CHAPTER 8
CONCLUDING REMARKS

In this thesis, some research investigations have been presented aiming towards theoretical and computational aspects of usability and its impact in software development environment. The important conclusions on the outcome of our research are summarized as follows:

1. Software quality attributes are the important characteristics associated with the development of quality software. Each quality attribute pertains to specific property of software and has unique effect in software development process. As a first objective of our research work, we have proposed a generalized classification of software quality attributes in view of software product and process both. In this classification, it is important to note that usability attribute is one of the software quality attribute. Further, we have presented classification of quality attributes that covers several classes and subclasses of usability attributes based upon usability considerations for producing usable software. Also, an attempt is made to focus on the comparison of generalized classification and classification of quality attributes in usability perspective for understanding the dynamic role/behavior of usability attributes. It will provide systematic organization of usability attributes for software usability assurance and also will be highly useful for developing/computing measures associated with software usability.

2. There exists many usability requirements associated with the usable software being developed. Usability requirements represent the user expectations with usability goals of developing software. For systematic representation of usability requirements, we have proposed a classification of usability requirements based on usability issues and usability attributes. This classification covers issues related to the process and product both. Based upon this classification, we have identified
different stakeholders such as; general users, developers and researchers and their usability requirements to produce a usable software. The usability requirements of these three stakeholders are collected to identify the most cohesive usability requirements. We have termed these highly cohesive usability requirements as the key concerns. For requirements of each pair of stakeholders, we have computed Yule's coefficient of association. The result shows the existence of association between different pairs of stakeholder's requirements with varying amount. Similarly, Karl Pearson's coefficient of association (Chi-square) has been computed for assessment of cohesion between all stakeholders' requirements. It shows a very high association between stakeholders' usability requirements. Thus, it may be concluded that the most cohesive usability requirements i.e. key concerns are the vital factors for software usability. The proposed classification will be useful to specify usability requirements in a systematic and structured manner and for usability assessment as well. In addition, the key concerns act as essential factors contributing soundly in usability of software and will be helpful in usability evaluation. It is desirable to provide the usability requirements specification framework to clearly understand, mention and implement usability requirements.

3. We have attempted to discuss some existing fundamental project parameters such as; project size, project type and project development approach. Also, some related terminology has been conceptualized including software project, level of influence, weight, project influence and average project influence. A mechanism has been developed for assigning levels of influence to these parameters and to assign suitable weights to each level of influence. We have proposed An Influence Assessment Method (IAM) for computation of project influence and average project influence of project parameters concerned with a variety of projects on n number of usability attributes. We have performed case study with cases of specific projects using various schemes of weights concerned with four levels of influences using IAM. The study reveals that the average project influence on some usability attributes is constant whereas variation has been observed in case of other attributes. Also, based upon individual project parameters in different projects average project
influence shows the dynamic behavior of some usability attributes. We have referred the views of developers to realize the performance of IAM on some live projects also. Thus, it is stated that project influence plays a vital role in projects (in usability perspective) and will be highly helpful to the developers to deal with these usability attributes during usable software development. This measure project influence will be highly useful for evaluating impact and ranking of usability attributes.

4. The strength of a usable software may be affected by the role, impact and ranking of usability attributes. Thus, we have attempted for ranking usability attributes. Based on project parameters and terminology stated earlier, some new terminology has been defined for computing the ranks of usability attributes such as; dependency level, dependency value, project dependency value and mean dependency value. An algorithm, “CompuRank” has been devised employing empirical approach for computation of ranks (termed as suggested ranks) of $n$ usability attributes in software projects. We have computed suggested ranks of twelve usability attributes using various schemes of weights of four influence levels with three project parameters using twenty four projects. It has been observed that always practicability attains highest rank whereas resilience attains lowest rank. A survey based approach is also used to compute ranks (termed as assessed ranks) of usability attributes. Using ethnography method of empirical research, it involved around sixty five software professionals (having varying experience) in three groups (viz. $G_1$, $G_2$, $G_3$) of varying sizes. As a measure of association between the suggested and assessed ranks, we have used a non-parametric correlation coefficient i.e. Kendall’s Tau ($\tau$). It shows that there exist lower association between suggested ranks and assessed ranks obtained using various schemes in groups $G_1$, $G_2$ whereas there exist a higher association in group $G_3$. It clearly indicates a remarkable closeness of suggested ranks and assessed ranks in $G_3$ i.e. a group of highly experienced professionals. Thus, we may conclude that the suggested ranks of usability attributes are acceptable in usable software development. The algorithm “CompuRank” may be further generalized for any number of software project
parameters. Ranking usability attributes will be of great support to developers while software development to produce usable software. Also, it will be highly useful for usability evaluation and measurement.

5. Scenarios are identified as the best means of determining development related qualities represented through quality attributes. We have proposed another method of ranking usability attributes using scenarios and termed as Scenario Based Ranking Method (SBRM). Considering project parameters, influence levels of project parameters, weight and project influence, we have defined some new terminology required for computation of usability attributes' ranks using SBRM such as; level of acceptance of scenarios, project influence, usability influence and overall influence. We have identified the scenarios associated with each usability attribute. We have studied the performance of SBRM with four cases of acceptance of scenarios such as; random number of scenarios accepted, all scenarios accepted, more than 50% scenarios accepted and less than 50% scenarios accepted for each usability attribute. Our study reveals that in all cases, computed rank of practicability is highest whereas resilience ranked at lowest rank all the time. We have computed Kendall's Tau (τ) coefficient to find any association of ranks computed using SBRM with ranks computed using algorithmic approach. The values of τ for above said cases show a high association in each case. This high association of rankings clearly states that the ranks computed using SBRM are remarkably close as compared to the ranks computed using algorithmic approach. Thus, SBRM is proved to be the best alternative to algorithmic approach of ranks computation with criterion of scenarios. Inclusion of these ranks during software development will be helpful for generating usability metrics, to provide measures of usability during software development and also to guess cost of usable software.

6. Classical Software Development Cycle (CSDLC) is the traditional way providing a systematic ordered manner for software development. There exists another approach Usability Engineering Life Cycle (UELC) as it is the contemporary manner of development focusing primarily on users' goals and satisfaction. It has
been observed that CSDLC with usability consideration from conception stage may lead to improve the user satisfaction. We have attempted for inducing UELC with CSDLC to present Collaborative Life Cycle (CLC) for attaining usability of software. In CLC, we have successfully collaborated the usability activities with each phase of CSDLC appropriately. Also, a Collaborative Development Model (CDM) as a Win-Win Spiral model has been introduced for this integrated CLC. CDM accommodated the phases of CLC and the usability factors productively and assured the development of usable software since each phase starts with usability goals and ends with usability evaluation. It provides measures during development required to assess user satisfaction iteratively. In general, CDM provides an iterative, incremental, evolutionary and concurrent approach for development. It has been observed that an Activity-Artifact Graph (AAG) is the best tool for approving existing and newly proposed process of development and thus, it has been used for verification of our CLC also. We have drawn AAG for each phase of CLC implying the successful tailoring of CLC process. It has been concluded that CLC will be helpful for developers having an integrated development life cycle (in usability perspective) and will be extremely useful to bring in software usability since conception of the software. Also, this contribution will be a step forward to bridge the gap between usability and software engineering practice.

Finally, we conclude that we have attempted to achieve the stated objectives by investigating the impact of some usability issues in software development environment. Our research has wide scope and also led to further exploration of possible research in usability of software. In addition to generate theoretical interest, the methodological and algorithmic procedures in the proposed research have potential applications in future enhancements in following areas:

- Investigation of effect on cost and effort of a software product developed with perspective/ inclusion of usability.
- Expansion of investigations in context of software development on cost and effort especially for each phase of the software process.
• Investigating the impact of usability in view of other user considerations such as; age, experience, level of abstraction, psychology etc.

• More metrics and methods may be established. A generalized usability evaluation and measurement method may be devised.

• The attributes have explicitly classified with defined role, but does not account for the crosscutting attributes. Therefore, in view of crosscutting attributes, still there remains a scope of further improvements.