CHAPTER 6

IMPLEMENTATION OF QUANTIFIED HYBRID SOFTWARE DEVELOPMENT MODEL

The previous chapter has discussed about the techniques adopted for the proposed study, aided by mathematical modelling, to support the effectiveness of the proposed integrated model developed from Lean principles, Six Sigma, and Agile manifestos. This chapter will particularly discuss the outcome of the proposed study that will assist the readers to elaborately understand the effectiveness of the proposed model.

6.1. Introduction

Analyzing the data and making a correct inference of data is one of the crucial steps in any research work (Langridge, 2009). However, the proposed research work is a bit of a different type compared to majority of the existing studies done till date, as discussed in Chapter-2. Usually, it was seen in review of literature that majority of the authors have formulated hypothesis, collected data, and performed statistical evaluation using various tools and discussed their data based on it. However, the proposed study is quite different from it. In order to understand the problems related to software development methodologies, the study has performed interactions with various participants associated with software development industries. The data collected in this stage is termed as pre-investigation stage, which is basically the fundamental core of the mathematical model being proposed in the study. Hence, a typical kind of data analysis is required to validate the proposed outcomes. As the
study uses a mathematical model, it performs convergence test (Oberkampf, 2010, p.179) to validate the mathematical approach. However, to understand the impact of the model, the study again performs post-investigation stage to further validate the user’s perception and effectiveness of the model followed by standard benchmarking with conventional models.

6.2. **Outcome of Pre-Investigation Stage**

Before formulating all the critical parameters and their modelling aspects, it is important that the current flaws in software development methodologies in IT industry be collected. By doing so, the new model can effectively identify the issues and can formulate countermeasures. Hence, a pilot study was conducted where the semi-structure questionnaire, designed and motivated by the study conducted by Wysocki (2004), was used. The outcome accumulated from the pilot study would assist in giving a clear picture of issues, if any, encountered by existing software development methodologies. Visualization of the current issues can be used for framing up a model/framework using Lean Six Sigma, Scrum, or Agile methodologies for enhancing/mitigating the issues in currently adopted methodologies. The pilot study reflects a research variable that correlates with the situational analysis such as:

- **Participant demographics**: Questions were formed to record the information regarding professional experience and skills.
- **Awareness**: Questions were formed to check the understanding of Agile, Scrum, and Lean Six Sigma.
- **Efficiency**: Questions were formed to check the speed of the development/delivery for the considered participants.
- **Integrity**: Questions were formed to check the extent of team work.
- **Quality**: Questions were formed for assessing the deliveries.
This phase uses a set of 28 questionnaires that further extracts frequencies of the above mentioned research variables. A set of 16 participants was selected for this semi-structured interview process. Discussion of individual questions was out of scope for this paper for two reasons namely i) massive data illustration in pilot study and ii) more emphasis on actual study. However, some significant findings of the pilot study can be stated as follows: (Please refer to Appendix - 1):

**Participant demographics:**
- All the participants originated from Software Development firm with more than 200 human resources deployed.
- More than 90% of the participants rated the development of business software (ERP) as the highest score followed by e-commerce at 56.3%.
- Majority of the participants were working for around 4-10 years in their respective organizations.

**Awareness:**
- Awareness of Lean Six Sigma was very poor among majority of participants while that of Agile was highest.
- Agile and Scrum were among the largest practiced software development methodologies followed by Waterfall and Lean software development.
- The professional experience required for all these software development methodologies was of 4-10 years.

**Efficiency:**
- Lack of authority was found as a major reason for ineffectiveness in software development methodologies followed by lack of planning, low documentation and lack of project structure.
- Brainstorming is the most widely practiced tool for analysis followed by project charter.
• Majority of the participants claimed that the maximum amount of time required for developing any single business requirement is more than 15 working days but less than 30 days.

**Integrity:**
• In case of a slippage in the software project schedule, the pending modules would be developed and delivered separately, as the schedule was data driven. Majority of the participants claimed that the task allocation is majorly done from backlogs followed by the product owner.
• Acceptance criteria for the iteration/task are majorly decided by the product owner followed by PO, SM and team.
• Most of the results showed that the minimum answering time for customer request was found to be less than 24 hours, whereas some results also stated that the answering time depended on the complexity of the customer request.

**Quality:**
• Most of the time, the product/services were delivered to the customer before the stipulated time.
• The majority of the participants claimed that if new technical tasks are discovered, then the product owner re-prioritizes the backlogs and the team determines how the work should be accordingly re-assigned.
• Adoption of Agile and Scrum has significantly altered the quality/cost of development.
• The majority of the participants followed an acceptance test for each iteration or task.
• Most of the acceptance criteria were decided by the product owner followed by the business analyst. A significant variation of 42% was observed in timelines while delivering different versions of the product from the same team.
The majority of the participants claimed that the testing for developed scenarios was done whenever the code was delivered by the developer.

The majority of the participants claimed that there were variations in timelines or quality while delivering two different products from the same team.

Most of the participants claimed they were not aware of any instances of system crashes or service interruptions reported by customers, after their products were installed.

The majority of the participants claimed that in order to ensure better predictability about the success rate of a new product, the core strengths and weaknesses of the technology/business should be thoroughly studied.

6.3. Discussion of Results of Mathematical Model

TOOL: Matlab

Matlab is basically an advanced programming tool that enables the programmer to perform some of the potentially complex mathematical operations that are highly difficult and lengthy in other programming tools like Java, C, or Microsoft.Net. Matlab offers some of the best toolboxes which can be used for addressing a number of issues that arise from different domains like signal processing, neural network, bioinformatics, parallel computing, fuzzy logic, and finance to name a few.

However, a contradiction in the thoughts arises when it comes to justify the selection of Matlab in the proposed study. It is very well established that the present study is basically investigating on the uncertainty and predictive attributes of the risk analysis of the proposed integrated model. However, the present study is completely unique in terms of predictive behavioral analysis as i) sophisticated APIS was not used ii) no applications were designed and iii) defect detection or prevention system was not being built. The prime reason behind this is that till now, the uncertainties in software
development methodologies in the IT industry was a big unanswered question (Yamada, 2013, p.86) (Escalona, 2014, p.475). Hence, it was strongly believed that in order to build a better hybrid modelling and uncertainty mitigation techniques, a clear visualisation of the real-time constraint should be formulated. Past researches have already considered numerous hypothetical constraints in software development methodologies (Hamido, 2009, p.175), where very few of them have successfully been proven as potential and standard models that could be benchmarked. It was also strongly believed that the prime reason behind this issue is that –it is a very challenging task to actually differentiate between the efficient manager and the inefficient manager; sometimes, for which reason, the inefficient manager bypasses the test of software quality assurance in project management. Various studies have been carried out in understanding the problems in software development methodologies (Marchesi, 2003, p.29) (Hibbs et al., 2009, p.39) An extended research in this direction is necessary to understand the uncertain and unpredictable behavior of software development methodologies for better outcomes in terms of software quality assurance. Hence, for the purpose of investigation, a tool with a set of complex mathematical operators was required, which could easily perform simulation based on probabilistic approach, as the strategic decision theory was implemented and the extensive data analysis was being carried out. Hence, Matlab was chosen to work on.

It was also seen that the majority of the prior studies had focused on using statistical tools like R and SPSS, where the estimations of the outcomes are pretty straight forward. Depending on the perceptions accomplished from the field survey, the researchers are often found to perform statistical testing using SPSS, where the outcomes are justified based on probability values. This process of data analysis however has benefit as well as potential drawback. The benefit lies only when the outcome depends only on the perceptions gathered from the participants involved in the study. Hence for the study, whose outcome completely depends on the participants’ feedback, performing
statistical analysis using SPSS is one of the best options. However, the outcomes could be possibly validated in terms of their applicability scenario. Moreover, if the researcher wants to understand the maximum possible calculations in extended scenarios, SPSS doesn’t have the capability to perform real-time computation and analysis of complex data. Moreover, the outcome as well as the contribution of the study becomes quite narrow as the outcomes are more or less directly dependent on the participants’ feedback. The approach of the present study is that the participants’ feedback was used in two ways: i) to understand real-world issues and constraints and ii) to understand the validity of the mathematical model in terms of users’ experience. Adoption of Matlab has two advantages i) performing convergence testing of model and ii) performing extended simulation of the model in the most challenging scenarios.

6.3.1. **Postulations in Implementation**

The simulation has been carried out in Matlab considering multiple different scenarios for evaluating the behavioral pattern of efficient and inefficient manager involved in the study. The simulation study basically considers the target of the inefficient manager for invoking a risk in software development methodologies while retaining all the strategies to be adopted for getting itself penalized or being reported by the efficient manager. On the other hand, simulation parameters are also designed in such a way that the update and reporting activities of Scrum by the efficient manager should reduce the extent of defect as well as risk in software development methodologies, caused by the impractical and unstructured decision of an inefficient manager. The similar simulation environment parameters will be used as detailed below. (Table 5.1 in Chapter-5):

The model is designed in such a compatible way that the policy makers can understand it. Some of the significant parameters like Technical Adoption, Interoperability, Cost Avoidance, Cost Reduction, Increased
Capacity, Retained Revenue, Retained Growth, and Risk Management are defined in this model. They can be also considered as performance parameters. However, it is quite challenging to design the process, as making the assumptions and considerations of the studies are quite a difficult part of the design principle.

**Formulation of Department:** For this purpose, clustering technique was applied, where the proposed model can actually consider n number of clusters, which should be equivalent to number of departments in an organisation.

**Formulation of Human Resources:** For this purpose, random distribution technique was applied considering n number of human resources to be working in every department of the considered organisation.

**Categorization of Human Resources:** Basically, this was not an easy task as there were various attributes that cannot be initialized. Hence, the manager variable was initialized with two attributes namely i) efficient manager and ii) inefficient manager.

Other simulation parameters are described as below:

- One hundred managers were randomly placed in a 900 m × 900 m space model which is evenly divided into multiple departments.
- Any two managers within the same department were considered as associate managers.
- Managers follow the department-based work assignment.
- Each simulation is repeated 1000 times, and the average data was used as the final result.
- The default number of inefficient managers is 40.

A variable Gain is defined as that which represents the payoff for the decision being taken by the manager. However, for better computational process, a variable Gain of SLA_violation is defined to solve two purposes: i) it will motivate the rational and effective decision taken by the manager by increasing their payoff represented by the variable Gain of SLA_Compliance
and ii) it will demotivate the irrational decision taken by the manager by decreasing their payoff. Another variable Gain for Scrum_Report is formulated only for efficient manager as it is meant to adhere to Scrum protocols. The variable allows extra payoff to the efficient manager after performing regular Scrum protocol. There are a few more variables like Cost of SLA_Violation, Cost of SLA_Compliance, and Cost of Risk. Cost of SLA_Violation is defined as the total consumption of resources in the situation, where the manager declines to adhere to the SLA protocols. This variable is continuously monitored on a time-frame and delivery cycles to understand the wastage of resources. Cost of SLA_Compliance is defined as the total consumption of resources in a situation when the manager always adheres to SLA protocols. This variable will give a tentative idea about the cost avoidance as well as cost reduction during the cycles of software development methodologies. Cost of Risk is defined as the total resources consumed and utilized after the inefficient manager has already been practicing his irrational decision. This variable will assist the computation to understand more about the risk factors. Certain parameters are also defined for checking the false alarm. The term false alarm means that an efficient manager reports another efficient manager as an inefficient manager. The proposed study uses a threshold based technique to formulate the rationality of the reports being generated by the managers. Therefore, the system ensures that it should minimize false alarm and adhere to furnishing the correct report of the productivity and operational cycles of software development methodologies. This is one of the unique properties of the model, as it not only detects the fault but also attempts to reduce it. On one hand, it identifies the problems arising from the wrong decisions and uncertainties and on the other hand, it also suggests the best software development model to be adopted in such odd situations that require a precise decision to be made.

6.3.2. Staged Observational Outcomes
A study of software development methodologies from literature suggests the existence of numerous mobility concepts. However, our simulation study is based on multiple departments where a set of defined managers (both efficient and inefficient managers) exist. The simulation needs to be carried out using multi-constraint based interaction, where it is important for a particular inefficient manager in a specific department to get synchronised with other sets of inefficient managers by adhoc means in other departments. This consideration is done for increasing the potentiality of the risk factor and thereby to testify the efficiency of the strategic decision theory to identify the uncertainties. Consideration of the strategy formulation (discussed in previous chapter) plays a crucial role in the simulation study as it has a significant impact on the quantity of the average adherence to SLA protocols among the departments, thereby influencing the performance of the managers. Therefore, randomization theory was deployed as it closely matches with the real-time based simulation requirements of the investigation.

The simulation environment chooses three essential SLA protocols based on i) Agile manifestos, ii) Lean Six Sigma, and iii) Proposed hybrid model. Figure 6.1 highlights the linearity of mean utility for the SLA compliance, where only Agile manifestos are seen to be operational. The graph exhibits the first stages of software development methodologies where the entire development cycle is observed from eight stages of time-bound progress. Here, it can be inferred that the proposed hybrid model has less chance to get adopted in the first cycle by both the types of managers owing to their own personal reasons. As Lean Six Sigma is the prime backbone of the proposed model, it has also got fewer chances of adoptability. Interestingly, the proposed model emphasised that if the manager chooses to opt for Agile manifestos in considered higher dimensional challenges of real-time constraints, their utility (or payoff) does not increase, which indicates the effectiveness in the decision.
Figure 6.1. Utility for the SLA Compliance

Figure 6.2 exhibits the test-scenario where the mean utility is captured for evaluating the extent of SLA violation for the second stage of software development process under study. The outcome shows that the mean utility is highly linear and doesn’t increase if the managers are found to violate SLA protocols of the proposed study. The outcome has both advantage as well as limitation. The advantage is that managers are provided with a hint that if they are opting for decision pertaining to SLA violation, they do not receive any increase in their utility factor. However, the limitation of this outcome is that both efficient and inefficient managers can take decisions that are detrimental for SLA compliance. It becomes a very challenging task to understand the extent of inefficiencies or risk related factors associated with the decisions of the managers in the second stage of observation.
Figure 6.2. Utility for the SLA Violation

Probability of model crash is evaluated by observing the high values of uncertainties and higher number of SLA violation. The outcome for the third observational round exhibited in Figure 6.3 provides two basic information such as i) probability of model crash using Agile is consistently zero and ii) utility does not seem to increase (as seen from Figure 6.1 and Figure 6.2). Hence, this is the mechanism adopted by the proposed model to demotivate the managers who decide to stick to Agile manifestos under the considered real-time constraint.
The proposed model emphasized that if wrong decisions are taken for ensuring software quality assurance, then it will lead to risk. Hence, identification of potential risk factor is very crucial. The outcome shown in Figure 6.4 for the fourth stage of observational round shows that by adopting Agile manifestos, the managers are not able to counter the real-time constraints considered for the proposed study. The risk factor although linear, values to zero, indicating that the model is failing to identify the risk factor for Agile manifestos. Hence, this is another form of demotivation for the managers if they stick to inappropriate quality standards and completely ignore efficient standards like Lean principles and Six Sigma.
The simulation study thereby considers managers to stick to their primitive Agile manifestos. However, from the fifth observational round onwards, the simulation study prompts the managers to equally consider both the proposed model and Lean Six Sigma principles. A closer look into the outcome in Figure 6.5 shows that Agile manifestos accomplishes the peak utility values only five times in a very controlled research environment, whereas Lean Six Sigma principles accomplish the peak utility values for six times. The proposed model successfully outperforms both Agile manifestos and Lean Six Sigma by accomplishing the peak utility values nine times. Also, a closer look into the pattern of adoption under the similar real-time constraint shows that both Agile and Lean Six Sigma principles have irregular frequencies whereas the frequency of the proposed model is less irregular.
Figure 6.5. Mean Utility for the SLA Violation

Figure 6.6 exhibits the mean utility of the SLA compliance. A closer look into the outcome shows that the proposed model is gaining better equilibrium stage in the sixth round of observation. Cumulative utility for the proposed system is better than Lean Six Sigma principle; however, in this observational round, Agile was seen as not to be performing well in decision making. The outcome clearly reflects that Lean Six Sigma is the backbone of the proposed hybrid model and that is why the proposed hybrid model dominates over conventional Lean Six Sigma principle. The prime reason behind this is that the proposed model performs continuous evaluation of the cost of SLA compliance and attempts to gain more in SLA compliance. In this case, the managers opting for primary models will be demotivated.
Figure 6.6. Mean Utility for the SLA Compliance

Figure 6.7 exhibits the probability of the model crash for the seventh observational simulation round. The outcome shows some of the interesting patterns of curves. The curve representing Agile shows a continuous pattern of increment showing higher probability of model crash for the given real-time constraint. The curve representing Lean Six Sigma is found to be optimally superior compared to Agile; however, the curve also shows slight linearity in the model crash which neither increases nor decreases much.

The proposed model includes heavy computation of algorithms due to which the model has to estimate the uncertainties and various other probability factors in the preliminary rounds. However, the curve seems to replicate the characteristics of the gradient descent exhibiting that the probability of model
crash may be there in the preliminary rounds, but it gradually decreases with the progress of the model computation.

Figure 6.7. Probability of Model Crash

Uncertainties are the root causes of majority of failures of the software development models. However, it is a very important parameter in the proposed study as the model also intends to find out what is the extent of the minimization of uncertainties in the eighth and final observational round. Figure 6.8 shows that Agile curve shows gradual descent characteristics for uncertainties, but Lean Six Sigma principle outperforms it. However, the proposed model was seen to have a slight increase in uncertainties for the second and the third population, but the
curve slowly descends down and exhibits better performance compared to other models.

![Graph](image)

**Figure 6.8. Analysis of Uncertainties**

6.3.3. **Manager Utility Outcomes**

This section of the outcome analysis discusses about both efficient and inefficient manager’s behavior and the utility performance of the proposed model. The outcome exhibited in Figure 6.9 shows the cumulative amount of utility being imposed on a manager who takes the correct decision. Correct decision is judged by the high reduction in uncertainties, higher number of detected SLA compliance, and higher number of Scrum reports. Hence, these are the inherent characteristics which the model chooses to map with the efficient manager. The outcome shows that if the manager chooses to opt for Agile or Lean Six Sigma principle, the outcomes for utility are found to be degrading in nature as compared to the proposed system. The outcome also
exhibits a higher gain of SLA compliance and adherence to Scrum reporting system, both of which are included in the proposed model as protocols.

One important point to be noted is that the proposed model doesn’t have the prior information about the manager types (efficient or inefficient), it just knows the number of managers. For the purpose of identifying the managers, the proposed model designs a protocol that maps the characteristics of efficient as well as inefficient managers. Therefore, for this purpose, the study performs continuous assessment of probability factors as well as tracks the utilities being imposed on their managers. Another important fact to be understood, for the clarity of Figure 6.10, is that the inefficient manager will never try to adopt the proposed model to make the decision. They assume that if they do not adopt the proposed model, their inefficiencies will be easier to get appraised in Scrum meetings. Hence, this is one of the most important part
of the discussion that an inefficient manager has inclinations towards risk whereas an efficient manager has inclinations towards the reporting system, as it can highly control and minimize the risk. Based on the real-time Scrum reports, necessary measures could be taken to ensure fewer defects and more productivity with efficiency. Hence, an inefficient manager has a higher likelihood of selecting only Agile or Lean Six Sigma principles (as the benchmarking has been performed with only these two) and is less likely to opt for the proposed model. The outcome clarifies the fact that if the inefficient managers opt for Agile manifestos under considered real-time constraints, then their utility are very high, which is again a demotivation for them. This is because the inefficient managers can maximize their utility further if they opt for the proposed hybrid model, which is not likely to be done in this case. Moreover, the adoption of Lean Six Sigma principle minimizes the average utility to negativity showing the extent of rare adoptability of Lean Six Sigma principle.

Figure 6.10. Analysis of Average Utility for Manager
Therefore, from Figure 6.9 and Figure 6.10, it can be seen that it is possible to keep a track of dynamic behavior of managers using the proposed model and categorizes them as efficient or inefficient. This can be done by continuously monitoring their utilities, which is quite higher for efficient manager and much lower for inefficient manager.

6.3.4. **Convergence Test**

Convergence test is one of the standard techniques to understand the nature of the mathematical model proposed. The basic purpose of this form of test is to validate the appropriateness in the mathematical model which should be very less divergent in nature and more convergent in nature. Hence, the study intends to find the nature of equilibrium as stated. The convergent nature of the proposed mathematical model built on real-time constraint can also be mapped with the hypothesised model to ensure similar behavior of convergence with real-time application. Basically, the phrase tries to exhibit that the proposed model ensures one state of equilibrium. Hence, if the managers, who were using Agile or Lean Six Sigma principle previously, can also adapt the proposed model with almost similar outcomes of reality. The prime purpose was also to exhibit the extent of technical adoption of the proposed model by various users.

Figure 6.11 highlights the convergence test for belief, where the x-axis represents index of constraints (where the proposed system has considered around 30 real-time constraints in software development methodologies) and y-axis represents estimated theta (Probability that a manager will opt for SLA violation). The outcome shows optimal converging behavior for all the models. Hence, it can said that the proposed model is more likely to get adopted by an organization which is already practicing Agile or Lean principles.

Figure 6.12 represents the outcome of the convergence test for uncertainties. The proposed system persistently monitored the uncertainties in every test environment and it was found that the proposed model exhibited
better convergence behavior along with Agile and Lean. Hence, the mathematical effectiveness of the proposed system is established. The model also needed to undergo more tests for a better understanding of the final outcomes. Hence, the proposed model was tried on selected participants, based on their circumstances. The participant’s perceptions were collected to study the user experience of the proposed system as discussed in the next section.

Figure 6.11. Convergence Test for belief
6.4. **Outcomes of Post-Investigation Stage**

Based on the results accomplished from the qualitative analysis, the study had a clear visualization of the issues related to the effectiveness of software development methodologies. After developing the model, the model was given for testing to the old participants, who were involved in the pilot study. New participants were also considered for further confirmation. This phase of the study considers the empirical evaluation with parameters like Cost of Quality (COQ), which is again classified into Cost of Good Quality (COGQ) and Cost of Poor Quality (COPQ).

To capture the user experience/feedback, a new set of questionnaire was designed using a six point Likert scale. Six questions were formulated each for evaluating the following performance factors of the proposed model namely i) cost avoidance, ii) cost reduction, iii) increased capacity, iv) retained revenue, v) retained growth, and vi) risk management. This phase performed close monitoring of the defect minimization process applied on the framework...
of software development methodologies discussed in Section 4. The next section will discuss the result accomplished from the study.

The proposed analysis was based on the fact that the proposed mathematical module was empirically tested on real-time data to explore the accomplished result from the model in the best way. Therefore, the study considers usage of Cost of Quality (COQ), which was again classified into Cost of Good Quality (COGQ) and Cost of Poor Quality (COPQ). COGQ was further classified into Appraisal Cost and Prevention Cost, while COPQ was further classified into Internal Failure Cost and External Failure Cost, motivated by the study of Laporte et al. (2008).

Therefore, the evaluation of the proposed mathematical module was done in the following steps:

6.4.1. Pseudocode for COPQ

**Input:** S, O, U, Y

**Output:** Value of COPQ

**Start**

Set S for sales / annual revenue in local currency.
Set O for output / number of units completed.
Set U for steps / number of process steps from start to finish.
Set Y for yield / percentage of units with perfect quality which means single touch and no rework.

Estimate the total opportunities of failure \( \text{TOP} = U \times S \).

Estimate the total number of defective units \( D = (100\%-Y) \times O \).

Estimate the defect rate per unit \( \text{DPU} = \frac{D}{\text{TOP}} \).

Estimate the defects per million opportunities \( \text{DPMO} = \frac{\text{DPU}}{\text{TOP}} \).

Estimate the quality level in the short-term/long-term \( \sigma = (\text{Say } 3.2/1.7) \)

Estimate the cost of poor quality as sales percentage in the short-term/long-term \( \text{COPQ} \)

Estimate the cost of poor quality in local currency, short-term \( \text{COPQ} (Z=0) \)

Estimate the cost of poor quality in local currency, long-term \( \text{COPQ} (Z+/-1.5) \)
Figure 6.13 shows the results of the mathematical module which shows that presence of High Yield is equivalent to Lower Defect and thereby ensures low COPQ. It should also be noted that the number of trials in x-axis represents the levels of Sigma (1-6), while y-axis represents the results (Yield/COPQ). However, at Six Sigma, COPQ was found to be below 5% of sales and at one Sigma, COPQ was found to be higher than 40% of the sales. The final outcome of the proposed model was based on the inputs received from 199 respondents.

Figure 6.13. COPQ as percentage of Sales versus Sigma Level

Figure 6.13 (simulated in Matlab) exhibits the cumulative outcome of the study. The x-axis of the figures represents the Likert scale ranges of 1-6 while y-axis represents the performance parameters. The values are plotted as per the feedback received from the participants after using the model. The outcome showed that the proposed model ensures better risk management as well as better retained growth. The proposed model also supported retained revenue, increased capacity, cost avoidance, and cost reduction. The outcomes of cost avoidance, cost reduction, increased capacity, retained revenue,
retained growth, and risk management are respectively exhibited in Figures 6.14-6.19.

**Cost Avoidance:** From the results as shown in Figure 6.14, it can be seen that the proposed model was successful in minimising the barriers of knowledge integration and Scrum meeting, which was the critical loophole in a majority of the existing work. The second attribute to have the positive impact was the establishment of stability of both pre- and post-phase design phase and reduced requirement volatility directly. It also equally assisted in building cost-effective Six Sigma design as well as meeting the targeted cost of the product.

![Figure 6.14. Outcomes of Cost Avoidance](image)

**Cost Reduction:** From the results as shown in Figure 6.15, the most prominent effect of the proposed mathematical model was the adoption of sprint, thereby ensuring less iteration in work. It also equally emphasized higher flexibility of tools' usage (open source/proprietary tools) and ranking with efficient maintenance of product backlogs that drastically minimized the bugs in software projects.
Increased Capacity: From the results as shown in Figure 6.16, the proposed model has evaluated using Lean Six Sigma and DMAIC principles, where the results showed potential strengthening of skills of the resources. Just like in cost avoidance, the set of results was also explored with uncertainty minimization and higher scope of cross-functional team involvement. To some extent, it also permitted the policy makers to formulate business intelligence.

Retained Revenue: From the results as shown in Figure 6.17, it can be seen that the best part of the model was in identifying the targeted defects in the software products. The model also stressed on Agile practices along with
Scrum and Six Sigma principles and efficient generation of product backlogs on time. It also effectively maintained the communication systems.

**Figure 6.17. Outcomes of Retained Revenue**

**Retained Growth**: From the results as shown in Figure 6.18, the prominent result accomplished in the post-phase evaluation was the minimised cost in business-process automation. Hence, it is very much ideal for SMEs to adopt Six Sigma principles in their daily practices. Consideration of the external factors like competitive strategies, customer demands, etc. and internal factors like cost, design, etc. was found to highly retain the growth scope. The model also showed a better integration of both DMAIC and DMADV principles for maximizing the business profit by reducing the defects exponentially.

**Figure 6.18. Outcomes of Retained Growth**

**Risk Management**: From the results as shown in Figure 6.19, one of the best part of the implementation of the proposed mathematical model was the
regular monitoring and using the DMAIC results in risk management, which was rendered feasible by adoption of scientific methods and usage of statistical tools (specifically, cause and effect diagram, confidence interval calculation, COQ calculation, failure mode effect analysis, measurement system evaluation, p control chart, regression analysis by using DMAIC process).

The interpretations of the accomplished outcomes of the study are as follows:

Apart from the above mentioned outcomes, the model also exhibited better technical adoption and interoperability. Due to easy deployment process and customized features, the framework can be used by any individual who has a fair knowledge about corporate project development methodologies. The framework is highly generic and it can be highly customized to attain interoperable features in software development process.

### 6.5. Benchmarking the Mathematical Model

The proposed study has formulated a mathematical model that integrates the potential of Agile, Lean Six Sigma, and Scrum methodologies and proved that SMEs too can adopt the model with assured business benefits. The brief descriptions of the models used for comparative analysis are as follows:
**Waterfall Model**: A conventional software development model that uses sequential design process where the progress is visualised to have a gradual descent from phases of conception, initiation, analysis, design, construction, testing, production and maintenance.

**Scrum**: An iterative and incremental Agile software development framework for managing software projects and product or application development.

**DFSS**: Also known as Design For Six Sigma, is an emerging business-process management methodology, related to conventional Six Sigma. It is also relevant to the complex system/product synthesis phase, especially in the context of unprecedented system development.

**Spiral model**: A risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, the Spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.

The work has also discussed a mathematical model, which was designed and based on preliminary investigation of the flaws of software development methodologies being currently practiced in software industries. After the empirical accomplishment of results, a quantitative study was performed on real-time participants in post-phase investigation to show that the model has a better supportability of the six research variables namely i) cost avoidance, ii) cost reduction, iii) increased capacity, iv) retained revenue, v) retained growth, and vi) risk management. This phase of the study will now discuss the extensive evaluation of the proposed model with the frequently used model in software development like Waterfall, Scrum, and Agile principles. Hence, it is crucial that the proposed model be evaluated on the basis of i) technical adoption, ii) cost-effective deployment, iii) solving uncertainty issues related to defect, cost, and risk management in present IT industries.

Various literature (Hamido, 2009) has already discussed that existing software development methodologies are shrouded by loopholes in
modelling that seriously impacts process development. It was also seen that renowned waterfall model (Mall, 2014, p.38) is characterized by knowable sequential properties with clear stress on structure and potential design reviews, which constitute the least responsible existing software development methodologies; whereas, the refined versions (Scrum and Spiral) can cater to the dynamic needs of the customer. Not only this, various literature (Larman & Vodde, 2008) have also proved that modelling approaches based on Lean principles and Agile components yield better cost-effective software development methodologies.

Therefore, the proposed comparative study considers multiple respondents who were also part of the analysis in the previous study, so that the outcomes are impartial and empirically reliable in nature. For this purpose, a new set of questionnaire was designed in relation to three evaluation parameters namely (i) technical adoption, ii) cost-effective deployment, and iii) solving uncertainty issues related to defect, cost, and risk management in present IT industries.

This section discusses the cumulative outcome of the proposed comparative study.
Figure 6.20 exhibits the technical adoption of the proposed system with respect to the existing conventional software development models. It can be seen that after using the proposed mathematical model, the respondents shared that the proposed model has perceived usefulness better and was also easy to use compared to its immediate model i.e. waterfall model. Although waterfall model has better adoption among SMEs, it is not a robust technique of methodologies in IT industries, as it is not feasible for any non-trivial software development project to get one phase of a software product’s lifecycle perfected before shifting on to the next consecutive phases and get upgraded. Interestingly, a majority of the respondents who have good idea about the topic have shared that Scrum may be the best exercise, which can bring revolution to software development methodologies. However, when it comes to be
compared with the proposed idea, which has actually used the amalgamation of Scrum, Agile, and Lean Six Sigma, the respondents expressed that the Scrum model may be the third choice after Waterfall model. The prime reason expressed by the respondents is that Scrum is very much theoretical if implemented on its own, as it gives a vague definition of a small team of members of right size, whereas software projects usually require bigger team size effort. The meetings, updates and sprints are usually violated when the clients are distributed globally.

However, the proposed system can outperform the existing system by integrating the potential factors of Scrum, Agile, and Lean Six Sigma principles. Hence, the persisting drawbacks will be overruled when the proposed mechanism is deployed in the cases of SMEs and highly distributed in nature. Figure 6.21 exhibits that the proposed framework has higher cost effectiveness when compared to Waterfall model, Spiral development, Scrum, and DFSS. The result shows that adoption of the proposed model doesn’t require any implementation of process management that raises cost and expenditure; and, it doesn’t require any bigger dimension of a change in management, which users normally worry about.

The proposed mathematical model can be integrated along with any existing software development model which can directly import the parameters required for project management and by the usage of effective tools like Cause & Effect Diagram, Confidence Interval Calculator, Failure Model Effect Analysis, Six Sigma DMAIC estimator, etc. However, migration from one software development methodology is quite expensive in nature as it requires the team to get used to a new environment of methodologies where knowledge about the new model plays a mandatory part. The proposed model does not require any such inclusion of unwanted exercises and lets the team to be more focused in increased capacity and retained revenue with less defect rate, thereby proving its superiority to any existing software development models.
Figure 6.21. Comparative Analysis of Cost Effectiveness in Implementation

Figure 6.21 exhibits the comparative analysis of cost effectiveness in implementation using the proposed model that integrates the potential capability of Lean Six Sigma, Scrum, and Agile. It is already known that the existing software project development procedures have been associated with various cadres of risk, which is sometimes a computationally expensive process for finding solutions. The above figure shows that the proposed model has accomplished a very high uncertainty minimization compared to conventional models like Waterfall, Scrum, DFSS and Spiral.

The issues of Waterfall model surfaces from the complexity of fully identifying the product and customer needs at the beginning of the software projects. Hence, Waterfall model, though sometimes preferred and suitable for small scaled industries, doesn’t cater to the dynamic needs of large scale industries. The Spiral model is most suitable for complicated software projects where both customer and performance requirements are poorly understood;
however, it doesn’t have any explicit procedural guidance in software development process. Further, DFSS and Lean principle by standalone cannot mitigate the various types of uncertainties that exist.

By creating transparency in its design approach, the proposed model addresses both detecting the ambiguities of the design process along with quantizing the associated influence on the project objectives. The proposed model also introduces a technique that gives a better elucidation of possible threat revelation in IT product development that is closely associated with the decision making process.

The proposed system creates a better resilient mathematical model that addresses the capability of an effective process design framework for mitigating the unwanted defects, encouraging more Scrum meeting, and lending more exposure to analysis on meta-process, thereby resulting in an effective process modelling which is always a better version of the existing software development models. The proposed model includes the receptive and defensive systems. The receptiveness of the proposed model describes how it reacts to unanticipated events, like the capability to spot and rectify defects quickly, cost efficient change in management, or flexibility and versatility of workforce. The defensive system addresses on instituting critical and risk-appropriate defense to accommodate negative influence, regarding cost, schedules, redundancy, or over engineering, and also lower-level buffers such as surplus capability of other manpower resources or testing facilities.
Figure 6.22. Comparative Analysis of Uncertainty Mitigation
Figure 6.23 shows the benchmarked outcome when the proposed system is compared with existing standards. For this purpose, a computational model in Matlab is designed considering the effects of all the Six Sigma deployment tools discussed earlier. As the complete evaluation is done based on the feedback of the participants as discussed from Figure 6.20 till Figure 6.22, it was essentially important to understand the behavior of the model when subjected to various real-time uncertainties (or rather impediments) in the existing software development process. Hence, the computational model is designed considering the variables of increasing uncertainties as well as defect minimization, by using the proposed tool. After plotting the results, it was found that the proposed system (depicted by the green dots) has a predominant positive effect on software development methodologies with increasing uncertainties, whereas Agile and Lean Six Sigma, when considered...
individually, are the second best method after the proposed model of integrated framework.

Hence, it can be seen from the cumulative outcome of the proposed comparative analysis that the proposed model bears a better suitability for design adoptability, cost effectiveness, and uncertainty mitigation. All the three evaluation factors were seen to excel better outcome when compared with the conventional software development modelling process. The prime cause of this fact is the mathematical approach of the proposed system, which can identify and reduce the defect, and hence stabilize the dynamicity in the process management, as it exists in the development process.

6.6. Summary

The prime concentration of the proposed system is to perform an investigation related to the performance analysis where the outcome is studied with respect to research variables. The proposed study has considered an extensive global perspective to illustrate the integral findings of the potential factors of the model discussed. For the purpose of furnishing efficient comparability, all the respondents were properly sampled. Considering the proposed framework, the respondents using the Waterfall model accomplished a comparably poor performance in cost and uncertainty evaluation, where risk factors were not much addressed as in the proposed system.

To some extent, the Spiral model performed better than the Waterfall model, as it accomplished better results with respect to design adoption. However, the Spiral model was not found to perform better when cost (reduction/avoidance) is considered. One of the interesting facts explored in this study is that the respondents who have prior experience working in Scrum methodologies have expressed their opinion which is a little contradictory as compared to that in the theoretical study related to Scrum methodologies. The respondents using DFSS claimed that it has only addressed the uncertainty issues to a minor extent. The investigated results show that the Waterfall model
is not an efficient methodology that any IT firm should work on, incase they want to circumvent any sort of future risk or uncertainty. Once again, the comparative analysis put forward a justification that the proposed model is easy to use and doesn’t require much engineering for implementing in real-time practice for catering to the dynamic needs of customers. The model also incorporates quality in process management where the new user, who wants to adopt it, shouldn’t hesitate to deploy it in real-time.