ABSTRACT

Software is an integral part of many of the modern intelligent systems used for critical applications like satellite launch vehicles, strategic missiles, aircrafts, nuclear power plants and patient monitoring systems. The objective of the study is to develop a model for the assessment of the quality of software used in such intelligent systems. The software used in onboard computers of satellite launch vehicles, denoted as flight software is used as a typical example in this study. The flight software has the mission critical function of injecting a satellite precisely into the required orbit. Any error or defect in the flight software can jeopardize the entire mission.

A standard method of assessing the quality of software is by software reliability prediction. The application of statistical modelling techniques has been pursued for this purpose. A number of models have been proposed for characterising software reliability. A study of the various models available reveals, no single statistical model can predict both the quality and reliability of software used in intelligent systems. Different defect prediction models were also studied. It was noted that though the models are able to predict the defects from the empirical data available, none of the available software quality prediction models are able to capture the complex uncertain relationship between software quality, external factors and internal factors effectively. Also these models often have restricted capability in cases where incomplete information is available, which is quite often the case in the early stages of any software development process.
At present flight software quality is achieved through a well-defined development process and thorough defect detection and defect prevention techniques, appropriately applied at all phases of software life cycle. A quality model which describes the important quality attributes the flight software should have was not formally defined for flight software. First step was to define a quality model for specifying the quality requirements and the evaluation criteria of quality of flight software. A quality model specifying the quality factors and their sub-factors was defined for onboard software.

From the verification and validation activities carried out during the software development life cycle of flight software, the factors that have an impact on the quality of the software was identified in the four phases- requirements, design, coding and testing phases. The attributes corresponding to these factors were finalized from the evaluator’s view. When the quality attributes are evaluated, if they do not meet the specifications, it is denoted as a defect. The methodology adopted in this work is to collect and analyze the defects that occurred during the different phases of the software development cycle. From the root cause analysis of the defects, it is possible to correlate the different input factors in the development environment responsible for these defects. Once the reason for these defects is identified, it is possible to develop a defect prediction model to predict the defects that are likely to occur when software is developed in similar environment. The correctness with which the number of errors can be predicted depends on how correctly the development environment is characterized. It was observed that these input factors and their relationship to software quality attributes is a complex, non-linear and multivariate problem. The available mathematical
models for reliability and quality predictions cannot capture the characteristics of the development environment. A possible solution is to develop a model that does not make any assumption about either the development environment or external parameters.

Hence in this research, we have adopted artificial intelligence oriented computing for knowledge engineering. It is the ability of these systems to infer from the information resident in a knowledge base, especially when the information is imprecise and incomplete, that makes artificial intelligence based computing suited for predicting quality of flight software. Neural networks have the advantages of adaptive learning, non-linear generalization, fault tolerance, resistance to noisy data and parallel computation abilities. Recent advances in neural networks show that they can be used in applications that involve predictions. Neural networks were studied as a means of capturing the intricate interrelationship between various input factors during the software development, and the outputs which is mainly the defects detected during evaluation. If the environment characteristics translated into input data along with the defects obtained in such an environment are used for training neural networks, the defects that are likely to occur for new software being developed can be predicted. In this research work, the complexity of the problem, documentation correctness and adequacy, experience of designer/tester and domain expertise, size, process models adopted are all used as inputs to predict the defects. Four Back Propagation Networks were built for each identified phase of the software development life-cycle, and trained with the data available from previous projects to predict the defects in each phase for a new software development. If the predicted number of errors in the requirements phase
and design phase are high, there is a very high probability for the quality of code that is delivered to be poor. The network was trained till a specific error tolerance was reached and was tested by varying the input factors individually and collectively by different percentages and the effect on the defects predicted was also studied. Thus it was possible to test the robustness of the model by perturbing the input parameters and study the effect on the predictions by the model. Through this we can thus identify the most important input factors which can introduce errors into the software developed. Thus identifying the most critical input and taking appropriate preventive and corrective action for preventing those defects, is possible using a neural network based defect prediction model. The model was validated using a fresh set of data and the outputs obtained were as expected.

Also from these defects which are predicted by the model, it is proposed to build a quality assessment model by generating a set of adaptive neuro-fuzzy rules which can assess the software quality and are optimized by training strategies originating from neural network. The integration of neural networks and fuzzy logic into a neuro-fuzzy model reaps the benefits of both neural networks and fuzzy logic systems. In neuro-fuzzy systems, neural networks are used in augmenting numerical processing of fuzzy sets, such as membership function elicitation, and realization of mappings between fuzzy sets that is utilised as fuzzy rules. The ultimate output of the Neuro- Fuzzy Model is the quality index from which an assessment of the quality of the software developed is possible. Depending on the quality index value it is possible to say whether the
quality of the software developed is excellent, good, fair or poor and also it is possible to assess the risk in using that software for an application.

The thesis work will deliver both research insights and practical methods for quality assessment of software used in intelligent systems, considering the Flight Software embedded in Flight Computers of Launch Vehicles as a typical example.