

Process of Requirement Analysis Link to Software Development

L. K. Pulasthi Dhananjaya Gunawardhana 

Department of Information Communication Technology, Faculty of Technology, University of Sri Jayawardenapura, Nugegoda, Sri Lanka
Email: lkpdg87@gmail.com

How to cite this paper: Gunawardhana, L.K.P.D. (2019) Process of Requirement Analysis Link to Software Development. *Journal of Software Engineering and Applications*, 12, 406-422.
<https://doi.org/10.4236/jsea.2019.1210025>

Received: August 30, 2019

Accepted: October 20, 2019

Published: October 23, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper is giving an overview of the process of requirement analysis for software development. Here I have discussed about key parts in requirement analysing, gathering relevant materials, functional analysis and allocations, how to improve and make a quality process and also document development as well and many more which relates to requirement analysis process. The scope of this study is not a generalized approach but rather discuss through specific cases such as like Dutch flower case. It describes the main areas of requirement process in practice, and highlights. I hope that readers will find this paper useful in guiding them toward the knowledge and resources they needed.

Keywords

Requirement Analysis, Software Requirement Analysis, System Development Life Cycle, Requirement Engineering, System Analysis

1. Introduction

Requirement analysis is involved with the customer objectives and their needs. This is the background for the function of system in standard system attribute, which are determined requirements, environment and plan. The purposes of requirement analysis are to:

- Identify and define constraints which have limited solutions.
- Define requirements and customer objectives.
- Based on customer provided measures of effectiveness, define the performance requirements and functional, tools [1].

Requirement analysis should be studied along with the following aspects:

- Functions: what will do from the system.

- Performance: How well and how quick functions are being performed.
- Interface: Environment where system will perform.
- Other requirements

A good requirement analysis leads to a successful design definition. Those requirements of a project should be testable, measurable, and must be defined to a level of detail sufficient for the system design. Conceptually, there are 6 types of activities in the requirement analysis [2]. They are shown as below;

1) *Analysing Requirements*

Analysing requirement is, firmly deciding are the stated requirements unclear, resolving the problem, incomplete, etc.

2) *Eliciting Requirements*

Eliciting requirement is called as requirement gathering. What we do in here is, to communicate with the client and discuss and get to know what their requirements are.

Prototypes: This is used to clarify unclear tools.

Interviews: This method is used for selecting the requirements for the project.

Scenarios: From this method, it is easy to elicit user requirements.

3) *Recording Requirements*

This is the way we record client requirement. Generally, it is as a written document, which means using a set of forms that related to user scenarios or process specifications.

4) *Stakeholder Interviews*

This is a commonly used method in the requirement analysis. Using the interview method can reveal the requirements, but not as previously thought as being with project scope. Anyhow each stakeholder has visualized requirements.

5) *Prototypes*

Prototypes are the ones, which will help the customer to get an idea of what kind of software he/she is going to make. It's a kind of mock-up an application or visualizing about application structure, which is supported to be made. From prototype, it makes easier to communicate between developers and customer their targets.

Anyhow there are some issues that cannot be sorted out. They are mentioned below:

- Designers wasting their time for redesigning, they anxious to patch prototype codes.
- Prototypes are there to assist and give an idea of interface design and design decision. Except it would not help provide initial requirements for a certain project.
- Although designers and users focus their concentration to design interface, they gave less attention on producing a system, which will serve a business process.

6) *Use cases*

This is a technique, which has been used to document the requirements for new project or software changes. Every Use case contains scenarios. Since these

scenarios resolve to provide an inspiration how the system works with the end users. Stakeholders and requirement engineers are co-authors of the Use cases. Use cases are the ones not explaining internal workflow in a system, however it shows how users pursue performance task.

2. Requirements Analysis Input & Output

Inputs

Typical inputs are Environments, Missions, Customer needs and objectives, MOE/MOS, Key performance parameters, Suitability requirements and finally program-based requirements. Input requirements for both system process and product must be define and comprehensive. Deployment, Verifications, Operations, and Manufacturing, training, support and disposals are some of the items going with it [3].

Output

When we talking this topic, we have to divide topic in to 3 parts. Those are Operational, Functional and Physical views. It is for all intents and purposes co-ordinate to spot-out with customer needs and objectives [3]. Here below shows 3 topics of outputs in the requirement analysis:

1) Operational view

In here we discuss about how the user work with the system. Information in Operational views should be documented as to the concept of Operational concept document which will identify:

- System mission analysis
- Operational Environment
- Operational need definitions
- Operational sequence
- Operational interface with other systems
- Mission performance requirements
- Operational constraints on system

2) Functional view

From this topic it's focusing on what are the actions system takes to produce the required operational behaviours. Input-output statements and transformation rules, include to this functional views. Below mention things include to functional views:

- Inter-function affiliation
- Performance constraints
- System functions
- System Performance
- Relationship in between Software and Hardware
- Verification requirements
- Unique software and hardware

3) Physical view

From this topic is mainly focus on how to construct the system. Physical view is the key to build physical interface among technology requirements and op-

erators & its equipment. Normally Physical view includes information which will appear as below:

System Configuration

- Interface descriptions
- Relationship to operator to system
- Characteristic of information displays and operator controls

User's Characterization

- Constrains
- Handicaps

Limitations of System Physical

- Limitation of a technology
- Directed Standards
- Physical limitations

3. Types of Requirements

There are several types of requirements we can see when we do a project. Those requirements can be categorized in the following manners which are the requirements related to technical management.

Customer Requirements

There are certain statements of facts and assumptions that define these expectations of the system. These statements are usually related with respect to mission objectives, environment, and constraints of the system. These are the additionally comprise measures of effectiveness of the exact system and on how apposite the system is [4].

Functional Requirements

This is a component of a software. With this requirement we are able to describe as inputs, outputs and behaviour. What should be accomplished, what are the tasks that have to be necessarily identified are explained in functional requirements. In addition to the use of top-level functions in function of analysis are also possible.

Performance Requirements

The degree to which a mission or function must to be carried out; this is generally measured in terms of quantity, quality extent of coverage, timeliness or readiness. When analysing the requirements, based on the factors that govern the life cycle of the system, requirements for performance will be interactively developed across all functions that have been identified; these will also be characterized based on the degree of certainty in the estimate, on how critical they are for the success of the system, and on how they are related to other requirements [5]. Performance requirement means the speed of Data entry, data transferring and processing. As an example in a Point of Sale system (POS) the Data entry must be speedy but not data processing. So performance of these aspects should go with the requirements of the customer, the business and the working environment.

Design Requirements

This shows the requirement process and expresses the technical manuals and data packages, such as Build to, code to, buy to and how to execute. For example, what are Software designs for? How they are designed? Requirements, Life span of software, Maintainability, Sustainability, Upgradability, Independency etc.

Derived Requirements

Derived requirements mean, requirement that are transformed from high-level requirements. As an example we can demonstrate, design requirements get low weight as results of the requirement for long range.

4. System Requirements & Software Requirements

Regarding to this topic system means “An interacting combination of elements to accomplish a defined objective. These include hardware, software, firmware, live ware, techniques, facilities, services and other support elements,” [5] as mention by in the International Council systems engineering. System requirements mean, requirements, which require for the whole system. Component of software contains in a system, and software requirements are obtain from with system requirements.

Process of Requirements

From here we are going to discuss process of software requirements. Showing and discussing 5 sub areas of the requirement’s process.

Process Models

The Process models are as below:

- Needs to be modify for the organization and project context.
- In the software life cycle front-end activity is not individually separate, but refine throughout the life cycle when process initiate at beginning of the project and when it’s continuing.

This topic concerned, how to configure different type of project and constraints with activities of elicitations, analysis, specification, and validations.

Process Actors

From here its introduce role of people who take part in to the requirement process. This kind of process is basically interdisciplinary, and in between software engineering and the domain of the stakeholders, needs to be mediate by requirement specialist. Stakeholders are varying across in the whole project, but they always include users and customers [6]. Typical example as below;

- Users: These are the group you operates the software.
- Customer: These are group whom targeted by the software.
- Market analysis: People in the marketing act as proxy customers, when they need to establish what the market requires, while commissioned customers will not have mass-market products.
- Regulators: Most of the application domains are regulators. Public transport and banking are some of them.

Process Support and Management

From this topic, its introduce project management required resources and requirement process consumed by. Context for the first subarea (Initiation and scope definition) of software engineering management establish by process support and management. Key information of it are, construct a connection to connect to the recognized behaviour of course of action and issues of human assets, tools, cost and the training [6].

Improvement and quality process

From this topic we are discussing the quality and improvement of the requirement process. Its purpose is with keeping customers satisfaction, while to playing key role of requirement process of making cost and timeline for software production. This will help to keep quality standards & process improvements of the model of software and system. These process quality and improvements are related to Software process engineering and Software Quality. This topic is cover;

- Improvement implementation and planning
- Requirement process measures
- Requirement process improvement standards and models.

5. Procedure for Requirement Analysis

There are 15 main task should be done in the requirement analysis. Those tasks are based on part of a national process of requirement analysis. The 15 tasks are as below.

1) *Customer expectations*

From this define the customer's expectations. They may approach from one of the primary functions mention in here. They are; Missions need, Operational requirements, technology based opportunities, high level system requirements and direct communications. The goal of this task is to understand what customer wants and how to accomplish the function of the particular system.

2) *Enterprise and Project Constraints*

This means define and identifying the solutions for the constraints impacting design. Things include both enterprise and constraints are as below;

- Costs
- Control mechanism
- Guidelines
- Project plans
- Updates technical
- Policies and procedures
- Structure
- Team assignments
- Domain technology

3) *External Constraints*

This means define and identifying impacting constraints designing solutions for the actives of the System Engineering Process. This can includes:

- Threat system capabilities
- Technology base
- Capabilities of interacting system
- Public and international laws and regulations

4) Operational Scenarios

Define and identify is, uses of system products for the anticipation of scope. Define is expected for each and every scenarios.

- Physical interconnectivities with Interface of a system and platform
- Interact with e environment and other systems.

5) Measures of effectiveness and Suitability

From here we define and identify overall reflect of customer expectation, which will measures of effectiveness of the system. Measures of effectiveness are relates to how well system performs with the clients missions. Performances, reliability, safety, and operability are the key missions of the measures of effectiveness. How well system performing in the intended environment and its measures of maintainability and supportability are related with the measures of suitability.

6) System Boundaries

- Expected interaction includes open systems approaches, interacting systems in outside the system boundaries, system elements under design control and its higher- level or externals.
- Which drop down outside from developers control and which system component are design command of the performing pursuit.

7) Interface

Products in quantitative terms, Interacting systems, and platforms are defined to external or higher level and physical and functional interfaces. In addition it would like to include electrical, Mechanical, procedural, thermal and data additional commands, to Physical and functional interfaces. An internal/external perceptive are might considered to interfaces.

Those address that the elements are inside the boundaries and established the systems addressed are called as internal interfaces. By contractor identify, controlled and responsible for these kinds of interfaces. Once which make entity relationship with outside-established boundaries are called as external interfaces. Government defines and control, these types of interfaces.

8) Utilization Environment

From this topic is defining the operational scenarios needed environment. Must be define and identify all the factors for environmental that may impact system performance. Below items include to environmental factors;

- Ranges of temperatures
- Weather conditions
- Biological
- Topologies
- Induced
- Time

9) *Life cycle Process Concepts*

There are 8 outputs in keys in life cycle process that should define. They are, develop, operate, support, produce, test, train, distribute, and dispose, are under development in system product. Focusing to the cost drivers and the elements of high risk, which are affordability and anticipate to impact supportability over system life of the system.

10) *Functional Requirements*

From this topic id define what system able to do. Functions are identified via requirement analysis. And it will decompose allocations and functional analysis.

11) *Performance Requirements*

System's high-level function performances are define as performance requirements. Other KPP and measure of effectiveness establish in test plans, which is focus and place by performance requirements.

12) *Operation Modes*

Under the development there are various modes of system products that we can define. Conditions that should include definition of modes of operations.

13) *Technical Performance Measures.*

Key indicators of system performances should identify, which will track the design process. If not met the project cost, schedules or performance risk, Technical performance measures should limited to goals and critical technical thresholds. Tracking the planed versus actual progress of the KPP is involved by the Technical performance measures. Somehow in some areas of the technical performance measures medley is stage reliant. This supposed to be review through beginning of the each stage in ladder of systems engineering procedure.

14) *Physical Characteristics*

Identify and define, under development software products require characteristics. Size weight, colours, text can take as an example for it. Based on trade studies can identify which characteristics have got true constraints, which is not and which can be change.

15) *Human Factors*

Identify and define under development software products, human factors that will affect to operations. Physical space limit, reach, eye movements, climate limits are some of them that we can take as an examples. Based on trade studies can identify which systems integration of human are constrains and which can be changed.

6. Process Models

Process model means, different type of processing levels. When process is an instantiation, process model is type of level of it. Thus process is instantiation, same process model use to develop software application repeatedly. We be able to propose on achievable use of Process model is how possibly contract to process will model what it made to process itself. Process model giving rough idea of what process looks like. The process models goals are as below;

Descriptive

- Track, during the process what will happen
- To make perform more effectively and efficiently, it takes the point of external view of observer whom looks at performance and determines the improvement of the process.

Viewpoint

- How probably will they perform and define the process of what they are preferred.
- Guidelines, lays down rules and behaviour patterns, which will leads to, desired process performances. As a result the part where they be able to austere enforcement to flexible guidance.

Explanatory

- Pre-define points which will make to easy to reporting purpose by extracting the data.
- Explains processes of rationale
- Possible factors based on rational arguments will be evaluate and explore from this topic.

7. Requirement Analysis Issues

Stakeholder issues

Steve McConnell mention about this from his book Rapid Development, “details a number of way users can inhibit requirements gathering” [6]:

- User will not commit to set of written requirements.
- Technically users are unsophisticated
- Lack of knowledge about current technology
- Is not speedy communication with users
- User don't understand about the development process
- Even not often users participates for review
- Users don't have clear idea about their requirements

Engineer Issues

Problems may occurs when the engineers during the requirement analysis. Those issues are listed as below;

- Relatively customer needs, common skills and domain acquaintance, are appropriately identify with by engineers whom frequently carry out by analysis.
- Different vocabularies may have got to end users and technical personals. So this causes to makes believe wrongly that they are in perfect agreements until they supplied finished product.
- Rather than develop system specification as to clients needs, engineers trying to make the requirements that suits to existent system.

Attempted Solutions

Employ systems analysis specialists may try to sort out communication problems in the project. Techniques such as Use cases, prototyping and Unified Modelling Languages introduces in 1990s to sort out the problems mentions

above. And also new application tools introduce to make Communication Bridge to fill the communication gap in between users and the developers or engineers. Furthermore this helps to test marketed earlier produced codes.

- Interactivity
- Ability to capture data needs and logic
- Ability to add requirements of context.
- Imitate the final application, which is capable to generate high fidelity prototypes.
- Capable to capture data needs and business logic
- There is a capability distributed for users interact with the simulation.

8. Requirement Specification

The term Specification means regarding to the engineering is assignment of limits to products designing goals. There are abundance of requirement have need of for typical software. Among large number of requirements, emphasis share to complexity interaction management and numerical quantification performance. So typically software requirement specification refers to the document of a production, so it can evaluate and can approve it.

System definition, System requirements, and software requirements, are the three different types of documents, which use to produce as substantial non-software components for a complex system. Only third one “Software requirements”, use as for simple software product [7]. Approaches for requirement specification are shown as below;

- Natural language
- Structured natural language
- Design description language
- Requirement specification language
- Graphical notation
- Formal specification

Software requirement specifications

Software requirement specification means the agreements in between customers and suppliers, what is expected and what is not expected to do from the software, which is willing to produce. This permits to rigorous requirement, assessment before design, so this can reduces later redesign matters of the software. This supposed to be giving practicable estimated cost of the product, and for the schedules. To make productively organizations validations and verification plans, they will produce their own software requirements. Regarding to the software requirement specifications, while transferring new software to a new user, developer has to give an idea about the software.

Normally software requirements are written in natural language. Apparently for that as to the software requirement specification, we have to note down information in formal or semi-formal method. Describe more sharply by giving clear information about the appropriate selected permit of notations in exact

requirements and aspects of the software architecture. General rules in individual notations, should allow to unfolding exact requirements. This is concerning the certain types of reliable software's and meticulous crucial safety critical [7]. Anyhow choice of notation is constraint for the authors and reader of preference's and skill and for training.

Couple of quality indicators, which related to the software requirement specifications and other project variables, had developed. Schedules, Cost, Performances, and etc can take as an example for it. In individual statement of software requirement specification's quality indicator includes, directives, options, weak phrases, and continuances. Documents for entire software requirement specification's indicators are, readability, depth, specifications, and structure of text.

9. Hindsight

For the hindsight I referred Dutch Flower Market competition and the, Multi-foods International Geneva ERP implementation.

Dutch flower auctions had played a significant task during the world cut-flower industry by means of provided that well-organized centres in lieu of price fortitude and transactions of flowers concerning sellers and the buyers. These auctions are own to Dutch cut-flower farmer cooperative enclose customarily used the Dutch auction seeing that the procedure for price fortitude [8]. This hindsight considers how shifting patters of the international competition, consumer preferences and information technology are possible to cause the company of the Dutch flower auction. As Ajit Kambil & E. Heck telling in their case, they make available an outline for analysing the qualities of out of the ordinary business deal models and utilize this outline to calculate the strengths and weaknesses of presented and wished-for electronic auction models for trading flowers [9]. The intend information technology will facilitate original forms of trading to facilitate will moderately reinstate and complement the traditional Dutch auction as a process of organizing price fortitude and transactions. They recognize how electronic trading will change as of previous mechanisms, and believe main challenges are to the execution of new auction models. In particular they demonstrate how the existing auctions have been prepared to provide the happiness of the farmers, although electronic markets resolve for the most part of the benefit buyers. Therefore they have highlighted the consequence of varying enticement and possession structures in the Dutch flower industry to for practical purposes changeover to novel IT based markets. This case illustrates the range of difficult issues, which take place in the design, and execution of IT based markets, technology changes characterized settings, pre-existing managerial process and authority structures. Key issues causes to for the failure of the project are;

- Changing the requirements while working on the project.
- Failed to recognize the errors and failures
- Assumptions problems

- Rapid development requirements risks

In the International Multifoods case they describes the make an attempt by a average sized business en route for advantageously make use of information technology for their competitive improvement. The managers in their company and its supplementary, Vending Services of America (VSA), saying a chance to leapfrog the competition in the food distribution services industry throughout progress of the good number superior structure available in the industry. The objectives of the new beginning information system was to VSA's vigour and customer understanding in supporting its consumers' and, in the procedure, produce a great deal considered necessary proceeds expansion to the International Multifoods Company parent firm. This situation provides a general idea of the International Multifoods Company and VSA companies, the strategy for improvement of the revitalization system, and the difficulties so as to arise, since the system was direct veteran and equipped for even out [9]. Key issues causes to for the failure of the project are;

- Endless Requirements Story
- Customers have no clear idea what they require.
- Expectation cloud
- Communication gaps

Here below mention the causes of the project failures.

Customers have no clear idea what they require.

Generally problem in the requirements analysis is that customers have only an unclear idea about their need, and it's up to developer to ask the right questions and execute the analysis compulsory to turn this unclear vision into a formally in to a documented software requirements specification, which can in turn, be used as the core for project plan and an engineering architecture.

Changing the requirements while working on the project.

General problem in this software projects is that requirements are defined in the primary stage change as when project progresses. This probably will ensue, whereas prototypes are developing and project is on progresses, customers are adept to obviously see their requirement problems with the preliminary plan and make de rigueur route corrections; this is probably will happens because of the changes in the external environment compel to re-shaping the preliminary production issues. Highly skill project managers are conscious of these naturally and potentially already keep backup plans to handle with these types of changes.

Communication gaps

Frequently customers and engineers are unsuccessful to communicate obviously in between of them. These types of issues are able to happen, because they are come from different worlds and generally customers do not understand what the technical language is using in the project. This conduct to misunderstanding and harsh to miscommunication, and during the requirements analysis stage, an significant task of the project manager is to ensure that both customer and engineering side have a exactly got the perfect idea of what tasks need to achieve and

the deliverable.

Assumptions problems

Risk of arrogant that the customers and developers are having similar judgment about the system requirements. A number of issues are frequently unidentified by the customer until they initially see their application. At this phase of progress, there will be just about undoubtedly negative impacts on the project cost and schedule.

Failed to recognize the errors and failures

Frequently find in many of the reviews is that the requirements elicitation, analysis, and management are organism conduct in an uneconomical method. Quite a few cases like these, requirements inadequacy contain confirm to be a most significant donor to the program person over the budget and behind schedule. It is essential to comprehend that the requirements be able to be a considerable issue to be successful project. The unassailability of the organization's requirements management procedure is supposed to be taken into contemplation when evaluating the risk that requirements might include on the success of the project.

Endless Requirements Story

Requirements risks are changing. These changes wish to twist out of control and avoid an application from being accomplished. Endless requirement stories are moreover not going to completing software projects. Supplementary risk of constant change is the lack of ordinary requirements considerate by the numerous stakeholders through the numerous requirements iterations. Result are confusion, chaos, misunderstanding, and software either never being completed, or if completed never being used.

Documented requirements

It is require to reach a generally considerate of requirements between main project stakeholders are essential to the success of a software project. To achieve general understanding is necessary that the requirements be documented. Text on software requirements force to indicate that good requirements' characteristics include the items below;

- Consistent
- Modifiable
- Complete
- Traceable
- Verifiable
- Understandable
- Creativeness
- Unambiguous

Incorporate requirements into a software life cycles

Up to now, requirements and importance of have a directorially accepted software lifecycle model and the methods supposed to be well established [9]. From the case studies and knowledge of a range of life cycle models, which is include a requirements analysis, requirements management, or stakeholder re-

requirements phase. Through assuring with the intention of the life cycle accepted to procedure within the organization oblige suitable requirements level of the supervision, the risk connected with projects comparative to the requirements will be abridged.

Requirement validation

Requirements validation is necessary step to make sure to facilitate the requirements are correctly recognized and understood. This is an alleviation policy, which goes hand in to hand by process of require for committed stakeholder participation. Since requirements are documented, the stakeholders are supposed to authenticate them. Frequently the requirements acts are validation will reveal requirements issues that know how to expose in no other way.

Managing the requirements changes

Strategies that survive to assist manage requirements creep. Strategies are occupying with supervision of good configuration, change requests formal reviews, and a control process of controller's formal change. Requirements creep is supposed to be approximately 1 percent per month. Apparently for that, creep which is grow more than 2 percent a month is almost certainly a in no doubt sign of a project that will by no means attain the completion. With not including sound strategies for managing changes, a project wish to be unsuccessful.

Though including the sound strategies, stakeholders are obliged to be aware of the high risk of changes and the high outlay. Meant for the high quality of a project, quantities of changes are basically too complicated to do. Those changes are required to be deferred until after a description of the product is effectively delivered.

Giving a training to for the responsible once in requirements management team

It is not an easy task of requirements elicitation, analysis, documentation, verification, and maintenance. Capability to make possible the elicitation of requirements and pursue the procedure throughout to completion requires skills and awareness. Once which is surrounded by the organization that are liable for assuring to facilitate requirements are managed, have to obtain the guidance and mentoring essential to bestow them with the skills to accomplish this responsibility.

10. System Development Life Cycle

Waterfall method is another name for this System Development Life Cycle. The 6 stages, which is connect to each other stages, to build a sound and quality software project. They are explaining as below [10];

1) Software engineering and Modelling

Since software is at all times for a sizeable system, work begins not later than by confirming all the requirements for system elements and subsequently allocating several detachment of these software requirements. This system observation is necessary once the software be required to interface with the further elements as like as hardware, people and other assets. System is the key and signifi-

cant requirement for the subsistence in some unit of the software. Consequently if the system is not in position, the system is supposed to be engineered and put in position. A number of cases, to take out the maximum productivity, the system must be spruced up and re-engineered. On one occasion the ultimate system is tuned, team of the software development studies the software requirement for the specific system.

2) *Software requirement analysis*

Software requirement analysis also knows as feasibility studies. From this stage we discuss about how the software development team reach to their customer and do study about their system. Developers examine the essential for achievable software computerization to the given system. At the end of the feasibility study, the team makes a document, which specify the different exact recommendations for the chosen system. This additionally includes, cost, personnel assignments, target dates, project schedule, etc. Requirements are gathered course of action is focused and intensified in to the particular software. Comprehend the character of software program is to be built, the system analyst are required to be familiar with the information sphere for the software, as well as to the behaviours, necessary functions, interfacing and performance.

3) *System analysis and design*

From this stage we discuss about the software development process, giving the clear idea of the software's structure and its nuances. In provisions of the server technology, the quite a few of tiers required for enclose structural design, database design, data structure design, and etc. All these are defined from this stage. In whole software development cycle, design and analysis are extremely important. Several malfunctions are in design stage possibly will be very pricey to sort-out later. Logical system of the product will be developed from this stage.

4) *Code generation*

Design obliged to be translating into a machine-readable form. From this stage we are discussing the steps of code generation performs. But when the designs are carry out in a detailed method, without having any difficulties can consummate the code generations. Programming tools, which are used to generate codes are; interpreters, compilers, debuggers, and etc. Unlikely high-level programming languages which are using for coding are; Pascal, C, C++, Java. By means of have a high opinion of to the sort of application; the accurate programming language is selected.

5) *Testing*

Formerly when the codes are generated, the software testing will begin. Several different types of testing methods exist to unknot the bugs that were unswerving throughout the previous stages. A typical methods and testing tools are by now existing. Several companies assemble their individual testing tools, which are modifiable for their own software development

6) *Maintenance*

Software determination certainly undergoes modify on one occasion it is deliver to the customer. This amend to happen regarding to the several reasons.

Amend possibly wish to be ensue as some of the unforeseen input ethics into the system. Apparently for that, the changes in the specific system possibly will straight touch the software operations. The software is supposed to be developed to have capacity for changes that might ensue throughout the post implementation stage.

11. Conclusions

From this paper, I identify what are the Software requirements and its specification and how to accomplish a project perfectly without doing or having failures. Furthermore, we identify the most common failures occur, while on going the project, from the hindsight.

To gain the flourishing software project, Requirement analysis and specifications is the path for it. For that it requires objectives and interest, capability that deal with different stakeholders in different kinds of backgrounds. Incredibly in project background, there are things that we are obliged to pay more attention on, analysis, risks associated with the elicitation and validation of requirements. Otherwise as we studied from the hindsight, failure will arise in the project and project will be unsuccessful. The necessary steps that have to take to avoid the failures are:

- The subsequent strong procedures
- Documented requirements
- Incorporate requirements into the software life cycles
- Requirement validations
- Managing the requirement changes
- Giving training for the responsible once in requirements management team

By identifying the risk that arose in the project and taking necessary steps can avoid them and make them strength for the project. As a substitute of requirements soul of the problems, a closely controlled software requirements management procedure knows how to help to guarantee the success of the software project.

Acknowledgements

I would like to express my gratitude to everyone who supported and guided.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] System Engineering Fundamentals, Defense Acquisition University, Fort Belvoir, Virginia, 2001, 31-45.
- [2] Ian Sommerville, Software Engineering 7, Pearson education Inc, 29-36, 94-100.
- [3] Sommerville, I. and Sawyer, P. (1997) Requirements Engineering: A Good Practice

Guide. John Wiley & Sons, Chichester.

- [4] Bourque, P. and Richard, E. (1999) *Guided to the Software Engineering Body of Language*. IEEE Computer Society Press, CA, 1-15.
- [5] Laplante, P.A. (2003) *Requirement Engineering for Software System*. CRC Press, 139-202.
- [6] Nuseibeh, B. and Easterbrook, S. (2000) *Requirement Engineering: A Road Map*. *ICSE'00 Proceedings of the Conference on the Future of Software Engineering*, Limerick, 4-11 June 2000, 35-46. <https://doi.org/10.1145/336512.336523>
- [7] Naik, S. and Tripathy, P. (2008) *Software Testing and Quality Assurance: Theory and Practice*. Wiley-Spektrum. <https://doi.org/10.1002/9780470382844>
- [8] Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. (2000) *Non Functional Requirements in Software Engineering*. Kluwer Academic Publishers, Boston. <https://doi.org/10.1007/978-1-4615-5269-7>
- [9] Van Heck, E. and Ribbers. (1997) *Experiences with Electronic Auctions in the Dutch Flower Industry*. *Electronic Markets*, 29-34. <https://doi.org/10.1080/10196789700000046>
- [10] Moore, J.W. (2006) *The Road Map to Software Engineering: A Standards-Based Guide*. 1st Ed., Wiley-IEEE Computer Society Press.