

Activities, Factors and Formal Technical Reviews on Software Quality Assurance: A Software Testing Strategy

Author

Najmul Islam Ansari¹, Mahesh Kumar Singh²

¹(Research Scholar/Department of CSE/BIET, Lucknow, India)

²(Professor//Department of CSE/BIET, Lucknow, India)

Abstract

To make a Modified Spiral Model using the Software Quality Assurance Activities, Factors, Formal Technical Reviews and Software Testing Strategies and comparison with spiral model make comparative chart .Our goal is to become a cost effective Software Testing and Quality Assurance service provider that help companies deliver reliable and quality software to their clients on time. We hope to achieve this by

- a) Developing a generic test methodology that can be used across projects*
- b) Working in conjunction with the development team*
- c) Carrying out independent, time bound and priority-driven testing*
- d) Freeing the development team from the testing burden thereby allowing them to focus on their core skills*

Keyword: Quality Control, SDLC, Validation, Black-Box Testing, White –Box Testing, Spiral Model, COCOMO, ESTIMACS, SEER-SAM, PUTNAM.

1. Introduction

Software Quality Assurance (SQA) as a planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements'. SQA does this by checking that^[6]:

- a)** Plans are defined according to standards;
- b)** Procedures are performed according to plans;
- c)** Products are implemented according to standards.

2. Development Methodology

Managing The Software Project needs a well-defined Project Management Framework. We, at MAAS InfoSoft follow the top project management practice especially for our offshore software development process. Our state-of the art IT Project Management process is both interactive and open that allows clients to keep track of everything related to his projects including important documents, communication, financial. For our outsourcing clients, we follow the Outsourcing Project Management Process that is transparent and efficient as well. Our effective IT Process management allows them to monitor the whole development process and note any unforeseen problems even from their own end.

Below is our Project Management Framework to give you an insight on our software development methodology.

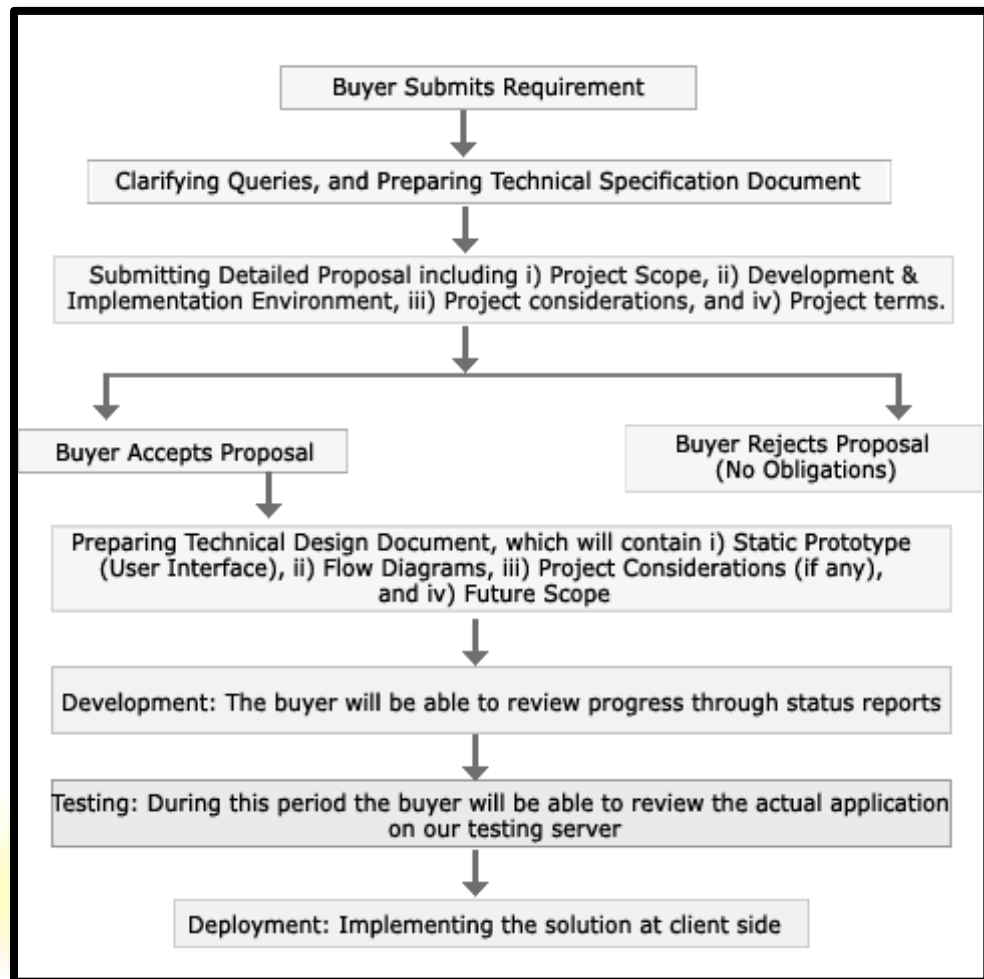


Fig: 1 : Project Management Framework

Major Phases of Project Lifecycle

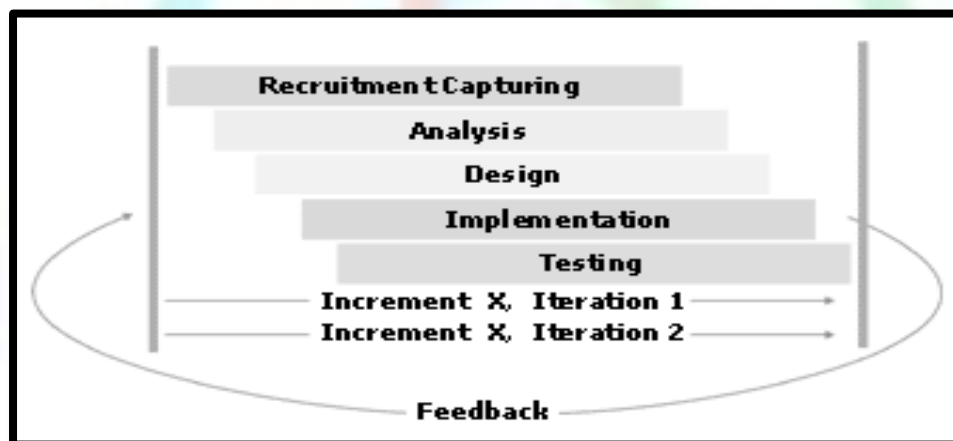


Fig 2 : Phases of Project Lifecycle

The major phases of project lifecycle are described in the following table:

a. Phase Description-

- i. Requirement Analysis
 - Gather details of requirements from the clients.
 - Finalize the User Interface and Flows.

2.1.2. Analysis & Design

- Prepare an Analysis model, comprising of Structural and Dynamic Models of the system.
- Prepare a Design model, converting from Analysis model.

2.1.3. Implementation

- Coding

2.1.4. Testing

- Testing the system

2.1.5. Deployment

- Deploy the system at Client site.

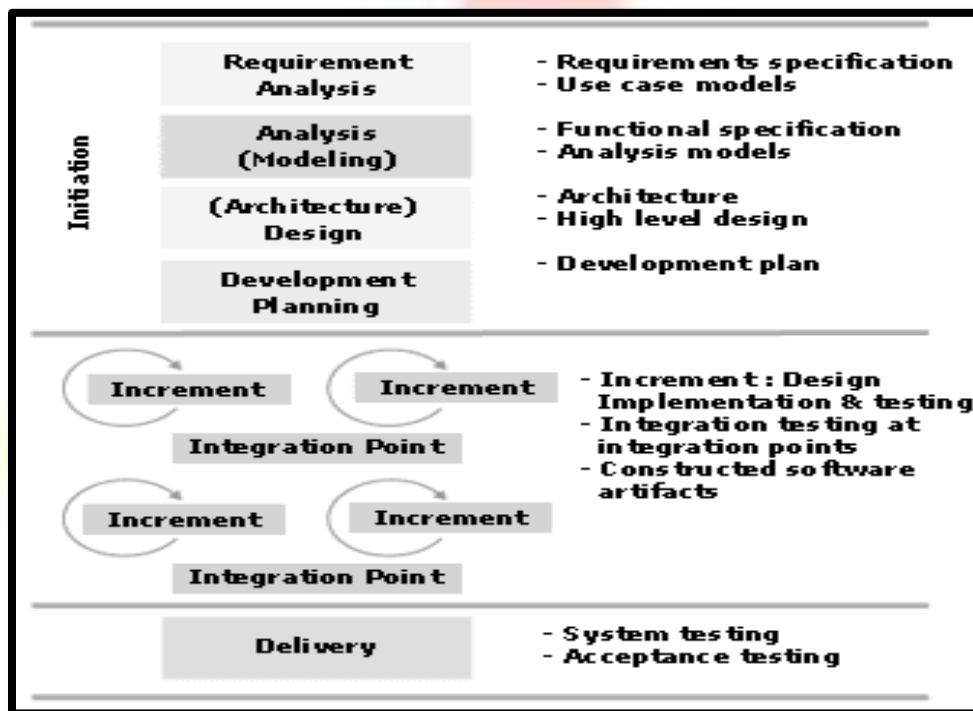


Fig:3 Software Quality Assurance- Formal Technical Review Methods

2.2. Advantages of Spiral model

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for large and mission-critical projects.
- Strong approval and documentation control.
- Additional Functionality can be added at a later date.
- Software is produced early in the SDLC.

2.3. Disadvantages of Spiral model

- Can be a costly model to use.
- Risk analysis requires highly specific expertise.
- Project's success is highly dependent on the risk analysis phase.
- Doesn't work well for smaller projects.

3. Result

Result of Technical Review of spiral model is given below that is outcome of different cost estimation model output of estimation taken by applying result at different parameter Extensibility, Flexibility, Traceability and result is compare by different model of SDLC.

3.1 Result of cost Analysis of Spiral Model

S.No	Model Name	Author	Year of Publication	Technique Used	Parameters			
					Extensibility	Flexibility	Traceability	Easy to Implement
1	ESTIMACS	Howard Rubin	1970	Function Point (TA)	√	√		Being proprietary, accessibility is less
2	PUTNAM's S/W Life Cycle Model (SLIM)	L.H. Putnam,	1978	Ballpark Technique (NAT), Function Point (AT),			√	√
3	COCOMO 81[3][4][5][6]	Barry Boehm	1981	SLOC ^{cs} , KDSI	√			√
4	SEER-SAM	Galorth	1983	Top-down, bottom- up	√	√	√	Suitable for SLC over 2,00,000
5	COCOMO II	USC-CSE & COCOMO II Project Affiliates Organizations	1995	Object Point, Function Point, SLOCs, KSLOC		√	√	Suitable for large size projects

3.2 Testing on reliability of Spiral model

Model/Feature	Waterfall	Spiral	Incremental/ Iterative
Specification of All the Requirements in the beginning	Yes	Not all and Frequently Changed	Not all and Frequently Changed
Long term project	Inappropriate	Appropriate	Appropriate
Complex Project	Inappropriate	Appropriate	Appropriate
Frequently Changed Requirements	Inappropriate	Appropriate	Appropriate
Cost	Not costly	Costly	Costly
Cost estimation	Easy to estimate	Difficult	Difficult
flexibility	Not	Less flexible	Flexible
Simplicity	Simple	Intermediate	Intermediate
Supporting high risk projects	Inappropriate	Appropriate	Appropriate
Guarantee of Success	Less	High	High
Customer Involvement	Low	Low, After Each Iteration	High, After Each Iteration
Testing	Late	At the end of each phase	After every Iteration
Maintenance	Least maintainable	Yes	Maintainable
Ease of Implementation	Easy	Complex	Easy

4. Conclusion

Software Quality Assurance (QA) and testing experience with a wide variety of projects and environments. Functionality, compatibility, reliability, exploratory/ad hoc, load/stress, usability, installation, security and related test methodologies. Testing of web application, client-server, network, multimedia, and database, educational and related projects on Windows, Macintosh and Unix/Linux operating systems. Mentor, coordinator and leadership roles in creation and execution of QA and testing processes, test strategies and plans, bug documentation and regression, and bug tracking system development. Experience with both team and individual work. Professional training in software testing theories, procedures and methodologies.

References-

- [1]. Boehm, B: "Software Engineering", IEEE Transactions on computers, vol. C-25, no.12, December 1976.
- [2]. Boehm: "The Hardware / Software Cost Ratio: Is It a Myth?" IEEE COMPUTER, vol.16, no.3, March 1983.
- [3]. IEEE standard glossary of software engineering terminology, IEEE standard 729-1983.
- [4]. Thayer, R, et al.: "Validating Solutions to Major Problems in Software Engineering Project Management, "IEEE COMPUTER, vol. 15, no.8, August.
- [5]. Thayer, R, et al: "Major Issues in Software Engineering Project Management, IEEE TSE, vol.SE-7, no.4, July 1981.