CHAPTER 3

FUZZY LOGIC BASED RELIABILITY QUANTIFICATION FRAMEWORK (FLSRQF)

3.1 BACKGROUND

Measuring quality is the key to develop high quality software. Reliability is the quality factor with more influence in later phase of software development life cycle (Wende Kong, 2009). Every software dependent organization has interested to produce reliable software. That’s why reliability of software system has been recognized as one of the most important attribute as long as quality of the software is concern. The problem of estimation or prediction of reliability of a software system has been discussed extensively in the literature (Ashish, et. al, 2014b; Bonthu and Khan, 2013; Fenton et al., 2008; Goel, 1985; Hsu and Huang, 2011; Lyu, 2007; Tyagi and Sharma, 2014a).

System developers analyze and understand the business of the proposed system by analyzing the user requirement documents as well as by studying the corresponding detail design document. The use of framework as an effective means of understanding any complex process and the interrelationships among different aspects
and requirements of the system is as old as engineering. Basically, a framework prescribed a road map that helps researchers as well as practitioners to perform any complex process in a systematic manner (Georgieva, et al., 2011).

After realizing the need and significance of the framework for quantifying reliability as discussed in the previous chapter, this chapter presents an integrated and prescriptive Fuzzy Logic based Software Reliability Quantification Framework (FLSRQF). The proposed framework, depicted pictorially in Figure 3.1, has been structured in a way that it could be easily implemented by industry personnel as well as researchers. The reliability quantification process, prescribed in the framework, is comprised of eight phases namely Conceptualization, Identification, Association, Quantification, Corroboration, Analysis, Assessment and Amendment and Packaging (Rizvi, et al, 2016c). All these phases are comprehensively described in the following sections.

3.2 PREMISES

A framework is a hypothetical description of a complex process. It provides a factual base for future research. The framework proposed in this chapter has the following assumptions:

- The framework quantifies reliability on the basis of requirements and design stage measures.

- Fuzzy logic inference system has been used for processing subjective information in terms of early stage software metrics.
• Reliability is being predicted before the coding phase starts.

• Research focuses only product oriented metrics for reliability prediction.

• Framework concentrates on four object-oriented design constructs. (Encapsulation, Coupling, Inheritance and Cohesion)

3.3 CONCEPTUALIZATION

It is the primary step to device a comprehensive solution for an important problem. Consequently more and more brainstorming is required in this phase. Significance of this step lies in the fact that it provides the basis for sprouting preliminary set of specifications to succeeding steps of development.

3.3.1 Assess Need and Significance

Recognizing software reliability as a critical quality factor, its timely prediction is extremely important. The proposed framework will provide a road map for quantifying reliability in early stage of development that will help not only researchers, but also industry personnel.

3.3.2 Explore Advantage at Early Stage

In general an accurate estimate of reliability can be obtained through software reliability models only in the later phases of software development like testing (Mohanta et al., 2010; Rizvi, et al, 2014). Predicting the software reliability early would
be useful for software designers since it provides vital information to take decision on design and resource allocation and thereby facilitates efficient and effective development process towards developing a reliable product (Mohanta et al., 2011). Therefore, it is reasonable to develop models that more accurately arrest the faults as early as possible, before they propagate undetected to later stages and cause severe and unrecoverable damage.

3.3.3 Assess the Contribution of Fuzzy Logic

Techniques based on fuzzy set theory have been emerging as robust optimization techniques that can decipher highly complex, nonlinear, correlated and discontinuous problems (Ross, 2010). As most of the early stage software metrics are not very comprehensible, based on expert’s opinion and involve high and complex dependencies among themselves. Therefore fuzzy logic inference systems have found useful in capturing and processing subjective information in terms of early stage software metrics (Rizvi, et al., 2016b).

3.3.4 Explore Developmental Feasibility

It is evident from the review of the literature that no such fuzzy logic based framework exists that quantified reliability of a software on the basis of requirement and design measures. This fact further strengthens its significance as well as developmental feasibility.
3.4 IDENTIFICATION

In order to reach an appreciable solution, it is needed to visualize the actual problem intelligibly, and identify the factors that are related directly or indirectly to the problem as well as its solution (Lyu and Cai, 2007). There is no doubt, that reliability will not have significant value if its underlying factors are not identified appropriately.

3.4.1 Identify Reliability Factors

Conservatively, most of the researchers had identified timing and the failure rate as the factors that affect reliability, but focusing only on these two is not enough (Pham, 2006). Therefore this sub-section of the framework, suggest the researcher to explore other factors also, those impact reliability more significantly.

3.4.2 Select One or More Key Factors

Although literature has been highlighting a variety of factors those may affect software reliability either positively or negatively. Considering all these factors simultaneously in a study would not be feasible, therefore framework advices to concentrate on few of them.

3.4.3 Identify Requirements and Design Level Metrics

As the motive behind the development of this framework is to quantifying software reliability early in the development life cycle, therefore after selecting the key factor(s), the next step is to identify those requirements and design constructs that may affect the key factor(s) identified in the previous section.
Figure 3.1 Fuzzy Logic based Software Reliability Quantification Framework
(FLSRQF)
3.5 ASSOCIATION

It is true that the quantified reliability value will not have any significance until the underlying components don’t have proper and justified relationship with each other. Therefore, the objective of this phase in the proposed framework is to align all the components together by justifying their role in the framework. Further this phase rationalize the association among the various artifact identified in the previous phase.

3.5.1 Correlate Requirements and Design Metrics with Key Factor(s)

As the proposed framework considering the requirements and design metrics, as the basic building blocks, for quantifying reliability, therefore before finalizing the metrics it is expected to justify how these are impacting the selected key factors positively or negatively.

3.5.2 Correlate Key Factor(s) with Reliability

After justifying the association of identified requirements and design stage metrics with key factor(s), the next concern is to rationalize how these factors are related with Reliability of the software. So, that correlation of identified metrics with software reliability could be justified.

3.5.3 Finalize the Metrics Set

After ensuring the above two correlations this sub section finalizes the requirements and design metrics to be used as input and output variables in the fuzzy inference process to quantifying reliability.
3.6 QUANTIFICATION

As the focus of this framework is to quantify the reliability with an early stage perspective, and most of the early stage metrics are subjective in nature, therefore, to deal with such lack of objective data and uncertainty this framework has focuses on the fuzzy inference system. The sub tasks for this phase are described as follows.

3.6.1 Select Input and Output Variables

Identification of suitable input (independent) and output (dependent) variables plays a significant role in the development of the Fuzzy Inference System (Kumar and Misra, 2008). This step will use the metrics finalized in the last sub-section of the association phase as input and output variables during fuzzy inference process.

3.6.2 Develop Fuzzy Profiles

Developing fuzzy profiles of identified input and output variables is one of critical step to integrate human knowledge with engineering systems (Kumar, 2009). Therefore for implementing fuzzification membership functions are derived for identified variables and subsequently represent them with appropriate linguistic variables like very low (VL), low (L), medium (M), high (H), and very high (VH).

3.6.3 Develop Fuzzy Rule Base

The next step after defining the fuzzy profiles is to specify fuzzy rules. These rules are specified as IF-THEN conditional statement (Zadeh, 1989). The quality
of these rules is very significant, so that the fuzzy system can imitate the conclusions close of an actual expert (Aljahdali and Debnath, 2004; Yadav et al., 2003; Zhang and Pham, 2000).

3.6.4 Perform Fuzzification

Generally, in real world scenario, values of most of the variable are crisp in nature. Therefore, to get the desired output through fuzzy reasoning it is needed to transform these crisp values into fuzzy. The process of mapping classical crisp set values into equivalent fuzzy set values is fuzzification (Ross, 2010).

3.6.5 Perform Defuzzification

The output of the fuzzification process is a linguistic variable and most of the applications required crisp output. Therefore, a process is needed that convert these fuzzy conclusion into crisp values, such process is referred as defuzzification. There exists a verity of defuzzification methods like Mean of maximum, Bisector, Largest of maximum, Center of Gravity and Smallest of maximum (Kai-Yuan, 1996; Ross, 2010).

3.7 CORROBORATION

The primary question for any newly developed model is its validity. Corroborations are generally categorized as theoretical and empirical. Theoretical corroboration or validation addresses the question ‘is the model measuring the attribute it is claims to measure?’, and empirical validation addresses the question ‘is the model useful in the sense that it is related to other variables in anticipated ways?’ (Fenton and Pfleeger, 1998). In other words theoretical corroboration allows the researcher to
conclude that the relationships involved in the model are valid in the light of theoretical concepts. While the empirical corroboration assesses how well the model is able to estimate or quantify the respective quality attributes (Berg and Broek, 1996). As pointed out in the second section of this paper that validity was a major concern in earlier models, therefore to fill this gap proposed framework emphasizes on quality validation. It has suggested that, while implementing framework, the reliability assessment model should be corroborated theoretically as well as empirically.

3.8 ANALYSIS

Quantifying and validating the software reliability is not sufficient, until it should be accompanied by valuable suggestions. The following sub sections describing that developing suggestive measures along with the guidelines for improving the reliability is a key task of this phase of the framework.

3.8.1 Analyze Quantified Reliability and Metrics

After successfully validating the reliability model, this is the phase of the framework where obtained quantified values will be assessed and analyzed collectively as well as individually to know how different early stage constructs influence the reliability separately and/or jointly.

3.8.2 Perform Contextual Interpretation

Different contextual interpretations and facts, resulted from the analysis of quantified values should also be discussed analytically.
3.8.3 Develop Suggestive Measures

Developing the suitable reliability improvement guidelines are the preventive measures that are advised to be taken in advance. These guidelines will assist to regulate the values of the requirements and design metrics, and improve the reliability of the developing software.

3.8.4 Finalize the Framework

The proposed framework would be finalized, after incorporating the appropriate suggestive revisions, came out of the above quantitative analysis, to achieve better level of the reliability. These revisions will definitely proved to be significant in making the finally delivered software more reliable.

3.9 ASSESSMENT AND AMENDMENT

The motivation behind the inclusion of this phase is to make the framework more flexible, progressive and implementable. This phase provides the opportunity to assess the current status and perform the needed amendment (if any) in any of the earlier phase. These revisions followed by assessments facilitate to improve the reliability quantification process proposed in the framework.

3.10 PACKAGING

Finally comes the conclusive phase of this Fuzzy Logic based Reliability Quantification Framework. It is the phase where the developed model is prepared along with the desirable accessories those makes this reliability quantification model a ready to use artifact.
3.11 FRAMEWORK’S KEY FEATURES

After describing all the phases of the framework along with their subsections, this part of the paper lists some of the salient features of the framework as follows:

- The framework is quite prescriptive in nature, and will definitely facilitate industry professionals and researchers to quantify software reliability in the early stage of development, and subsequently decrease the probability of software’s unreliability.

- Based on the analysis of quantified values the framework assists developers by providing them an opportunity, to improve requirements and design related internal characteristics ahead of writing the final code.

- Consideration of the requirements phase along with the design provides this framework an edge over other frameworks or approaches those are based on only design phase, because ignoring or overlooking requirements deficiencies and only concentrating on making the design constructs superior will not seems good enough.

- In order to overcome the limitations of subjective values of early metric, the framework has utilized the strength of fuzzy inference in its quantification phase.

- To ensure the validity of the developed model, the validation phase suggests a systematic methodology through pre-tryout followed by integrating valid Changes and then to perform final try-out.
• The ‘assessment and amendment’ phase of the framework further strengthens its practicality as well as viability by keeping the doors of improvement open for any of the earlier phases.

• In most of the cases, developed models only provide quantitative values but neither provides suggestions on how to make improvement, nor the precautions on how to avoid abnormalities. Therefore, to fill this gap framework recommends to provide needed suggestive measures based on the results and contextual interpretations.

• Apart from the above, reassessment of previously developed or underdevelopment reliability models could be done as per the guidance of the proposed framework.

• Beside this, as far as further research is concern, the framework may open fresh avenues for the researchers, doing research on reliability estimation.

3.12 CONCLUSION

As a solution to the shortcoming of earlier reliability prediction efforts this chapter has proposed a structured and prescriptive framework that may overcome the inadequacies of earlier studies and quantifies the reliability on the basis of the requirement and design phase measures, before the coding starts. Salient characteristics of the framework have also listed just before this section. As far as implementation and validation of the framework is concern, it is presented in the following chapters.