International Journal of Humanities, Engineering, Science and Management (IJHESM)
Volume II, Issue – I, JUNE 2021
SJIF Impact Factor 5.81
ISSN : 2582-8169

The Role of Analysis Phase of SDLC for Small Scale Business Application- A Review

Irfan Ahmad Khan[1], Dr. Dipti Kumari[2]
[1] Research Scholar, [2]Assistant Professor
[1]Department of Computer Application, RKDF University, Ranchi
[1]Email id: irfan16143@gmail.com
[2] Department of Computer Application & Engg., RKDF University, Ranchi
[2]Email id: dipti5679@gmail.com

Abstract:

Software quality completely depends on various phases of SDLC. When we referred to the term "Software quality" it means well analysed and completely defined software according to the user requirements. The different phases of SDLC play a vital role in software development. Selection of correct model and implementation of all its attributes is the basic for any good quality software. Number of studies and model has been proposed to check the quality of good software. The objective of this paper is to present a review on the role of analysis phase of SDLC for small scale business application. And also, the reasons why small scale businesses are not trying to accept software in their business practice. During the review process we found analysis phase is the most important phase which is actually the base factor of any good quality based software either used by small, medium or large scale based business application.

Keywords: Role of analysis phase in SDLC, small scale application, Software analysis,

Introduction:

Software is developed or engineered; it is not manufactured in the classical sense. A software component should be designed and implemented so that it can be reused in many different programs. Software is a logical rather than a physical system element [1]. Anything which is developed, engineered or manufactured having chances to be better than before. Software development is the process in which top place of quality is always empty. There is huge opportunity for Software developers and companies can do better and introduce high quality software. In fact, there are many high quality based software launching in market on every intervals/certain interval but when we talk about small scale businesses and low budget software the quality of software not obtained. There are a lot of issues and multiple updating

somewhere disturbing the business goals of the Client and also putting many technical difficulties to run his business smoothly. Result in many small businesses still not shifted to the software based working. Also the feedback of other businessman and the handling difficulty not encouraging small scale business to step forward on software based working system. Software development and its integrity is a challenge in small scale industries. Many companies thinking to sale the software only and not focusing on the parameters which actually required to fulfil the need of the software. Sometimes software development companies want to deploy the software as early as possible just to close the deal only, later demanding money to update the requirement of the user. In software development the main and basic step is to understand the requirement of the Client. While analysing the requirement proper documentation and steps should be followed. According to IEEE software development process is a process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use [2]. Quality of software cannot be measured or defined in one or two steps. Entire SDLC steps are valuable and need to be followed for any types of software development. Complete documentation isalso important for software quality. Some times for small software developers skip some of the step and handle those steps on the basis of their experience but this is not a professional way that is why some time quality and durability of the software has been questioned. While analysing the Client requirement sometimes budget of software also play a major factor in software quality, if customer is not having sufficient budget so the quality of the software cannot be imagining on top line. One more point to be considered here i.e. the technical awareness of the concerned Client. If Client is technically sound he can explain his requirement in better way that will help analyst to analyse his requirement in better way.

There are different Software development models available and it is used by the companies to develop quality software but there are some issues in some of the traditional development method. And selection of right model is one of the important factors in software development. In today's challenging era quality and error free software is in the high demand. To delivers good products every steps of the SDLC to be observed and followed during the development of software. Software development is becoming more and more challenging because of the raised demand from every business sectors.

The remaining paper is structured as follows: the subsequent section presents important facts about software quality, different SDLC models and importance of analysis phase, security issues in SDLC process. Section 3 describes the metrics of analysis phase and its uses. Section 4 presents effects of analysis phase on software quality. Section 5 presents businessman and analyst perspective on not developing quality software in small scale industries. Section 6 conclusion discussed.

Review on Analysis Phase:

Software Quality and SDLC

If the software is fulfilling the requirement of the user, it can be considered as quality software for the specific user. Fulfilling the requirement of user is a method of customizing the parameter of standard scaled software for the specific client. Especially in small scale industries Software quality may vary from user to user requirement and also sometimes depends on types of business. Software development process also termed as software development life cycle (SDLC) is divided into four phases (Fig. 1): software specifications, software design, software coding/implementation, and software testing. These phases have been used in different models. Each process has a significant impact on software quality characteristics. These stages are critical for improving the quality of finished products [3-4]. In India, software development services go through a series of stages in a step-by-step phase that nearly any developing company follows. The six phases of SDLC includes analysis, design, development, testing, implementation, and maintenance. [5].

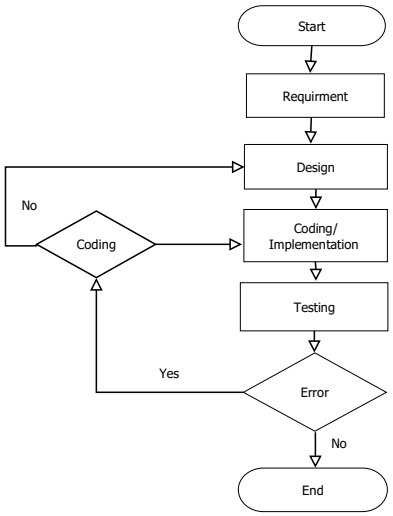

Fig-1 four phases of SDLC

Different models specify the various phases and order in which way the entire process carried out. Some existing and well known SDLC models are (Fig-2):

(I) Waterfall Model- **(II)** V-Shaped Model **(III)** Iterative model- **(IV)** Spiral Model **(V)** Big Bang **(VI)** Agile Model [6].

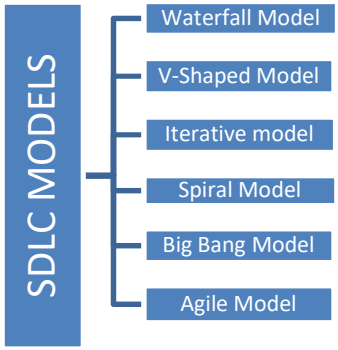

Fig-2, well known SDLC models name

Not all process models are appropriate for all projects. Some process models are best suited to large projects, whereas others are better suited to smaller projects. However, there are several efficient process models that can be used for a variety of projects. Any process model should not be limited to a single form of project or scope of work. The model should be developed in such a way that it can accommodate various types of projects with varying scopes by tailoring the process. As a result, project suitability can be used as an assessment criterion for a process model.

Importance of Analysis Phase

Analysis is the first phase of the SDLC where requirements of the client to be discussed and basic data to be collected. In this phase IT analyst collecting inputs and observing the manual business methodology. While capturing the entire information analyst need to be very careful because this phase plays very important role in entire process. Whatever information is collected all to be implemented in final product. All the requirements should be documented and especially those who define the business need. Analysis of software includes client requirements, system requirements also functional and non-functional requirements to be noted down to match the basic business parameters of the client.

Analysis phase is handled by the senior members of the company. It is completely brainstorming phase which impact various phases. This phase gives a clear picture of the entire project, complete time and budget of the project to be decided in this phase. Both parts of the study, problems and threats, should be monitored. Threat detection and countermeasures must be prepared in conjunction with likelihood.

Threat identification—during the analysis, this procedure is adopted to detect risks. To define threats and direct succeeding design, coding, and testing judgments, threat investigation approach is used. Since several systems have specific criteria that present distinctive risks, identifying safety risk is a strategic task that needs some imagination [7, 8].

Different companies and analyst using their own way to handle the analysis phase, as per the study and observation there is no fixed mechanism for this phase. Sometimes software development companies dealing this phase according to their team structure and availability of the infrastructure, some samples of companies mentioned below.

- **During the analysis process, end-user business requirements are evaluated, and project goals are translated into the specified system functions that the company plans to create. The following are the three main tasks involved in the review phase:**

A. Compiling a list of company specifications
B. Developing flowcharts
C. Conducting a thorough investigation

At this stage of the SDLC, gathering business requirements is important. A collection of business functionalities that the system must fulfill in order to be efficient is referred to as business requirements. This process does not include the definition of technical information such as the types of technology used in the system's implementation. “The system must monitor all employees by their respective agency, area, and designation,” for example, may be a business necessity. This requirement does not specify how the system can execute this requirement; rather, it specifies what the system must do in relation to the company. [9].

- **This segment focuses on security issues that are specific to the SDLC process. The following are some of the most critical security activities for this phase:**

A. Conduct a risk assessment and use the findings to complement the security measures that are already in place.
B. Analyze the protection needs
C. Check for functionality and security.
D. Gather preliminary documentation for device certification and accreditation.

Although the information security elements are presented in this section in a top-down order, the order in which they are completed is not inherently set. Complex device security analysis would need to be iterated before accuracy and completeness are achieved. [10].

- **During this process, companies will concentrate on finding the root of their issue or the need for improvement. In the event of a challenge, potential solutions are submitted and evaluated to determine which is the best match for the project's ultimate goal(s), This is where teams understand the project's or solutions functional criteria. It's also where system analysis, or evaluating the needs of end users to ensure the new system can fulfill their requirements, takes place. Businesses may use a number of techniques that are unique to the second step. They are as follows:** [11]

A. CASE (Computer Aided Systems/Software Engineering) is an acronym for Computer Aided Systems/Software Engineering.
B. Collecting requirements
C. Structured research

The Requirements Analysis Phase's aim is to turn the needs and high-level requirements defined in previous phases into unambiguous (measurable and testable), traceable, full, reliable, and stakeholder-approved requirements. [12].

When having exact specifications from the customer at the start of the project is difficult, **prototyping** becomes necessary. Given the system's prototype, users continue to provide

feedback from time to time, and appropriate changes are implemented into the proposed or to be created system in response to the feedback. Through doing so, the system's secret, undisclosed user requirements can be discovered during the early stages of development. By doing so, project failure risks can be reduced whereas user satisfaction and system efficiency are improved. [13].

Following flow diagram (Fig-3) will completely show the effects and importance of analysis phase in entire SDLC which is actually needed for better quality software.

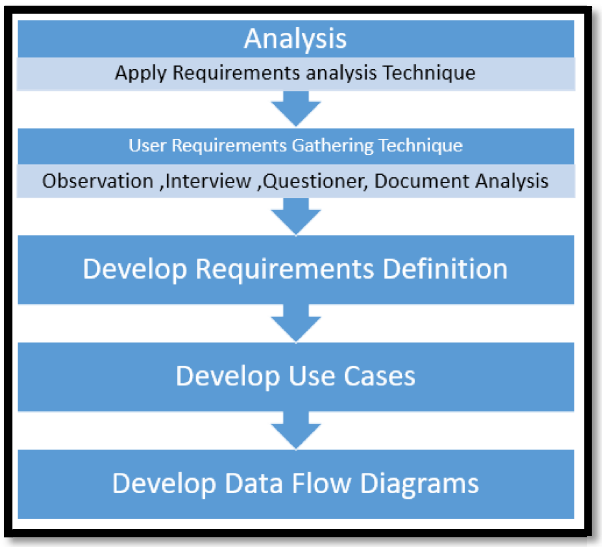

Fig-3, Flow diagram of the analysis phase

Documental Formalities between Client and Company during analysis Phase

As per online data review and discussion with some software analyst of software development companies following documentations are handled during the analysis phase of the SDLC.

BRD - Business Requirement Document
FSD-Functional Specification Document
BRS-Business Requirement Specification
RTM- Requirement Traceability Matrix
SRS- System Requirement Specification

In above listed documents, Business requirement document(BRD) is given by the client side if the client is clear with his requirement on the basis of BRD requirement traceability matrix

(RTM) is prepared by the analyst. This is one of the key stages where client and company agreed with all requirements.

Role of Company Team in Analysis Phase

Project Manager –	Play overall responsibility, specifically on time and Budgeting.
Business Analyst –	Specific domain expert to understand business process of the client
Subject Matter Expert–	Plays role in key point of the project.
System Analyst –	Frame the team requirement for the development and monitoring on quality assurance and control.

Requirement Phase Software Metrics:

In the last two decades, a variety of software defect prediction models based on software metrics have been proposed. These models' defect prediction can be useful for ensuring the quality of software. A previous study found that choosing the right metrics is critical for enhancing defect prediction. The right software metric will help increase prediction accuracy.

Requirement Stability (RS): Requirement modification requests are inversely proportional to requirement stability. There are several methods for collecting requirements. The biggest issue with requirement collection is that stakeholders (users, customers, developers, and project managers) do not have a good understanding of what they want. During the development of a software project, requirements can change at any time. There are two forms of requirements changes: regulated and uncontrolled. Controlled specifications adjustments can occur to provide improvements to the software system's features or to meet evolving consumer needs, along with many other things. These modifications to specifications may be necessary for a system's adaptation to changes in hardware or software. Uncontrolled specifications adjustments can have a negative impact on the project's expense, efficiency, and reliability, as well as its timetable. According to studies, imprecisely specified specifications account for more than half of all software development errors.

Requirement Fault Density (RFD)*:* This metric calculates the percentage of standards specification documents that are incorrect. During the requirement analysis process, the requirement fault density provides a measure of the software quality of developing software.

It's an unintended consequence of requirement engineering. The density of requirement faults can vary from simple to complex, and can affect a large portion of the specification.

Review, Inspection and Walkthrough (RIW)*:* This metric cleans up the software product and can be used at any stage during the production of a software project. Both the software developer and the customer are actively involved in the software specifications specification (SRS) document analysis. The assessment is carried out on many levels. The review process' aim is to make sure the SRS is feasible, total, reliable, and accurate. It is a vital metric from the perspective of efficiency. [14].

Effect of Analysis Phase on Software Quality:

Gathering specifications, negotiating, defining, and validating are all part of the study process. Low-level design, high-level design, architectural, and interface design are all examples of design work. According to a variety of studies performed by software communities, the majority of software product failures are triggered by errors in the specifications and design processes. The following are some of the most common causes of software defects due to negligence at analysis phase.

- Incomplete Specification - A systematic analysis is needed for recommendations to address certain inconsistencies in requirements specifications. The following questions are included in the requirements analysis questionnaire:
  - Are the requirements correct?
  - Have any of the conditions been met?
  - Are they time-bound to ensure that projects are completed on time?
- Customer Communication Misinterpretation-
- Data Representation Error
- Design Logic Error

It's necessary and examines existing requirements to ensure that consumer expectations are accurately converted into product specifications. An iteration of interactive sessions with the customer will aid developers in knowing the true requirements. [15].

"**According to the Crosstalk, Journal of Defense Software Engineering, USA, Errors in the requirements and design phase contributes to 64 percent of total defect costs (Fig-4)**".

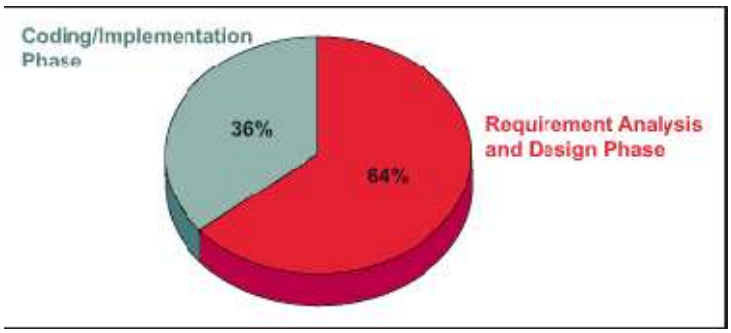

Fig.4. Origin of Defects

(Source: Crosstalk, Journal of Defense Software Engineering)

The project team collects and records formal user specifications during the analysis process of the SDLC." The software requirements are the foundation for subsequent phases (Pressman, 2005)".Defining, evaluating, approving, and laying out the criteria are all significant steps in this process. Since a lack of trust in specifications leads to a lack of quality, it is critical that effective quality controls are applied at this phase. The specifications are checked during this process, and the SQA team ensures that the required documents are being produced.

"Defining exactly what must be done to solve the problem by gathering and analyzing requirements and creating the initial software system model (Davis and Yen, 1999; Galin, 2004)" [16].

Functional and non-functional parameters are the two types of parameters. Functional parameters deal with the application's functionality or factional aspects, while non-functional parameters deal with non-functional (desirable) parameters like usability and maintainability that a developer might not consider during development. Non-functional parameters are usually considered only during the maintenance phase or later in the software development period, resulting in rework or additional effort. As a consequence, it is best practice to consider quality early in the software development process and to produce high-quality software on time. [17].

Reasons of Not Development of Quality Based Software for Small Scale Businesses

❖ **In businessman perspective:**

- A lot of issues and multiple updating somewhere disturbing the business goals of the Client
- Facing many technical difficulties to run his business smoothly.
- Difficulties in software handling due to non-technical background.
- Sometimes software development companies want to deploy the software as early as possible just to close the deal only, later demanding money to update the requirement of the user.

❖ **In software analyst perspective:**

- Software development and its integrity is a challenge in small scale industries.
- Many companies thinking to sale the software only and not focusing on the parameters which actually required to fulfil the need of the software.
- While analysing the requirement proper documentation and steps should be followed.

Conclusion:

The main objective of this review paper is to project on analysis phase of the SDLC. Each phase plays important role in software quality but as per the observation of different papers, online data and personal discussion with some software analyst the most significant phase is analysis phase of the SDLC. Given equation defining the importance of analysis phase (Fig-5):

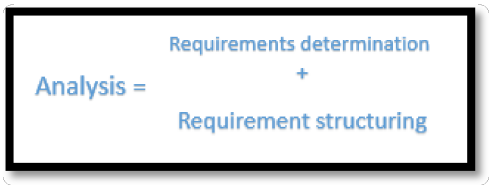

Fig-5, equation defining the importance of analysis phase

Importance of this phase is noted because of the involvement of the client. When two parties are involving in any deal we cannot target any one party even that party is more responsible for maintaining all technical parameters. If clear requirement is not documented, it will surely effect the entire development. Analysis phase starts with providing basic information from

the client side or replying the query of the analyst by any way this phase start with the involvement with client. Both the parties need to pay proper attention for successful completion of this phase which ultimately turns in to a base factor of the quality software. Focusing on the small business software many parameters is not considered by software developing companies that majorly affects the quality of the software. Even lack of proper documentation is also noticed during the entire review.

References:

- Pressman, Roger S. Software engineering: a practitioner's approach / Roger S. Pressman. —5th ed. p. cm.— (McGraw- Hill series in computer science)
- J. Radatz, A. Geraci, and F. Katki, IEEE standard glossary of software engineering terminology, IEEE Std. 610.12-1990, 1990.
- N. S. Godbole, Software quality assurance: Principles and practice, Alpha Science Int"l Ltd., 2004.
- B. Singh, Quality Control and Reliability Analysis, Khanna Publishers, 2011.
- https://www.synapseindia.com/
- https://www.roberthalf.com.au/blog/employers/6-basic-sdlc-methodologies-which-one-best
- M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, CA, 1979.
- http://www.comp.lancs.ac.uk/projects/RenaissanceWeb/
- https://www.testlodge.com/
- https://en.wikibooks.org/
- https://www.innovativearchitects.com/
- https://doit.maryland.gov/
- Mandal, A., & Pal, S. C. (2013). Investigating and Analyzing the Desired Characteristics of Software Development Lifecycle (SDLC) Models. International Journal of Software Engineering Research & Practices, 9-15.]
- Yadav, H. B., & Yadav, D. K. (2015). A fuzzy logic based approach for phase-wise software defects prediction using software metrics. Information and Software Technology, 63, 44-57

- Mohan, P., Shankar, A. U., & JayaSriDevi, K. (2012). Quality flaws: Issues and challenges in software development. *Computer Engineering and Intelligent Systems*, *3*(12), 40-48.
- Owens, Dawn &Khazanchi, Deepak. (2009).Software Quality Assurance.((PDF) Software Quality Assurance (researchgate.net)).
- Retna, J. E., Varghese, G., Soosaiya, M., & Joseph, S. (2010). A study on quality parameters of software and the metrics for evaluation. International Journal of Computer Engineering & Technology (IJCET), 1(1), 235-249.