CHAPTER 4

DESIGN OF QUANTIFIED HYBRID SOFTWARE MODEL

The previous chapter discussed the essentials of the standards of software quality management adopted in the software development industry. This chapter discusses the techniques and standards of the research methodology that was adopted to carry out the study. The chapter will discuss the research methods in-depth and highlight the various types of design principles and validation mechanisms used to carry out the intent of the study.

4.1. Introduction

The domain of software development techniques researched in the past decades uses certain standard techniques to perform research. The proposed study has drawn a significant amount of motivation from the research techniques discussed by Wang, Lin, and Huang (2010) and Fujita and Zualkernan (2008). These authors have discussed some of the interesting patterns of modern software development methods as well as emphasized that significant research on project management requires proper investigation of process management. It was observed that in the past, the majority of studies towards quality standards such as Scrum, Agile, Six Sigma, Kanban, Total Quality Management (TQM), etc. were carried out using qualitative techniques (Hennink, Hutter, & Bailey, 2010) while a few used quantitative techniques. However, both qualitative and quantitative techniques have their advantages and disadvantages (Katsirikou and Skiadas, 2010). Hence, taking this fact into consideration, the proposed study has been carried out by using a hybrid model—created by integrating qualitative, quantitative, analytical, and
experimental approaches—to excavate all relevant details from the data collection. The proposed study intends to understand the process effectiveness in SMEs using an integrated design of Lean and 6σ principles, Agile, and Scrum methodologies as depicted in Figure 4.1.

Figure 4.1. Approach and Principles behind the Proposed Model

The proposed study also uses strategic decision making theory and probability theory to perform mathematical modeling. Brief details about these theories follow.

**Strategic Decision Making:** The software development methodology adopted varies between IT companies. With the rise of Information Technology, the level of interaction among the various individuals associated with the software project development not only increases, but also leads sometimes to higher levels of uncertainty. Hence, in order to retain the proper order of the software development process, a manager will need to adopt the highest degree of fail-proof strategic decision making. The standard theory of strategic decision making (Tschäppeler & Krogerus, 2011) studies via managers the problems in decision making of the various actors involved in the interaction between clients and developers. Strategic decision making is an analytical theory that
enables the decision maker to develop a mathematical model for various conflicts and cooperation between the two parties (Harrington, 2014). The two parties can be either i) client or manager, or ii) manager and developers, or iii) among two organizations themselves. This theory assists in modeling all the strategic interactions that can possibly occur between any two individuals (called players). Each player is defined by a unique and specific set of actions, where the combination of a discrete set of actions is also called a strategy (Tschäppeler & Krogerus, 2011). The outcome of each interaction is basically the relay of a possible set of strategies leading to allocation of specific pre-defined payoffs (or incentives or utilities) (Harrington, 2014). The concept also calls for maximizing the payoff function for each player involved in the interaction process (Harrington, 2014). The proposed system adopts strategic decision making to design a mathematical model for ensuring superior software quality assurance in the IT industry.

**Probability Theory:** Any business model has a higher number of variables cannot be discretized into a specific value while performing mathematical modelling. Hence, a set of such variables are generalized and enclosed within a boundary governed by an upper limit of 1 and a lower limit of 0. Hence, probability theory is all about finding the best favorable outcomes to certain problems divided by the total number of outcomes. It applies the principles of randomization, stochastic theory, and statistics (Kharin, 2013).

The study is carried out in two stages, i.e., i) **pre-investigation** and ii) **post-investigation**.

### 4.1.1. Pre-Investigation Stage

It is imperative to understand the real-time issues, constraints, and limitations of the existing software development methodologies before formulating a new model. It would be quite unfair to propose a model without visualizing the real-time issues of quality assurance in software development methodologies as various studies speak about the different issues, and these
existing issues are quite extensive. Therefore, the proposed study has performed a preliminary investigation considering a sample set of participants to understand the existing issues. The design principles of the proposed model were completely based on the outcome of this phase of the investigation. The pre-investigation stage, also called a pilot study, was carried out on multiple participants. The pilot was supported by a questionnaire containing 28 questions related to the effectiveness of the software development exercise. A mathematical model was designed using strategic decision theory and probability theory, depending on the outcomes. The design of the mathematical model was inspired from the standard literature of Tom Devane (2004) and Mary Poppendieck (2006).

4.1.2. Post-Investigation Stage

After the formulation of the new model, it was essential to understand its effectiveness. This phase of the study considered the feedback from the same set of the participants involved in pre-investigation stage. The prime motive was to understand the extent of the solution that the proposed system provides for the problems described by the participants in the pre-investigation phase. In order to increase the accuracy in data analysis, this phase of the study also considered additional participants to capture feedback for the proposed model.

The post-investigation stage started by evaluating the model considering the participants from the pre-investigation stage along with some new participants. All the participants were employees of different software development organizations which employ more than 200 human resources. More than 90% of the participants reported that they belonged to a custom application development group, followed by 10% from e-commerce application development.
4.2. **Research Design**

This section describes the various procedures that were adopted in the due course of the investigation process for the objective of accomplishing the research goals mentioned in the previous section. The proposed study can, therefore, be discussed under structured stages of the research methodologies as depicted in Figure 4.1

![Diagram of the Adopted Research Methodologies]

**Figure 4.1. Schematic Diagrams of the Adopted Research Methodologies**
4.2.1. **Design of One-Team Framework (OTF)**

After the completion of phase one of the literature review, the issues existing in the current scenario related to software quality assurance and loopholes in the SDMs were evident. Therefore, at this stage, a ‘One-Team framework’ was formulated that is based on the principles of Lean Six Sigma and also adheres to some of the five key elements of LEAN. This framework aims at delighting the customer with reliable and quicker service by improving the processes to display lesser variations and by ensuring process flow. It also ensures a high learning curve for all the employees in each One-Team. Implementation of these ideas requires extension of responsibilities and a standard mode of raising issues. Each development (DEV) program will have a self-sufficient service team, known as the One-Team. This will ensure focus on the activities related to individual program landscapes. The One-Team framework is diagrammatically represented in the Figure 4.2

![Figure 4.2. One-Team Framework](image-url)
4.2.2. **Design of Scrum Methodologies in Highly Distributed Software Projects**

The ‘One-Team Framework’ was designed based on hypothetical interpretation after visualizing the potential benefits of Lean Six Sigma. However, an empirical background is required to analyze the research goal. In this phase of the study, the focus was on those organizations that are multi-located and connected to multiple partners. The scope of study was on small and medium enterprises (SME) which fulfill the aforementioned criteria and had a higher scope of adopting the proposed scrum methodology. The study in this phase introduced a methodology that involves the scrum technique in the software project team allocation considered for highly distributed projects. This study presents a technique with the understanding that by extracting the determinant of Agile called Scrum; it will be possible to maintain quality. Moreover, regular scrum meetings were conducted even for small and medium scale software development organizations which operate in a highly distributed environment. This concept is published in the ISI Cited Journal.

4.2.3. **Design of Quantified Software Methodologies**

The empirical findings accomplished in phase three require proper validation with feasible enhancement. Hence, it was decided that the proposed empirical model be enhanced to an innovative mathematical model using qualitative and quantitative approaches.

Figure 4.3 highlights the architecture of the quantified software methodologies that exhibits five different tiers.
**Figure 4.3. Proposed Model of Study**

**Tier-1 (6σ Tools Deployment):** This tier highlights the tools adopted for the proposed system. The input for this tier is basically client requirements that are captured effectively by the *Process Map*. The Process Map is fundamentally responsible for defining the standards of the business entities to be involved in the proposed software development framework. It is usually accomplished by performing process identification, collection of relevant information, requirements mapping with existing technologies followed by analysis. The output of the Process Map is then subjected to further analysis using the *Control Plan* and *C&E matrix*. The objective of the Control Plan is to enhance team performance over a period of time and it is also one of the potential modules in the DMAIC principles. The Control Plan also provides a written summary of the process that is iteratively enhanced. The *Cause and Effect matrix* (C&E) assists in understanding the attributes that significantly affect the outcomes of the proposed Six Sigma framework. The
C&E matrix is potentially associated with the factor known as Critical-to-quality (CTQ) that is a precise and measureable statement of the quality objectives. After the statements are drafted using C&E Matrix, it is updated to the Control Plan and subjected to further processing using the Statistical Process Model. The Statistical Process Model is another significant quality control factor that frequently adopts statistics for monitoring and controlling each ongoing process in the software development process. It is further noted that the outcome of the C&E matrix is also fed to Measurement system analysis, which is used for identifying the parameters of variation when analyzing the process. Various risk factors and uncertainty factors are well identified in this process as it continuously monitors the methods used for testing as well as measures the tools in the development process. ANOVA is one of the common tools used in this tier. As various sub-processes are subjected to different statistical tools iteratively, the proposed system also performs Capability Analysis for gauging if the outcomes meet the client’s specified standards. The Control chart is a common tool used in this process. Further precision of the process is sustained by including Multivariate data analysis, which adopts statistics to evaluate the data evolving from multiple variables. The system also performs evaluation of reliability by adopting Failure Mode and Effect Analysis (FMEA) that involves cross-checking all the components and their internal layers to explore any possibility of software defects. The entire process is carried out in a sequential process, where the final outcomes result in the superior practice of the software development process.

**Tier 2 (Execution Layer):** This layer acts as the SLA protocol on the tools adopted for the Six Sigma implementation. The core design of this layer is performed using the DMAIC principles of Six Sigma, i.e., Define, Measure, Analyze, Improve, and Control (Coleman, 2012).

**Tier 3 (Support layer):** After the error free process arrives from the Six Sigma tools compliant with the execution layer, the processes are now
ready to be used by the actors in the software development process. The Application Integrator is a module responsible for integrating the various modules developed by the designers and passed successfully through the quality analyst. The Role Manager is a module responsible for integrating all the developed modules and interacts with quality assurance team to get these verified. The Report Generator is responsible for maintaining the updated development process being carried out and the specifications of all the tools and techniques, while Project Tracking is responsible for monitoring the progress of the ongoing software projects. The support layer also repositions each event in the database which is frequently accessed and updated depending on the level of continuous improvement in the process management.

**Tier 4 (Agile-Scrum Layer):** To ensure the spontaneity of the continuous enhancement principles, the Agile-Scrum Layer assists in ensuring that all the actors involved in the process management adhere to the SLA provenance designed based on Lean and Six Sigma principles. This tier is responsible for maintaining the *Sprint Backlog Customization* for further scrutiny of the continuous improvement process. The tier also ensures the *Daily Scrum* and *Sprint* modules to be used in *Product Backlogs* for customization.

**Tier 5 (Integration layer):** This layer mainly performs a double level of the scrutiny of the data deposited by the Support layer. It consists of a *Management Tool* that is usually a custom made framework for checking each process. It also consists of an *Analysis Tool* (e.g., Minitab, Matlab, SAS, R, etc.), which mainly performs data analytics to find defects and the uncertainties included in the recently evolved software prototype. Finally, the *Integration Layer* updates the *Support layer* to perform better decision making in real-time before dispatching the final product to the client.

This phase of study proposed a mathematical model that integrates the potential of Agile, Lean Six Sigma, and Scrum methodologies and proves that SMEs too can adopt the model with assured business benefits. The study
discussed a mathematical model, which was designed based on preliminary investigation of the flaws of methodologies practices currently in software industries. After the empirical evaluation of results, a quantitative study was performed on real time participants in the post-phase of investigation to demonstrate that the model has better supportability of the six research variables, i.e., (i) cost avoidance (CA), (ii) cost reduction (CR), (iii) increased capacity (IC), (iv) retained revenue (RR), (v) retained growth (RG), and (vi) risk management (RM).

4.2.4. **Comparative Model of Quantized Framework**

In this phase, the mathematical model formulated in the previous phase is benchmarked. The current work discusses the comparative analysis of the performance of the outcome of the proposed study with respect to conventional SDM, e.g., waterfall, SCRUM, DFSS, and spiral. For the purpose of the performance analysis, the proposed study considers a similar set of 199 respondents who were introduced to the initial model as a beta version for expert evaluation. A new questionnaire was designed based on the research variables, which is governed by three main performance evaluation parameters, i.e., Design adoptability (DA), cost effectiveness (CE), and uncertainty mitigation (UM). The following chapters describe the outcome of the study to help the readers understand the effectiveness of the proposed study.

4.3. **Summary**

This chapter has discussed the fundamentals of the models that have been proposed in the study for ensuring better quality standards in software development methodologies. The next chapter discusses about the mathematical model that was used to design the principles discussed in this chapter.