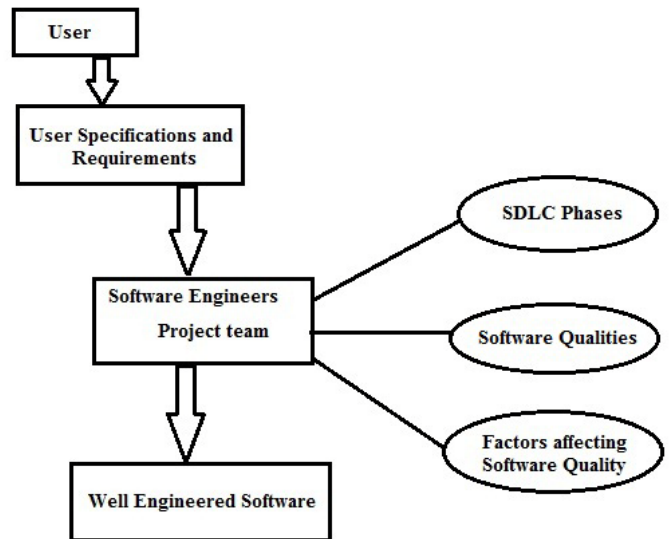


Fig 4. Process of well-engineered software

3. **Usability** - Usability or the effort required locating and fixing errors in operating programs. If software is not user friendly then it is hard for user to use.
4. **Extensibility** - Extensibility is the ability of the software to be extended beyond the functionality of the original product. If software coding is not extendable by adding new features then it has no worth.
5. **Reusability** - Reusability is the extent to which parts of the software can be reused in other related applications. If software is not reusable then it is limited product.
6. **Testability** - Testability is the effort required to test to ensure that the system or a module performs its intended function. If testing is not done properly then it makes software errors.

7. **Portability** - Portability is the effort required to transfer the software from one configuration to another. If software is not able to install any other machine then it is wasteful.
8. **Maintainability**- Maintainability is the effort required to maintain the system in order to check the quality. If software is not able to add new features or to remove error, then it has no worth.
9. **Efficiency** - Efficiency is a factor relating to all issues in the execution of software; it includes considerations such as response time, memory requirement, and throughput. If software is not making computer fast then it is wasteful.

VII. CONCLUSION

Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. Software Development Life Cycle (SDLC) is concept used in software engineering to describe a procedure for planning, developing, testing, and implementation of user requirement specification. Software engineers should follow SDLC phases for creating well engineered software product for users and must concerns on factor affecting software quality.

VIII. REFERENCES

- [1] Shubhmeet Kaur, "A Review of Software Development Life Cycle Models", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 11, November 2015 ISSN: 2277 128X.
- [2] C. SenthilMurugan and S. Prakasam, "A Literal Review of Software Quality Assurance", International Journal of Computer Applications, ISSN - 0975 – 8887, Volume 78 – No.8, September 2013.
- [3] Anu Gupta, "Quality Assurance and Its Standards: Importance in Various SDLC Models", International Journal of Advanced research in Computer Science and Management studies, Volume 2, Issue 12, December 2014, ISSN: 2321-7782.
- [4] Chaitali Roy and Mousumi Saha, "The Realm of Software Quality Assurance", International Journal of innovations and Engineering and Technology.
- [5] Shubham Dwivedi, " Software Development Life Cycle Models - A Comparative analysis", International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 2, February 2016.