CHAPTER 7

CONCLUSIONS AND FUTURE SCOPE
Quality is never an accident; it is always the result of high intention, sincere effort, intelligent direction and skillful execution; it represents the wise choice of many alternatives.

~ William A. Foster

CONCLUSIONS AND FUTURE SCOPE

Software development has revolutionized in the recent times to adopt component based software development. Components in the form of specialized independent modules are built and joined to form software. The quality of such components together forms the quality of complete software.

The quality of software needs to be assured throughout the software development process. The development team need appropriate information about the quality factors and sub-factors to better understand their impact on managing the component quality. However, previous component quality models do not attempt to systematically capture and document the vital, crucial quality factors which can target the quality of the component during its development. Also, there is no broader framework which describes the mapping between the quality metrics and the sub-factors and the impact of one on the other.

The quality evaluation of ready-to-use component provides no opportunity to the developer in improving the quality of the ready component. Thereby, there is a dire need to set up a mechanism to
Conclusions And Future Scope

tasses quality while the component is being developed to mend the flaws in component. Most of the work in the domain of the software quality has been confined to commercially-off-the-shelf software. Further, quality attributes have been recognized in the past studies but in-depth study of sub-attributes and software metrics relations have not been analyzed. Keeping in view the existing research gaps and its significance the following study was undertaken:

"Component Based Systems: A Quality Assurance Framework"

The broad objective of this research was to study the conceptual and practical framework for the evaluation of the software quality. The specific objectives of the study were:

1. To design a framework by constructing a structure wherein the attributes of the structure are defined.
2. To identify the various component quality metrics and to create the framework of these metrics.
3. The identified metrics in the form of quality assurance framework will be applied on to the component(s) to validate the identified metrics.
4. To map the developer's perspective of the various quality factors and sub-factors to the identified metrics.
5. To propose a quality assurance model for Component Based Software Development.

Numerous software quality factors exist but due to the time constraint and as per the objectives of the study five quality factors were considered namely efficiency, maintainability, reliability,
portability and usability. These quality factors are common and vital as highlighted in the internationally recognized software quality models viz. McCall, Boehm, ISO 9126 and Gillies relational quality model. The scope of the component quality metrics was limited to the study of four software quality metrics suite which are CK, MOOD, LI and Lorenz & Kidd suite. These suites are globally acknowledged, pertained to OOD and are well suited as per the objectives of the study. To validate the metrics various OSS components are available with source code but due to the time constraint the scope of the study was confined to only nine versions i.e. Apache Ivy (three versions), Heritrix (three versions) and JFreeChart (three versions).

The objectives were achieved using a combination of theoretical and empirical approach. Theoretically the quality models were analyzed to identify the quality factors, subsequently the corresponding quality sub-factors of each factor were identified. The metric suites were studied to ascertain the software metrics. An empirical approach was followed to validate the metrics on the components. Also, to investigate the correlation analysis between the metrics and sub-factors empirical methodology was adopted. On the basis of the mapping between the quality factors, sub-factors and software metrics the quality assurance framework was proposed.

7.1 Major Findings

The major findings of the study are summarized as follows:

⇒ The most common significant quality factors among the prevalent quality models (McCall, Boehm, ISO 9126 and Gillies
Conclusions And Future Scope

relational quality model) are efficiency, maintainability, reliability, portability and usability.

⇒ The identified sub-factors corresponding to different quality factors are:

- **Efficiency**: time behaviour, resource behaviour, efficiency compliance, reply time, processing speed, execution efficiency, hardware independence and robust.

- **Maintainability**: analyzability, changeability, stability, testability, correctability, extensibility, reusability, modularity, adaptiveness, perfectiveness, preventiveness, system age, understandability, documentation, error debugging, maintainability compliance.

- **Portability**: adaptability, installability, coexistence, replaceability, portability compliance, conformance, reusability, transferability and flexibility.

- **Reliability**: maturity, fault tolerance, accuracy, completeness, recoverability, survivability, consistency, simplicity, error tolerance, statistical behaviour, availability, integrity and reliability compliance.

- **Usability**: understandability, learnability, operability, attractiveness, ease of use, communicativeness, user friendly, accessibility, customer satisfaction, documentation, training and usability compliance.

⇒ The software metric suites were examined to identify the metrics to evaluate quality of the components. The metrics were
identified from the study of CK, MOOD, LI and Lorenz and Kidd metrics suites. List of identified metrics are WMC, RFC, DIT, CBO, NOC, AIF, AHF, MHF, MIF and PF.

⇒ Mapping of quality factors and sub-factors with the identified metrics was performed.

7.2 Conclusions

On the basis of findings it may be concluded that among the metrics increasing value of MHF improved almost all the quality sub-factors. The developers of component are suggested to have high values of MHF except for attractiveness and reusability. To make the component more efficient WMC, RFC, DIT, CBO and AIF have to be decreased. However, attractiveness improves with WMC, RFC, DIT, NOC and AIF. To improve operability, RFC and NOC may be increased. DIT has direct relation with reusability, modularity, transferability, recoverability and fault tolerance. The CBO metric seems to be reduced to the minimum level as it does not have direct relation with any quality sub-factor.

♦ Efficiency sub-factors decline with increasing values of WMC, RFC, CBO and AIF. On the contrary, PF and MHF improve sub-factors of efficiency. Processing is adversely affected by DIT.

♦ Maintainability sub-factors improve with MHF with the exception of reusability. Analyzability, changeability, testability, understandability, error debugging, extensibility and documentation have adverse relation with WMC, RFC, CBO and AIF. Stability, adaptability and correctability have negative
Conclusions And Future Scope

relation with RFC and CBO. DIT and NOC have a mixed response with sub-factors of maintainability.

- Portability improves with decreasing values of WMC, RFC and CBO. Increase in MHF makes component portable. Increase in the DIT and AIF makes component reusable and transferable but deteriorates replaceability. Adaptability and transferability improves with lower value of NOC. Replaceability and reusability improves with high values of NOC.

- Reliability sub-factors improve with increase in MHF and NOC values. On the contrary, quality improves with diminishing values of WMC, RFC, CBO and AIF. DIT has a mixed response with sub-factors of reliability; rising DIT helps fault tolerance, recoverability and error tolerance. Whereas, declining DIT alleviates simplicity, survivability and maturity sub-factors of reliability.

- Usability sub-factors decline with increasing values of WMC (except for attractiveness), DIT (except for attractiveness), AIF (except for attractiveness), CBO and PF however, usability sub-factors are directly related with NOC. Other metrics have a mixed response with usability sub-factors.

- The quality assurance framework was accordingly developed on the basis of mapping of quality factors, sub-factors with software metrics.

205
7.3 Future Scope

Software developers may be provided with the proposed quality assurance framework and accordingly the component may be developed. Later, the opinion of software developers and as well as of the users, may be gathered about the quality of the component, to consolidate the perspective of this study. Further, opinion can be gathered from scholars to strengthen the view point regarding the framework of quality assurance to assess the component quality.