

# Requirements Engineering Techniques in Software Development Life Cycle Methods: A Systematic Literature Review

Adetoba Bolaji. T, Ogundele Israel. O

**Abstract—** Requirements engineering (RE) is the comprehensiveness of user requirement. It comprises of all processes of life-cycle in identification, specification, analysis, development, validation and evaluation of the requirement. This ensures that the system satisfy the requirement of the customer. Software Development Lifecycle (SDLC) aim to produce a high-quality systems that meets customer expectations, delivered within specific time, able to minimize cost and maximize profit. The SDLC methods considered literature on using Spiral Method, Waterfall Method, Verification and Validation (V-Model) and Rapid Application Development (RAD). However, researcher has not been able to select and review literature that focus on this research work. This work uses method that is proposed by Kitchenham to conduct a Systematic Literature Review (SLR). Some of the reviewed papers considered are from Research Gate, Science Direct, Springer, Institute of Electrical and Electronics Engineers (IEEE), Association of Computing Machinery (ACM) and World Wide Web (WWW). 385 published papers were identified, 27 of the papers are relevant to RE used in SDLC which provide information about the topic. The paper systematically review literatures that will help researchers and practitioners to better understand RE techniques that are used in SDLC.

**Index Terms—**Requirement engineering, SDLC, Systematic literature Review, RE Technique.

## I. INTRODUCTION

Requirements are foundation of every projects, it specify the user requirement and what must be done to satisfy that the requirement. It is generally written in natural language for the user to understand, it captures the need and challenges to drive the processes of activities. Requirement enable the management of risks from the first stage of development. The risk discovered can be seen, their impact assessed, limitation plans understand. The three types of requirement are functional requirements, Non-functional requirement and domain requirements.

Functional requirements describe the functions the system should perform and the input it will take. It should be able to explain the behavior of what the system will do and the limitation. Functional requirements can be detailed to express the input, processes, output and exceptions. Non- functional requirements (NFRs) focus on the external part (i.e the environment) where the system is going to be used. NFRs includes performance, security, cost, interoperability, flexibility, accessibility. Domain requirements specify the computation that should be performed on the system, this is in terms of functional requirement. It is the application

domain looking at the constraints of the existing system to develop the new system.

Requirements engineering is a field of systems engineering that is concern with the process of software requirement from customer to analyze and document. It is also concerned with specific requirement of the customer, design, development and to the generation of the system over time [1]. The requirement engineering causes an improvement to the front-end of the system, it arises the need for feasibility study, requirement gathering, software requirement specification and validation. Requirement engineering is the first stage of the process of software engineering, it is the most essential task that must be performed in software development [2]-[3]. Unclear requirement can result to software failure and software product delivered will be defective [4]. Likewise, an effective requirement will lead to success of the software system which equally increase productivity and quality product. [5]. The quality of a system is a function of the quality of the requirement to meet the user's expectation [6]. Requirement engineering processes is the most important in the development of software systems [7]-[9]. It has the capabilities on the overall product of the systems. Requirement engineering is complex and critical phase because of the multidisciplinary role of requirement engineering process and the set of product demanded from practitioners (i.e stakeholders). Today, there are some number of requirement methodologies and techniques that are available for practitioner. However, choosing the most appropriate techniques to develop a system is critical to the quality of the system because a techniques that is used for one system, might not be suitable for another. The change in the environment may affect the requirement engineering process. When time to market a particular software is of essence than quality of the software product thereby leading to system failure. Requirement engineering should model a guidelines to practitioners that ensures quality software production within a specific time that eventually lead a software product to success while avoiding old-fashion of requirement engineering problems. Several research work has put more attention on requirement engineering practices because of the future application.

SDLC consists of various stages in the development, design, testing, deployment and maintenance of the software and to ensure that the development system satisfy the expectation of the customer [10]. The SDLC aim in the delivery of quality software product within the set time that meet customer's expectation. It is a framework that defines each stages and steps involves in the software development processes. The processes that are followed to develop software product within a software organization are the methodologies of SDLC. These methodologies assist to deliver quality software product based on the customer requirement. The SDLC models are followed in the development of the software, each of the processes follows series of unique steps

to ensure that the system do not fail in the software development.

The SDLC methods considered literature on using Spiral Method, Waterfall Method, V-Model and RAD. However, researcher has not been able to select and review literature that focus on this research work. This work uses method that is proposed by Kitchenham to conduct a Systematic Literature Review (SLR). Some of the reviewed papers considered are from Research Gate, Science Direct, Springer, IEEE, ACM and WWW. 385 searched published papers where identified, 27 of the papers are relevant to RE used in SDLC which provide information about the topic.

The research paper present a SLR with regards to RE techniques that are used in SDLC in order to be able to critically analyze current use of the techniques in satisfying customer's expectation. A SLR is the process of identifying, evaluating and understanding all the variable research that is relevant to the research question, area of study and field of interest. The paper systematically review literatures that will help researchers and practitioners to better understand RE techniques that are used in SDLC.

## II. REQUIREMENT ENGINEERING IN SDLC

This section presents an introduction on Software Development Life Cycle (SDLC) models and the concept of requirement engineering.

The SDLC methods discussed here are: Spiral Method, Waterfall Method, V-Model and RAD. The most popularly used is the spiral method explained by [11]. Spiral methods consists of the understanding of iterative development and linear development model. Spiral methods consists of four stages.

- i. Identification: requirement are gathered at the foundation level as the project advances. At identification stage the system requirements, subsystem requirements and unit requirements are considered. The understanding of the system requirements is crucial at this stage to develop the system.
- ii. Design: it involves the conceptual design of the spiral model. It consists of architectural design, physical product design and logical design.
- iii. Build: the build stage refers to how the real software product will look like at every spiral. Customers response is gotten based on the proof of concept.
- iv. Evaluation and Risk Analysis: this includes the identification, cost/benefit, risks, feasibility and estimation. The customer evaluate the software and provide response to improve on the development.

Requirements Engineering (RE) is the process of establishing the services that the customer needs from a system and the limitations to its operation [12]. Different methods have been employed in RE activities [13], and these activities differs from each other depending on the environment where the software will be used and the company documenting the requirements. RE stages are elicitation, analysis, specification, validation and management of the software [14]. These are explained below:

- i. Elicitation: elicitation aim to deliver what problem is to be solved, identify the objectives to be achieved and stakeholders that will be involved [15]. It attempt to find out the requirement for the propose software systems. Requirement elicitation consists of the following processes; requirement gathering, requirement

organization, negotiation/discussion and documentation. The techniques used for requirement elicitation are interviews, surveys, questionnaire, task analysis, domain analysis, brainstorming, prototyping and observation [16].

- ii. Analysis: involves developing conceptual models to achieve the whole requirement development of the system. This helps to understand organizational structure, the rules and data necessary for the system.
- iii. Specification: in this requirement process, two types of document are necessary; user and system requirement. It's the process of writing down information gathered from elicitation and analysis through the use of the techniques of templates, scenarios, use case and natural language [17].
- iv. Validation: it checks the requirement for consistency and completeness. This ensures a clean and clear requirement document that is free from errors. Some techniques used are reviews and traceability [18].
- v. Management: management recognized changing requirement of the software. The system should be flexible to handle continuous change in the environment as a result of customer's demand for new software product. These should include techniques for configuration management and control [19].

In considering requirement engineering techniques, we have the following: (i) Interviews (ii) Brainstorming, (iii) Prototyping (iv) Questionnaire (v) Documentation Study (vi) On-site observation (vii) Use Cases (viii) Scenarios (ix) Focus Groups (x) Natural Language and (xi) Form of contract.

**Interviews.** Interview is the face to face interaction with the intended user and stakeholders of the system. This is a strong methods of collecting requirement. Interview can be oral, structured, Non-structured and written. One of the benefits of the interview is that it helps the developer to get a good collection of information and one of the discouraging factor is that, analyzing the large data is tedious because different users may provide confusing information [18].

**Brainstorming:** it's a group meeting that aim to contribution to the development of the software by good criticism and judgement. At this point, a wide variety of views to see problem that exist and how to further provide solution. This is mostly applied at the starting of the requirement engineering process [20]. Brainstorming is of disadvantage because addressing major issues might be difficult, due to several opinions of stakeholders [21].

**Prototyping.** Prototyping is a useful technique to develop user interface without detail functionality for the software. The development can creates a prototype, if there is no software installed at client's side for developer's view and the clients doesn't understand its own requirement. It assist to give a good idea and understanding of requirement. The client's response is used as an input for requirement information. One of the disadvantages of this techniques is that user resist changes because of the frequent use of the system overtime. Cost and time also discourage stakeholders [21].

**Questionnaire:** The questionnaire is a method by which a document is prepared with a pre-defined objectives questions for user to provide answer to them. It uses lesser time and cost to get information about the system. To get accurate information, the structure of the document should be clear and understood to obtain good user requirements. The fallback is that if an option is not

mentioned in the document, the issue might not be addressed [21].

**Documentation study:** this study provides a depth reading on the application domain of the system to be developed. The documents shows the user guide, installation guide, procedure and functions. This are analyzed to deal with area related to the objectives of the organization.

**On-site observation:** this is the observation of the standard practices that are employed and used in the development of the software that present a high quality product.

**Use cases:** use cases define its requirement by the trends of event. It inform the stakeholders by storytelling. They are easy to use and understand to validate requirement [21].

**Scenarios:** requirement define the action and communication with users and stakeholder. This is important to validate requirement with test cases [21].

**Focus Groups:** This technique is applied to requirement elicitation that helps to resolve issues between stakeholders looking at all areas applicable with better suggestion by the members of the group. Disadvantage of the focus group is that it requires a lot of effort and time in gathering stakeholders for meeting [21].

**Form of contract:** the requirement allows you to fill forms or contracts. It may be tedious when the project working on is large within a software development.

**Natural Language:** most application domain makes use of natural language to represent information. Requirement are written in natural language for the clients and stakeholders to understand the system. These is classified into two: (i) direct interaction with the user (ii) requirement elicitation using natural language. Natural language is most popular means of communication, it's a common language to both the clients and analyst. Disadvantages of the natural language is that is too complex [22].

From the above explained on requirement engineering techniques, so many benefits have been drawn in avoiding errors in other to develop a high-quality product. When the system is already development, it's difficult to amend and very expensive. The major focus of this is to develop a high quality software product, with minimal cost that deliver within a short time.

### III. SYSTEMATIC LITERATURE REVIEW

Systematic Literature Review (SLR) comprises of processes involves in the development of a review, identification of the study, selection of the study, extraction of useful publication, analysis of the review and evaluating the results. The review work considered using a guidelines for conducting systematic literature review by [23]. The aim of this research is to review a requirement engineering techniques [24] used in SDLC specifically in software development processes and the activities involve such as elicitation [15], [13], specification [25], validation [26] and management [18] to pave way for future research work.

#### Research Questions

Kitchenham classified research questions into four: (i) population (ii) Intervention (iii) Comparison and (iv) outcomes. This classification is called PICO.

Population is to deal with people involve, projects and application types that are influence by the intervention. Intervention is the tools, technology infrastructure and the methods that generates the results. Comparison means some other intervention that may be applicable while Outcome is the technological effect on the significant information.

The PICO has been employed for our research and is described here as:

- Population: Population is a software engineering duty (e.g stakeholders, analyst) and software engineer type (e.g an unexperienced and experienced engineer).
- Intervention: are tools and equipment used such as Information Technology (IT) systems and control systems. Technology used to perform different task in the development of the software
- Comparison: not considered
- Outcomes: considered criteria that are of importance to the user and stakeholders such as security of the system, increase in reliability, reduced cost and time and deliver higher functionality. It also determine whether or not it support requirement engineering techniques.

The review study aim to provide solution to the two research questions.

- **RQ1** – What software development methods exist in applying traditional RE techniques in their SDLC?
- **RQ2** – What requirement engineering (RE) techniques are mostly applied in SDLC activities to obtain user's needs?

#### Search Strategy

The search strategy is systematic. We have earlier define our goals and research questions, next is to formally search materials that are related to the research review work.

Google search engine and google scholar (conference, journal and workshop) with advance search will be used by applying different keywords and terms coined from both research questions considered. The sources of the research are publications from secured access namely: Research Gate, Science Direct, Springer, IEEE, ACM and WWW. These libraries of publication were choosing because they are relevant to the topic in study. Furthermore, we also contacted relevant publications of the authors of different research paper identified in Digital Bibliographic Library browser (DBLP).

#### Inclusion and Exclusion Criteria

The criteria determine the study to be included or excluded.

**Inclusion criteria:** The inclusion criteria only include publications that are relevant to requirement engineering activities that are used in SDLC methods. The study is in English and are related to only the techniques; it is relevant to the search terms. The inclusion criteria has been choosing: (i) the study is published between, January, 2008 to August, 2018, (ii) requirement engineering development activities (iii) SDLC study and (iv) inclusion relevant to both research questions to achieve the objectives.

**Exclusion criteria:** exclusion criteria ignore studies that are not relevant to RE in SDLC methods such as topic that do not discuss requirement engineering (RE) or SDLC (ii) studies that does not part of inclusion criteria (iii) repeated document from the same research.

#### Research Quality Assessment

Research quality assessment is based on the place of publication and diffusion of the papers. This refers to where

the paper were published, it involves wide range of search while the diffusion of the papers refers to the methods corresponds to the academic or industry. We formulate some questions to the quality assessment in order to evaluate the relevance, credibility, completeness of the selected paper. The first search result gives a total of 48 documents without exclusion criteria and 3 of those document were duplicated. 27 of the document were relevant after applying the exclusion criteria and quality assessment were carried out to confirm the information extracted.

**Data Extraction**

The data extraction forms are prepared to collect vital information essential to address the research questions. The form is used to collect the information extracted from the search strategy storing the author’s name, research focus, the journal or conference where the paper was accessed, reviewed and published and the publication date. After quality assessment, data synthesized were performed. This was carried out by 27 total assessment of the relevant published papers shown in table 2.

IV. DATA ANALYSIS

Data analysis is the process of organizing and selecting the results of the search strategy that are included. Data synthesis can involve descriptive with quantitative conclusion which are specify in the review. In other words, data are analyzed to resolve issues that are arise as a result of the review. The SDLC methods considered for this work are: Spiral Method, Waterfall Method, V-Model and Rapid Application Development (RAD). The included studies gives a full proof in providing solution to the research questions (RQs) as explained below:

**RQ1 – What software development methods exist in applying traditional RE techniques in their SDLC?**

For Spiral Method, Waterfall Method, V-Model and Rapid Application Development (RAD) methods exists in applying traditional RE techniques in the software development processes, the authors apply different traditional techniques. In the work presented by [27], the authors present in their work Comparison between waterfall model, V-shaped model, iteration model, spiral model and extreme model. In the work presented by [28], the authors considered different requirement elicitation techniques, such as prototyping, questionnaire, use-cases, brainstorming, modelling etc. In the research carried out by [29] focuses on agile and waterfall model which uses the techniques of interview, brainstorming and phototyping, another methods that include Xtreme programming, feature driven Development, uses Use-Case, interview, brainstorming and feature list extraction

Here, we have considered four SDLC methodologies for detail of the work but there are other traditional SDLC methods that are not considered such as agile model, iterative model, test driven development (TDD), agile development, Lean software development etc this methodologies also make use of requirement elicitation techniques.

**RQ2 – What requirement engineering (RE) techniques are mostly applied in SDLC activities to obtain user’s needs?**

RE techniques applied in Software Development Life Cycle (SDLC) methods. [30] did a research on engineering tools and how to select the particular tool and also provide different types of tools such as interview, surveys, card sorting, repertory grids, Card responsibility collaboration,

questionnaires, task analysis, domain analysis, brainstorming, Joint Application Development (JAD), observation, protocol analysis, prototyping analysis etc.[27] the author represents five of the development models research on requirement Engineering tools such as waterfall, Iteration, V-shaped, spiral and Extreme programming. It presents four stages of descriptive process of identification, design, validation and evolution. The modification of the waterfall model can be done using JADs, prototyping, brainstorming or other methods can be [31]. [32] Researched on Requirement Engineering in Rapid Application Development.

A better understanding of the domain helps to reduce failure and guide against risk when a standard practices is followed. JAD can consist of 20 to 30 stakeholders from different domain to better understand and explain their requirements. Some of the techniques that applied RAD are JAD, prototyping, domain analysis etc. [33] discussed the spiral model involving the analyses and development of the system, such as using use-case model that identify the requirements.

**Table 1: RE techniques in SDLC methods**

RE in Traditional Methods	RE techniques in SDLC
Interview	Waterfall Model, RAD
Use case	Spiral
Prototyping	Waterfall, RAD, Spiral
Priotization	RAD
Analysis Meetings	V-Model, Spiral
Social Analysis	Waterfall
Workshop	RAD
Domain Analysis	RAD
Reviews	Waterfall, RAD, Spiral, V-Model
Software Requirement Specification	Waterfall, V-Model, Spiral, RAD
Modeling	RAD
Card Sorting	RAD
Group Work	Waterfall
Joint Application Design (JAD)	Waterfall, RAD, Win win Approach

V. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) METHODS ANALYSIS

SDLC is an approach or methodology that are used in the development of software. This consists of several stages from the problem definition stage, feasibility stage, analysis stage, design stage, implementation stage, deployment stage, down to maintenance stage. There are different model for SDLC based on its concept [34], SDLC is a model that are followed to develop a software.

Requirement satisfaction is essential to the development of software. Most software generate errors during the RE stage. Correction of errors after the development of the software is expensive and to guide against such, organization must adhere to the software development process by ensuring that software development team to do this if they want their software product to be more efficient and effective. The execution of the Capacity Maturity Model Integration (CMMI) needs a documented and formalized RE within organization [14].

The choice of model choosing determine the development strategies that will be employed by the software developers. Each of the model chosen has its own benefits and drawback because of influence of model characteristics. Comparison of

the various model is of essence in choosing the best to use for a particular problem. This research consider the following model; Waterfall Model, Spiral Model, V-Shaped Model, and Rapid Application Development (RAD). SDLC comprises of different stages which are explained below:

**Feasibility Analysis:** this is the first phase of the SDLC. It determine the input data, processing required and the output data. It also analyzed the cost/benefit of the project, term of reference. Feasibility study is done at this stage to know the components of computer needed and personnel that will be involved to do the work. Feasibility report is generated once this stage is completed.

**Requirement Analysis and Specification:** at this stage, both the functional and non-functional requirement are gathered and evaluated by the developer. Software Requirement Specification (SRS) document are generated.

**Designing:** this stage gives a blueprint on how the system will look like, if developed. The SRS is seen in a logical structure for visualization. The design document is created.

**Coding:** the programmer works on this stage by inputting some syntax and function to activate the design. The design document is used, which is translated by the programmer to source code.

**Testing:** in this stage, we use various tools such as test cases, tester to test the source code in order to evaluate the correctness and validity of the software.

**Implementation:** the system is implemented and deployed for use. The system is released by the developer to user and if any error is generated in the process, it is reported to the system analyst.

**Maintenance:** this stage is necessary to eliminate error that are encountered during its working life. This also calls for review of the system from time to time. It ensures proper working of the system within the environments.

**Table 2:** Review of RE Techniques in Software Development life Cycle Methods

Author's	Research focus	Publication	Year
[35]	The research work is a knowledge based system that uses agile methodology based on ontology to integrate different models of users and developer.	Advances in Intelligent Systems and Computing	2018
[32]	The research used agile practice rather than agile method for assessment model. The focus is on usability of some specific agile method to increase speed and productivity	International Journal Of Research In Computer Applications And Robotics	2018
[36]	They did a review on the need for	International Journal of Scientific	2017

	requirement engineering Rapid Application Development (RAD). Such system is effective as it relate to cost, time, reducing errors and risks	Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)	
[37]	The work compared three methods (i.e V-Shaped Model, Parallel Development Model, and Iterative Model) of software development in SDLC. This provides an understanding to software developer in selecting the best method to use.	International Journal of Computer Applications (0975 – 8887)	2017
[38]	They did a review work on various aspect of security in each of the SDLC stages and also performance.	Advances in Computational Sciences and Technology	2017
[39]	A detailed analysis of different SDLC method were considered. Advantages and disadvantages of the three SDLC model were analyzed to know the strength and weakness of each of them.	International Research Journal of Engineering and Technology (IRJET)	2017
[40]	They research on the various models of SDLC and how they perform their activities for better understand of the developer and stakeholders	International Journal Of Engineering And Computer Science	2017
[41]	They researchers conducted a survey that integrate both V-model and agile	International Journal of Management, IT & Engineering	2017

	methodology for impact in the industry		
[42]	The research conducted an analysis on five different software development life cycle models to see the strengths and weaknesses of each.	International Journal of Computer Applications	2016
[43]	They did a comprehensive study and analysis on SDLC model. They classified all model into three such Traditional models, Agile models and Hybrid models. They gave an understanding on how practitioners can chose the best model for practice.	Association of Computing Machinery (ACM). Software Engineering Notes	2016
[44]	The research gave a solution to the problem of choosing appropriate techniques by comparison between requirement elicitation techniques. The researchers used online survey and interview	International Journal of Scientific & Engineering Research	2015
[45]	The paper gave an introduction on quality assured model-driven requirements engineering and software development methods and also develop a tool that can automatically generate several engineering process.	Advance Access publication	2015
[46]	The paper focus is on comparison	International Journal of	2015

	and consequences of SDLC models. They compares five software development model looking at the benefit and fallback.	Computer Science and Information Technologies. Vanshika Rastogi / (IJCSIT)	
[47]	The researchers identifies and analyzes various aspects of conventional and Web-based development and proposes a lifecycle model, which incorporates the aspects of agile and plan-driven development to develop a Web-based application successfully.	International Journal of Computer Applications	2015
[48]	The paper focus on the following: (1) Review and comparison an existing Open Source Software Life Cycle (OSSLC) with propose software development by researchers (ii) presented a various development approach that are used in case of OSS (iii) to understand how the development of OSS begins and advances.	International Journal of Software Engineering and Its Applications	2014
[49]	The paper reviewed current challenges and issues that are specific in the use of requirement engineering to a software method. They also reviewed the need for standardization of RE activities.	International Journal of Computer Applications	2014
[50]	The paper	International	2014

	reviewed an understanding on elicitation techniques that tailored towards the need of the stakeholders.	Journal of Information and Computation Technology.	
[51]	The researchers did a systematic literature review work that analyzed existing research on variability in software systems.	IEEE Transactions on Software Engineering	2014
[52]	The researchers propose an effective requirement engineering (RE) process model to generate appropriate, accurate, consistent requirements.	Global Journal of Multidisciplinary Studies (Research gate)	2014
[53]	They conducted a systematic literature review (SLR) for the understanding of SE techniques by researcher and stakeholders	Journal of Information and Software Technology	2013
[54]	The work gave an in-depth study on the problems that are faced in RE, processes of RE and selection of appropriate model.	International Journal of Engineering Science and Technology (IJEST)	2012
[55]	The research focused on the comparative analysis of different SDLC models and on how organization can easily convert from manual to automatic following the right procedures.	Journal of Engineering (IOSRJEN)	2012
[56]	The paper gave an understanding on an effective RE methods and their important in different stages of SDLC	Journal of Emerging Trends in Computing and Information Sciences	2012

[57]	This paper focus on the four stages of software life cycle, different techniques that are used and methods of software development life cycle	International Journal of Computer Science and Information Technologies (IJCSIT).	2012
[58]	They research on requirement elicitation methods and its use in SDLC processes.	Recent Researches in Automatic Control (Research gate)	2011
[59]	This paper introduces some of the fundamental aspects of value from the perspectives of economic theory and define value for business and product.	International Journal of Computer Science and Network Security(IJCS NS)-Research gate	2010
[30]	The researchers work focus on the following such as requirement engineering tools, technologies used, tools employed and methodologies applied. They also looks into the benefits, limitation and also recommend for organization to automate their RE processes.	International Journal of Reviews in Computing	2009

## VI. REQUIREMENTS ENGINEERING PROBLEMS

Major problems that are related with requirement engineering (RE) process are connected with elicitation requirements and analysis requirements [60]. Practitioner's involvement in both requirement engineering cause of the problems. This paper therefore, present usual problems associated with elicitation and analysis requirements.

### Elicitation Problems

The requirement elicitation is highly tedious and consume time, although it appears to be a simple method [61]. It involves stakeholders to meet and interact on the requirement changes. Examples of requirement elicitation used are prototyping and meeting [62]. This techniques offers great solution but have some tradeoffs.

The effectiveness of one requirement elicitation technique might not be applicable to the others because different

techniques requires a different guidelines [24] and this are easily amendable for practitioner's [63]. Wrong selection of elicitation technique will result to low quality product and this is due to poor understanding of the [64].

Changes in the business environment and trends of technology applied, makes software product to deliver at limited time frame and this causes the software development life cycle and requirement processes to be shorten [65]. Some researchers have identified problems that occur during requirement elicitation processes [66]-[67].

Some other usual problems related to elicitation are volatility, limitation and understanding [61].

#### **Analysis requirement Problems**

Incorrect, inconsistent and duplicate are usually presented in elicitation phase and using manual approach to eliminate this problems can be time taken. Automatic elimination of these incorrect, inconsistent and duplicate will help or if systematically formalized. We can apply the same automatic elimination to remove anomalies [68]. Stakeholders are also involve in this stage and some of the major problems are having different views, perspective, goal, concern and responsibilities of stakeholders [69].

### VII. ISSUES AND CHALLENGES OF RE IN TRADITIONAL SOFTWARE DEVELOPMENT

The traditional methodologies of software development necessitates the user to provide full clue of the right requirements with regards to expected system. The methodology have a comprehensive requirements model as a term of reference for the development, coding and testing to be used by the development team. The development team will analyst the system until it's fully developed into a software product [70].

The issues and challenges of the traditional software development are explained below:

#### **Addressing NFR (Non Functional Requirement)**

Traditional software development focus lesser attention to NFRs. Some of the NFRs includes portability, quality, security, operability etc. different domain application are peculiar with their own attributes. An interface can be developed to collect attributes from various user in terms of what is expected.

#### **Poor Requirement Traceability**

Tracing requirement with its design is always a tedious task. Most project employ the manual approach to mapped requirement with design and the architecture of it is tasking even when tools are used. When requirement is not properly traced with the design, actual changes might not be effected. The traceable tool is important in order to trace appropriately requirement processes.

#### **Immutable Requirements**

Immutable requirements as to do with putting a stop to the work when all the stakeholders have agree on the system to develop. Customers might change requirements of the software development at any time due to change in environment. Requirement will not change once the specification phase has been concluded. The problem of immutable requirements can be solved by using iterative software engineering models.

#### **Elicitation End Point**

Elicitation process can end or continue even after software product delivery depending on the model considered. There are lot of challenges been faced in deciding when elicitation

should end in traditional software development. The challenges can be solved if all stakeholders decide the freezing point for elicitation process.

#### **Business Agility**

Business agility is the process by which business requirement changes to gain strategic benefit in terms of competition and response of the system. The changes of business requirement is unpredictable. The traditional development can be outdated, irrelevant and obsolete with time due to the changing environment and business processes [71]. To solve this problem agile methodology can be used.

#### **Customer Involvement**

In traditional approach, the need and requirement of the customer are given at elicitation phase. Customer needs are volumes of requirement by the stakeholders because the participation of the customer is partial to elicitation and specification. The aim of the customer as it relate to the efficiency and effectiveness may not be accommodated in the transitional phase of development. Less involvement of customers will help minimize the problems.

Some of the other issues and challenges of requirement engineering (RE) are improvement of requirement quality, No standard for RE activities, Elicitation techniques selection, Conflicting views of the teams, effectiveness and performance measurement of requirement, Prioritization of requirements etc.

### VIII. SDLC MODELS CONSIDERED

**Waterfall Model:** waterfall model is one of the earliest SDLC model. It is a model one phase has to be done before proceeding to another. The model is easier to use and managed. Sometimes this model might waste time because you can't proceed to another stage, if one phase has not been completed. It follows a sequential phases of requirement analysis, system design, implementation, testing, deployment and maintenance. One of the advantage is that phases are executed and completed at once while one disadvantages is that it does not give much room for reflection and revision.

**V-Model:** this model is also called verification and validation model. Execution of processes occurs in a sequential manner. This is a highly formalized model and the next phase execute only when the previous phase has been completed. There are various phases in the V-Model such as business requirement analysis, system design, validation, system testing, and acceptance testing etc. the V-model is closely related to waterfall model because it's both sequential. One good advantage of the model is that is highly-formalized and phases are completed one at a time but one disadvantage is that is have high risk and uncertainty.

**Spiral model** it combines the iterative development process and sequential linear development model. The release of the software product is done incrementally. The spiral model consist of four stages: (i) identification, (ii) design, (iii) construct and (iv) evaluation and risk analysis. It is used in a software organization and integrated with natural development of the software product. Spiral model accommodate changing environment and allow full use of the prototype techniques as its advantages. Its disadvantages is that spiral model is more complex to manage and the process can go on indefinitely.

**RAD Model:** The RAD means Rapid Application Development. RAD is the process planning to write a software product. It uses prototyping and iterative techniques



for software development. The teams used in RAD project are developers, domain experts, customer representatives and other IT personnel. RAD Model design follows different phases such as business modelling, data modelling, process modelling, application generation, testing and turnover. The customer might need some changes after the product have been delivered and the change effect in RAD is very rigid. Some of the few advantages are: lesser development time, customer feedback is encouraged and accommodate changing requirement. Some disadvantages are: required a skilled developers, requires user's involvement in all the stages of development and can only develop systems that are modularized.

#### IX. CONCLUSION AND FUTURE WORK

The research work shows a systematic literature review on the Requirements Engineering Techniques in Software Development Life Cycle Methods. The review guidelines used is explained by [72] to understand different researches on Software Development of Life Cycle methodologies and practices. In the review we use criteria inclusion and exclusion to choose the study that are relevant to the research question and neglect the one that are not relevant. The SDLC methods considered are Waterfall Model, Spiral Model, and V-Model and RAD model. 385 published papers were identified and accessed through electronic search engine and out of these papers, 27 of the papers were relevant to our research questions for quality of the proof they give and further analyzed. RE techniques are used to obtain user's requirement in SDLC processes, moreover, only four methodologies of the SDLC relating to RE were considered in this review work. For future research, a systematic literature review on requirement engineering in other SDLC methods not considered in this work, such as iterative model, Big Bang model, agile model etc. in this view, there is a need for research work that compares various RE techniques in software development and traditional software development. In addressing some of the problems of RE in traditional software development much attention should be given to tool support and technologies to address scalability and popularization. In conclusion, it is difficult to elicitate all user's requirement at the starting of the SDLC project. The Software RE practices like customer involvement, analyzing meeting, software requirement specification, modeling, Joint Application Design (RAD) are specific attributes that are not included in the traditional software development with requirement. Furthermore, the studies of the research questions were properly addressed. We have highlighted some of the issues and challenges of RE in Traditional Software Development of adopting SDLC models.

#### REFERENCES

- [1] A. Affleck and A. Krishna, "Supporting quantitative reasoning of non-functional requirements: A process-oriented approach," in *Proceedings of the International Conference on Software and System Process*, 2012, pp. 88–92.
- [2] D. A. D. Shinde, "Requirement Engineering: How To Make It Complete And Correct," *Int. Res. J. Comput. Sci.*, vol. 4, pp. 12–15, 2017.
- [3] S. Asghar and M. Umar, "Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components," *Int. J. Softw. Eng.*, vol. 1, no. 1, pp. 32–50, 2010.
- [4] H. F. Hofmann and F. Lehner, "Requirements engineering as a success factor in software projects," *IEEE Softw.*, no. 4, pp. 58–66, 2001.
- [5] D. M. Fernández, K. Lochmann, B. Penzenstadler, and S. Wagner, "A case study on the application of an artefact-based requirements engineering approach," in *Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on*, 2011, pp. 104–113.
- [6] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," *Inf. Softw. Technol.*, vol. 58, pp. 148–169, 2015.
- [7] D. Pandey, U. Suman, and A. K. Ramani, "Role of Interviews in Requirements Gathering," in *Proc. of National Conference on Intelligent Computing and Information System, DP Vipra College, Bilaspur (CG)*, 2009, pp. 26–31.
- [8] N. Juristo, A. Moreno, and A. Silva, "Is the European industry moving toward solving requirements engineering problems?," *IEEE Softw.*, no. 6, pp. 70–77, 2002.
- [9] S. Komi-Sirviö and M. Tihinen, "Great Challenges and Opportunities of Distributed Software Development-An Industrial Survey.," in *SEKE*, 2003, pp. 489–496.
- [10] T. J. Lehman and A. Sharma, "Software development as a service: agile experiences," in *SRII Global Conference (SRII), 2011 Annual*, 2011, pp. 749–758.
- [11] D. M. Buede and W. D. Miller, *The engineering design of systems: models and methods*. John Wiley & Sons, 2016.
- [12] A. De Lucia and A. Qusef, "Requirements engineering in agile software development," *J. Emerg. Technol. web Intell.*, vol. 2, no. 3, pp. 212–220, 2010.
- [13] E. Hull, K. Jackson, and J. Dick, "Defining requirement engineering," *Requir. Eng. Springer Sci. Bus. Media, London*, pp. 6–8, 2010.
- [14] O. Dieste Tubio, M. López, and F. Ramos, "Updating a Systematic Review about Selection of Software Requirements Elicitation Techniques," 2009.
- [15] M. A. Ramdhani, D. S. Maylawati, A. S. Amin, and H. Aulawi, "Requirements Elicitation in Software Engineering," *Int. J. Eng. Technol.*, vol. 7, no. 2.19, pp. 772–775, 2018.
- [16] M. Yousuf and M. Asger, "Comparison of various requirements elicitation techniques," *Int. J. Comput. Appl.*, vol. 116, no. 4, 2015.
- [17] M. Rudolph, D. Feth, J. Doerr, and J. Spilker, "Requirements Elicitation and Derivation of Security Policy Templates—An Industrial Case Study," in *Requirements Engineering Conference (RE), 2016 IEEE 24th International*, 2016, pp. 283–292.
- [18] M. B. Chrissis, M. Konrad, and S. Shrum, *CMMI for development: guidelines for process integration and product improvement*. Pearson Education, 2011.
- [19] M. Wolski, B. Walter, S. KUPIŃSKI, and J. Chojnacki, "Software quality model for a research-driven organization—An experience report," *J. Softw. Evol. Process*, vol. 30, no. 5, p. e1911, 2018.
- [20] C. Pacheco, I. García, and M. Reyes, "Requirements elicitation techniques: a systematic literature review

- based on the maturity of the techniques,” *IET Softw.*, 2018.
- [21] T. ur Rehman, M. N. A. Khan, and N. Riaz, “Analysis of requirement engineering processes, tools/techniques and methodologies,” *Int. J. Inf. Technol. Comput. Sci.*, vol. 5, no. 3, p. 40, 2013.
- [22] K. Pohl, *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant*. Rocky Nook, Inc., 2016.
- [23] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering. 65 Google Scholar,” 2007.
- [24] B. Nuseibeh and S. Easterbrook, “Requirements engineering: a roadmap,” in *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 35–46.
- [25] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “Agile software development methods: Review and analysis,” *arXiv Prepr. arXiv1709.08439*, 2017.
- [26] A. Fuggetta and E. Di Nitto, “Software process,” in *Proceedings of the on Future of Software Engineering*, 2014, pp. 1–12.
- [27] N. M. A. Munassar and A. Govardhan, “Comparison study between traditional and object-oriented approaches to develop all projects in software engineering,” in *Software Engineering (MySEC), 2011 5th Malaysian Conference in*, 2011, pp. 48–54.
- [28] A. Rivera, D. Armando, and J. A. González Londoño, “La ingeniería de requisitos en las metodologías ágiles.” Universidad EAFIT, 2012.
- [29] L. Bormane, J. Gržibovska, S. Bērziša, and J. Grabis, “Impact of Requirements Elicitation Processes on Success of Information System Development Projects,” *Inf. Technol. Manag. Sci.*, vol. 19, no. 1, pp. 57–64, 2016.
- [30] Q. K. Shams-UI-Arif and S. A. K. Gahyyur, “Requirements engineering processes, tools/technologies, & methodologies,” *Int. J. Rev. Comput.*, vol. 2, no. 6, pp. 41–56, 2009.
- [31] J. J. Kuhl, “Project Lifecycle Models: How They Differ and When to Use Them,” 2002 *www.businessolutions.com*, 2002.
- [32] P. Maheshwaran, R. Kumar, S. Rajeswari, and J. Mungara, “A Review on Requirement Engineering in Rapid Application Development,” 2017.
- [33] A. Ayes, *Essential UMLTM Fast: Using SELECT Use Case Tool for Rapid Applications Development*. Springer Science & Business Media, 2002.
- [34] K. J. Knapp, T. E. Marshall, R. K. Rainer Jr, and F. N. Ford, “Information security effectiveness: Conceptualization and validation of a theory,” *Int. J. Inf. Secur. Priv.*, vol. 1, no. 2, pp. 37–60, 2007.
- [35] P. Ciancarini, A. Messina, F. Poggi, and D. Russo, “Agile knowledge engineering for mission critical software requirements,” in *Synergies Between Knowledge Engineering and Software Engineering*, Springer, 2018, pp. 151–171.
- [36] A. Shahmoradpour and S. Afraz, “A New Approach In Aspect-Oriented Requirement Engineering For Agile Software Development,” 2018.
- [37] S. Nugroho, S. H. Waluyo, and L. Hakim, “Comparative analysis of software development methods between Parallel, V-Shaped and Iterative,” *arXiv Prepr. arXiv1710.07014*, 2017.
- [38] A. Sharma and P. K. Misra, “Aspects of Enhancing Security in Software Development Life Cycle,” *Adv. Comput. Sci. Technol.*, vol. 10, no. 2, pp. 203–210, 2017.
- [39] H. S. Modi, N. K. Singh, and H. P. Chauhan, “Comprehensive Analysis of Software Development Life Cycle Models,” 2017.
- [40] S. B. A. S. U. Rani, “A detailed study of Software Development Life Cycle (SDLC) Models,” *Int. J. Eng. Comput. Sci.*, vol. 6, no. 7, 2017.
- [41] A. Mateen, M. Tabassum, and A. Rehan, “Combining agile with traditional V model for enhancement of maturity in software Development,” *arXiv Prepr. arXiv1702.00126*, 2017.
- [42] M. E. E. Daleel, “Software Engineering Development and Analysis of Life Cycle Models,” *Int. J. Comput. Appl.*, vol. 133, no. 10, 2016.
- [43] U. S. Shah, “An excursion to software development life cycle models: an old to ever-growing models,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 41, no. 1, pp. 1–6, 2016.
- [44] N. Sultana and N. Iqbal, “An Iterative Technique for Requirement Elicitation in Global Software Development.”
- [45] L. Lengyel *et al.*, “Quality Assured Model-Driven Requirements Engineering and Software Development,” *Comput. J.*, vol. 58, no. 11, pp. 3171–3186, 2015.
- [46] V. Rastogi, “Software Development Life Cycle Models-Comparison, Consequences,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 1, pp. 168–172, 2015.
- [47] N. Uikey and U. Suman, “A Lifecycle Model for Web-based Application Development: Incorporating Agile and Plan-driven Methodology,” 2015.
- [48] M. Saini and K. Kaur, “A review of open source software development life cycle models,” *Int. J. Softw. Eng. Its Appl.*, vol. 8, no. 3, pp. 417–434, 2014.
- [49] T. Shah and V. S. Patel, “A review of requirement engineering issues and challenges in various software development methods,” *Int. J. Comput. Appl.*, vol. 99, no. 15, pp. 36–45, 2014.
- [50] S. Khan<sup>1</sup>, A. B. Dulloo, and M. Verma, “Systematic review of requirement elicitation techniques.” India, 2014.
- [51] M. Galster, D. Weyns, D. Tofan, B. Michalik, and P. Avgeriou, “Variability in software systems—a systematic literature review,” *IEEE Trans. Softw. Eng.*, vol. 40, no. 3, pp. 282–306, 2014.
- [52] S. Sankhwar, V. Singh, and D. Pandey, “Requirement engineering paradigm,” *Glob. J. Multidiscip. Stud.*, vol. 3, no. 3, 2014.
- [53] H. Zhang and M. A. Babar, “Systematic reviews in software engineering: An empirical investigation,” *Inf. Softw. Technol.*, vol. 55, no. 7, pp. 1341–1354, 2013.
- [54] L. Ramingwong, “A review of requirements engineering processes, problems and models,” *Int. J. Eng. Sci. Technol.*, vol. 4, no. 6, 2012.
- [55] R. G. Sabale and A. R. Dani, “Comparative study of prototype model for software engineering with system development Life Cycle,” *IOSR J. Eng.*, vol. 2, no. 7, pp. 21–24, 2012.

- [56] A. Chakraborty, M. K. Baowaly, A. Arefin, and A. N. Bahar, "The role of requirement engineering in software development life cycle," *J. Emerg. trends Comput. Inf. Sci.*, vol. 3, no. 5, pp. 723–729, 2012.
- [57] S. T. Krishna, S. Sreekanth, K. Perumal, and K. R. K. Reddy, "Explore 10 Different Types of Software Development Process Models," *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 4, pp. 4580–4584, 2012.
- [58] R. Silhavy, P. Silhavy, and Z. Prokopova, "Requirements gathering methods in system engineering," *Recent Res. Autom. Control*, pp. 105–110, 2011.
- [59] S. M. N. Hasan, M. S. Hasan, A. Al Mahmood, and J. Alam, "A model for value based requirement engineering," *Int. J. Comput. Sci. Netw. Secur.*, vol. 10, no. 12, pp. 171–177, 2010.
- [60] R. Elamin and R. Osman, "Towards Requirements Reuse by Implementing Traceability in Agile Development," in *Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual*, 2017, vol. 2, pp. 431–436.
- [61] S. Nisar, M. Nawaz, and M. Sirshar, "Review analysis on requirement elicitation and its issues," *Int. J. Comput. Commun. Syst. Eng.(IJCCSE)*, vol. 2, no. 3, pp. 484–489, 2015.
- [62] L. Liu, T. Li, and F. Peng, "Why requirements engineering fails: A survey report from china," in *Requirements Engineering Conference (RE), 2010 18th IEEE International*, 2010, pp. 317–322.
- [63] C. Coulin, A. E. K. Sahraoui, and D. Zowghi, "Towards a collaborative and combinational approach to requirements elicitation within a systems engineering framework," in *Systems Engineering, 2005. ICSEng 2005. 18th International Conference on*, 2005, pp. 456–461.
- [64] S. Kausar, S. Tariq, S. Riaz, and A. Khanum, "Guidelines for the selection of elicitation techniques," in *Emerging Technologies (ICET), 2010 6th International Conference on*, 2010, pp. 265–269.
- [65] S. Thew and A. Sutcliffe, "Investigating the Role of Soft issues' in the RE Process," in *International Requirements Engineering, 2008. RE'08. 16th IEEE*, 2008, pp. 63–66.
- [66] D. Apshvalka, D. Donina, and M. Kirikova, "Understanding the problems of Requirements elicitation process: A human perspective," in *Information Systems Development*, Springer, 2009, pp. 211–223.
- [67] A. N. Rahman and S. Sahibuddin, "Extracting soft issues during requirements elicitation: A preliminary study," *Int. J. Inf. Electron. Eng.*, vol. 1, no. 2, p. 126, 2011.
- [68] M. Modarres, M. P. Kaminskiy, and V. Krivtsov, *Reliability engineering and risk analysis: a practical guide*. CRC press, 2016.
- [69] S. Ahmad, "Negotiation in the requirements elicitation and analysis process," in *Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on*, 2008, pp. 683–689.
- [70] M. Stoica, M. Mircea, and B. Ghilic-Micu, "Software Development: Agile vs. Traditional.," *Inform. Econ.*, vol. 17, no. 4, 2013.
- [71] W. Helmy, A. Kamel, and O. Hegazy, "Requirements engineering methodology in agile environment," *Int. J. Comput. Sci. Issues*, vol. 9, no. 5, p. 293, 2012.
- [72] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering. Engineering 2 (2007), 1051," *arXiv Prepr. arXiv1304.1186*, 2007.