CHAPTER 1

INTRODUCTION

1.1 PREAMBLE

In today's highly competitive era and market demand, it has become necessary for software developers/companies to develop high quality software and at the same time to improve the quality of existing software. Otherwise, sooner or later, it may transform into unrecoverable tangible cost. There exist ISO9126 quality standards to identify quality attributes for a variety of software. Usability is recognized as one of these major attributes towards the operational/functional feasibility of software. Software usability is the degree to which the software is easy to use i.e. it should be understandable, learnable and operable [PRE04]. Users always have high expectations from software. These include mainly, easy to learn, easy to use system that help in work without getting slow down, should not be tricky or confusing, with no mistakes and facilitate easy job finishing [CON95]. Usability is also defined as a measure of the extent to which a computer system facilitates the learning, helps learner remember lesson learned, reduces the likelihood of errors, enables users to be efficient and increases satisfaction with the system [DWN99]. Usability is undoubtedly central to the area of Human Computer Interaction (HCI), which emerged with widespread usage of computer-based systems and especially studies design, implementation, and use of interactive computer systems. An extra attention is posed on effect of computers on individuals, organizations, and society aiming to achieve more usable system/software with reduced development time and costs [BHI96, SG96].

These reasons lead to ponder usability during software development. The importance of usability evaluation has been recognized since long time and found critical for the success of interactive software systems [TAO05]. It is thus desirable to notice improvements observed/realized using usability principles/considerations into software development.
Earlier, usability has been mostly associated with the interfaces only. Later, Web based software being developed with intent to cover wider range of users. Researchers were more concentric towards identifying and resolving issues related to Web usability for many years. However, through various findings it has concluded that usability has to be crucial feature for all types of software for their success and acceptance by users now days. Fundamentally, user satisfaction has to be kept at the highest priority to get benefits of diverse aspects related to software. And, user-centered analysis and design is the key for the success. Its many fold benefits include; improved maintenance efforts; reduced chances of errors in design and coding; reduced cost and efforts of maintenance; reduced software defects by 2/3, which can further be improved more; results in usable system with improved overall productivity; and on top of it, improved satisfaction of software users [Pl09a]. It emphasizes that usability must be placed at the top of the quality hierarchy and must be given full attention from conception phase of software development [CK03, DSR07, TAO05].

Many researchers have presented certain concerns related to the usability issues. Each one of them highlighted primarily on the essentiality of usability feature in the software and emphasized on the usability matters to achieve the usability in variety of software. There exists a representation which shows that usability challenges exist in various aspects in both content and functions. Many problems reflect a lack of user-centered design and interactivity in the software development. A well-designed system should consider the distinct information seeking behaviors of different users which is especially vital for any software [ZD05]. For usability evaluation, a multilevel classification of usability has been presented with inclusion of some issues. Concept of development, the appearance of the system, its structure, the navigation, interaction with the system, performance and effectiveness of the system being used are the issues mainly discussed [PAO99, LP98]. Usefulness being major concern of usability of any software has supported by the need to provide effective interface. And therefore, improve cognition, timeliness of information, to provide continuous access to information, availability of required information and correlated links to availability of information have been identified to be issues to obtain usable software [CK03]. Conflicts between security and usability goals can be avoided by considering the goals together throughout an iterative design process with security as major issue for usability of system.
design [YEE04]. It is observed that in addition to useful interfaces, there exist few more issues that contribute in the usability of software. These issues are namely; consistency, feedback, forgiveness, simplicity, familiarity, user control and flexibility, clearly marked emergency exits etc. [CHA98]. However, the usability issues are to be identified solely in context of software product in general applicable uniformly for all categories of software without arguments. To bring usability in the software at its early stages of development, it must be associated with the development process right from its conception. It provides a wide scope of identifying more usability issues associated with usable product. Till time, issues have been pampered for the product only. However, a strong need has observed for isolating issues related to the process for effective development in addition. This appears to be helpful in incorporating usability thereby total quality improvisation in view of usable software development.

Usability issues are viewed mainly from two perspectives; the process and the product. The Classical Software Development Life Cycle (CSDLC) has been used as the basis for this demarcation of the usability issues. The phases of CSDLC and the work products of respective phases have supported to isolate the usability issues for process and product. Repeating through all the phases of CSDLC in the same manner, all the issues have been distinguished [PI09c]. The usability issues investigated at the process level of software development mainly concerned with the process of usable software development. The important issues are namely; reduced efforts required to learn the system, reduced efforts required to interpret the system, providing motivation to users, helpfulness, understandability, simplicity, structuredness, ensuring everything behaves as expected, well documented, efficiency, affecting users in a positive manner, control on execution path, availability of support throughout the development, systematic representation, context sensitive usage, easy links to the other parts, easy progression, efficacy, stress free elaborations, visibility, feasible, define measures to control behavior of the system, adequate on-line help, structured representation of functionality, easy to understand functional components, easy to understand features of the system, feasibility to perform difficult operations, self explanatory, self documenting, responsive, verifiable, meaningful, accuracy,
reviewable, repeatable, recoverable, testable, executable and fulfilling intended purpose of the process.

Correspondingly, a set of usability issues has been recognized at the product level i.e. for the software. It includes issues such as effective user interface, easy to use the product, easy operations on product, feasible to perform difficult operations, timely error messages, well documented, self explanatory user interfaces, self documenting user interfaces, responsive user interfaces, adequate on line help, adequate error message points, easy to learn, efficiency to use, helpfulness, understandable, availability of support (free or paid), simple, structured representation of functionality, context sensitive, easy links, easy progression, visible system details, easy to understand functional components, easy to understand features of the system, ease to use code, performing easy operations, assurance of thing behave as expected, verification, meaningful error messages, efforts required learning, efforts required operating, efforts required interpreting output of a program, easy to learn, affecting users, control on execution, tolerance, stress free use, visibility, easy to recall, emotional response, accuracy and correctness etc. Collaborating issues of process and product will lead to a successful development process thereby generating a successful product.

Current software quality levels in US result in software with approximately 4.5 defects per 1000 lines of executable code. Such level is unacceptable. In Japan the rate is improved with 1.5 defects per 1000 lines. Motorola and IBM have launched quality programs striving reducing the rate. In 1988, US Government Accounting Office surveyed for the success of software projects for their division and investigated 6.8 million of software. The results were ridiculous with facts that 47% (3.2 million) software were delivered but never used. 29% (2.0 million) software were paid for but not delivered. 19% (1.3 million) software were abandoned or reworked. 3% (0.2 million) software were used after changes. Only 2% (0.1 million) software were used as delivered. It has been concluded that it happens due to non-usability considerations [LOU93]. Few more statistics have been provided to justify usability principles in software development. Problem must be fixed during testing phase otherwise it would become 40-100 times more expensive, if these are fixed in
maintenance phase [BOE81]. Frequent changes requested by user, ignoring certain tasks, users lacking in understanding their requirements and insufficient communication and understanding between user and analyst are few reasons which exceeds the estimated cost of the software project. Without usability it is increased by about 63% [LP92]. Adequate efforts in early development stages reduced the cost and time intensive repair and correction tasks in the maintenance phase [RS92]. It has been observed that after the complete development if the changes are to be made it costs $400 while the changes incorporated during development costs $10 only [PRE92]. Considering usability during development has increased productivity by 25% and quality by 30% [GG92]. Similarly inclusion of usability in analysis and design process has increased overall user satisfaction by 40% [GG92]. 80% of the maintenance is due to unmet or unseen user requirements. Only 20% maintenance cost is due to bugs or reliability problem. Total maintenance cost of s/w development is assumed to be 80% of the total life cycle cost. Hence, 64% of the cost can be reduced with usability [KAR93]. Usability engineering if applied for system design, around 25% of the training time can be reduced. Errors can also be reduced from 5% down to 1%, if user centered design have been used. To improve speed of use and to reduce errors user centered design has to be used without which around 40 flaws have been observed in user interface [LAN95].

Revenues of DEC digital equipments have increased 80% for the new version that was developed using user centered design techniques [WJ95]. Another case was an insurance company in Australia, calls to its help desks reduced by 2/3 after one of its core application was improved using user centered design [NOR95]. In a study into 18 major companies in Australia, the cost of end user computing has been found to be $10,000 on average for a single work station. 50% of this cost is the hidden cost, hidden cost is the cost of productivity loss due to users supporting each other with computer problem. This 50% may become 0 using user centered design [KH95]. 60% of software defects result due to absence of usability. Only 15% are due to functionality hence if user satisfaction is kept in centre these 60% defects can be optimized [VPL96]. Without user centered design the average annual bill for supporting single PC is $13000. This is the cost of time amount spent by employees futzing with computers [GIB97]. New York stock exchange upgraded its core trading system with user centered design technology, productivity increased with a large margin and error
rates fell by a factor by 10 with workload more than double [GIB97]. In around 300 American companies, and for about 8000 software development projects, it has observed that without user centered design only 16% of projects were successfully completed on time and in budget, without all featured and functions as initially specified. Lack of user input, incomplete requirements and specifications and changing requirements and specification are the main reasons found. If user centered design is used, the percentage of success can be improved more than 50% [LINK01].

Further it has been observed that a good user experience will drive loyalty and trust in a positive direction without a price cut. This is the value which can be capitalized virtually at any time, versus price which is more of a difficult short term maneuver. Usability proves to be an extremely effective cost cutting tool. Usability helps designers create more simple products. Simple products are cheaper to build. Simple products are easier to sell and they are also easier for customers to maintain. Also, it drives down costs. Usability helps you maximize your investments. The wise person, armed with usability, knows that a good user experience is derived from simplicity. It may be noted as usability = simplicity = user satisfaction = increased profits. Usability can be used as a breakaway weapon to drive down costs, improve customer satisfaction, and increase speed to market. With usability spending about $20,000 could yield a savings of $152 million dollars, shown by Sun Microsystems. Each and every dollar invested could return $7,500 in savings. Usability is about hard core profits. Usability offered a strategic cost advantage that keeps developer and customer closer. They may satisfy better than their competitors. Usability offers sustainable competitive advantage [RHO00].

Users performed more quickly and successfully with the simple than the complex tasks. Even advanced, professional users were challenged by the most difficult tasks [KFBE01]. A study on website development has exposed that the economic effect of embracing usability promises faster user task performance. At the same time the probability of successful design/ usability evaluation was traced up to 90% as compared to the 60% probability of successful design without usability evaluation. Probability of better results due to usability evaluation was noted as 30% [KRI01].
1.2 STATE OF THE ART

The need of usability arises when faced with the kinds of questions during design such as; visibility, scalability, fault tolerance, mistake proofing, memorability, variability in user interfaces [NL95]. The effectiveness of Usability Engineering lies at certain levels of acceptance such as; Skepticism (the company is focused on product features and the development schedule. There is fear that usability evaluations will lengthen the development cycle); Curiosity (there is recognition that products need improvement, and a usability engineer is consulted late in the development cycle); Acceptance (there is recognition that usability evaluations must be an integral part of the process, and multiple iterations of design and evaluation are performed); and Partnership (consists of a cross-functional group working to gather early customer input to drive the product definition and scope, along with multiple iterations of design and evaluation) [NMS95].

The scope of usability is not just user interfaces. Rather, usability is conceived as the most general ergonomic quality concept that applies to all kinds of interaction between a user and a product within a given context of use [DW96]. In order to achieve the stated goal, the concept of usability and its core elements must be defined, taking into account different approaches and interpretations as a first step. Next, usability methods, tools and evaluation techniques embedded in industrial, ISO standards and research projects must be reviewed. Further, user features involving social and demographic, cultural and other factors crucial for usability of technology-based services must be analysed. And finally, current usability awareness and developments must be surveyed.

The need for user-centred design of systems emerged with the widening use of systems by users without specific education in computer science. Decreasing hardware prices, new programming styles and the rise of cognitive sciences encouraged a shift of focus from technology to human factors and their inclusion in the realm of computer systems design as an important issue. The broad scope of human context relating to computer-based systems determined the interdisciplinary nature of HCI, embracing a wide range of disciplines such as cognitive psychology, ergonomics, computer science, systems engineering and so on. Growing rivalry in the market for information technology, and user
acceptance of systems as one of the main success factors, led to an increasing focus on usability in the professional HCI community [SA97, HAR98]. Despite this increased focus on usability, there is no common agreement between HCI professionals on a definition of the concept, with diverse names employed to express what usability is or with different meanings being applied to the same term [AKSS90, BM02]. However, several widespread approaches to usability are discussed in the following sections. Initially, usability referred to such terms as ease of use, user-friendliness, and ease of learning that implied providing users with systems requiring minimum cognitive and physical effort to accomplish their tasks successfully [BHT03, B+09, BEV01].

For instance, the Institute of Electrical and Electronics Engineers defines usability as the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component [KK03, AKSS90]. In order to design a usable system, it is argued that there should be measurable criteria to refer to during the development process [BEV99a, BEV99b, COV02]. Following this view, attempts to define usability have been tailored to the creation of metrics. Some of the pioneers of usability research, B. Shackel and J. Nielsen, included usability in their models of general user acceptance of the computer-based system and presented it as a set of attributes required to design a technology that would be accepted by the user. Many of these attributes identified features that can be placed in different hierarchical levels, with some overlaps; and some of them emphasize specific aspects. [LINK02, LINK03].

By raising the issue of technology as a universally accepted mediator in any field of human activity in the information age and migration of public and commercial services to digital space, Shneiderman suggests a concept of universal usability that would guarantee successful utilisation of technology by any citizen. In this context, he offers not a description of usability components but a broad research agenda. Shneiderman goes beyond universal access and builds his model of universal usability on three main issues, or challenges namely; Technology diversity, User diversity and Gaps in user knowledge [SH01].
In the literature on usability, two common approaches to usability are met. First, Product-oriented approach where usability is treated as a set of requirements that software or hardware products should meet in order to become usable. Second, Process-oriented approach that puts a focus on the integration of usability issues into the process of development to ensure a usable system at each stage of production and use. Both approaches were introduced in a set of ISO standards and led to the development of a new understanding of usability. The ISO 9241-11 definition points to the context-specific nature of usability that is described by users, their tasks and goals, environment, equipment, and is measured against criteria of effectiveness (i.e. how goals are achieved with accuracy and completeness), efficiency (i.e. comparison of expended resources to accuracy and completeness of goals achieved) and satisfaction (i.e. subjective user attitude to the product) as shown in framework presented through Fig.-1.1. In different situations, the context of use will vary, so there are no universal usability attributes that would prescribe what a usable product is. There is rather a common set of activities that would help to identify what needs to be done in a particular situation for a certain product to become usable.

Modern usability approaches represent miscellaneous attempts to delineate limits and organize the content of the field. However, the initial goals for which usability is defined precondition a fragmented view. Thus, industry, driven by commercial benefit, usually concentrates on those usability aspects that would influence sales; developers of standards and practitioners seek to provide tools for real technology development processes; while other models present a broad framework for usability in the modern world. Although the value of all efforts in the usability area is not in dispute, the benefit would increase if all approaches were to be integrated. More practical approaches to usability, dealing with the development of measurement systems, refer to the operational level and provide tools for implementation of strategic goals. Detailed analysis of the described models leads to the conclusion that they are complementary and that they should all be considered when making decisions concerning usability, on both strategic and operational levels. The idea of usability as an integral part of organization performance expressed in ISO 9241-11 is emphasized by ISO 13407 in its description of human-centred design processes. [ISO98].
Fig.-1.1: ISO 9241-11 Usability Framework
Life Cycle includes continuous management of the final product at the stage of actual use [ERGO0, ERGO2, ERG98]. User feedback mechanisms are established in order to collect valuable information for future products and to increase the business potential of the current offering [BHT03]. The great number of usability methods with different names for similar or the same techniques, and diverse grouping criteria (either according to their application during the project or principles of gathering information), makes an overview quite complicated [NIL94, FB04]. However, arrangement of methods according to the project lifecycle doesn’t take into account repetitive application of the same techniques while addressing diverse usability issues related to product and process both [MM98, NIL97, NIL95, RFR95, RII02]. Also, categorization according to aims and procedures does not refer to methods that can be assigned system development, such as prototyping or scenarios. Similarly, proposed classifications do not provide a framework for rational choice of relevant usability methods in the course of a project. Due to complementary roles, some methods are not suitable for selection. For example, it is impossible to choose between expert-based usability evaluation and competitor analysis [ZBS99].

Nevertheless, with some improvements, existing views can be categorized in a more rational manner. The most mature approach is demonstrated in classification of usability methods according to their procedures, applied techniques and tools to usability inspection, testing and inquiry. This broad group of methods perform a single usability assessment function [RFR95, RII02, ZBS99, NIL95, UG04]. Cultural, age, ability along with some other factors poses impact on the product usability or service usability as well. These may be thought of as horizontal factors and influence any of the groups of users discussed above. Moreover, there is a strong correlation between these complementary items. Some challenges have been identified mainly; Technological variety is a challenge for providers of quality services that are usable by various target groups due to multiple standards and platforms that often are incompatible. User knowledge refers to a set of miscellaneous skills enabling the user to interact with the system in an effective and efficient manner. Gaps in all kinds of necessary knowledge for successful use are affected by income that may prevent or lower the quality of the interaction [LEV98]. Often low-income persons cannot afford a necessary
technology though the demand for job market is increasing. Problems of low income and a broader context of social inclusion cross the whole spectrum of usability concerns but have an influence on the minimum features needed for the interaction [LEV98, SHA03, MEU07].

In view of the above, it may be stated that there remain ample scope of research in the field of usability in terms of issues, common platform, generalized methods and investigation of factors responsible for its impact in software development environment.

1.3 OBJECTIVES

Literature review reveals that still there is a scope of investigating various usability issues in software development environment. Although usability has appeared as a significant feature of software, most of the usability research is inclined towards product’s usability. Observations articulate the fact that the process of development is vital and influences the ultimate product highly. In light of this fact, along with the usability issues related to the product, issues associated with the process must be investigated to bring in software usability. Our research aimed at investigating these issues to strengthen usable software development.

The objectives taken up in this thesis are as follows:

Earlier, usability has been considered merely a software quality characteristic. Later, in last decade or so, it has gained immense substance as it dealt with the efficiency, effectiveness and ultimately; user satisfaction. The first objective is to review various software quality attributes’ classifications and relationship and to propose a generalized classification of software quality attributes. Also, based on the established knowledge of usability, we attempt to classify usability attributes on the basis of generalized classification. Since the change in view alters the purpose, sequence and role of the software quality attribute, we have attempted for a comparison of above mentioned classifications.

The backbone of any software development is the requirements without which it is not at all possible to develop the software. At primary level, requirements are the users’
expectations from the software and to fulfill these requirements, more requirements may be generated at the developers' level in terms of resources, technology, methods, procedures, tools etc. Since usability has to be incorporated from the conception stage of the software, it is desirable to determine the usability requirements of the developing software. Our next objective is related to propose a classification of usability requirements. Usability issues and usability attributes are the foundation of the proposed classification. Also, with the help of usability requirements we have identified various stake holders and their respective usability requirements. Association of their usability requirements is noticed to identify key concerns of usability. Key concerns are supposed to be the vital factors performing usability evaluation and measurement simultaneously.

Most of the software estimates are being assessed for completed product. A major estimate is required to provide quantitative measure. This is the project influence and no evidence is available about it. Thus, next objective is related with computation of project influence on usability attributes in software projects. In this regard, an Influence Assessment Method is proposed to compute project influence and average project influence in domain of software projects. We conceptualize some of the terms based upon fundamental project parameters required for the method. Primarily, these fundamental project parameters are used to characterize software projects. We develop a mechanism to assess influence of project parameters on usability attributes along with assignment of weights to influence levels of project parameters. We study some live projects to realize the project influence.

Usability attributes possess specific characteristics/property related to usable software and act as decisive factors in process of development. Therefore, another objective is related to rank usability attributes to understand role, behavior and precedence of these attribute in usable software development. We propose an algorithm "CompuRank" to rank the usability attributes empirically. Also, generate new terms are defined to rank the usability attributes using algorithm. We propose another survey based approach for ranking usability attributes with the help of opinion of software professionals. The association of ranks obtained by using above mentioned approaches is also presented.
Scenarios are well proven instruments for user centered development and have sufficiently recorded the utility for assessment of development related qualities. Thus, using scenarios being strong tool we propose another Scenario Based Ranking Method (SBRM) to rank usability attributes with quantitative measures supporting computation of ranks. On the basis of the results, the unification of ranks computed by using algorithmic approach and ranks determined using SBRM is presented.

Another objective is to collaborate the classical software development life cycle (CSDLC) and usability engineering life cycle (UELC). We have proposed a development model based on which the collaborative development is achieved. Here, our objective is to stitch the phases and activities of both of these cycles in a collaborative manner so as to make them processed simultaneously. It will reduce the efforts and time consumed in performing activities of these two cycles independently and will assure the systematic development of a usable software from the conception phase.

1.4 OUTLINE OF THE THESIS

The thesis is organized in eight chapters to cover the research issues related to the usability of the software especially emphasizing on the aspects of process and product usability. These aspects mainly include establishment of understanding with usability, current trends in usability, usability methods, requirement of usability etc. A general overview of the said research field is provided along with its various viewpoints. The related state-of-the-art is presented subsequently. Also, the objectives of the proposed work have been mentioned in this chapter. We provide the literature survey along with our view to the usability research domain in the thesis. However, the related recent literature has been covered in the starting of each chapter. And, at the end of each chapter the summary including findings of the respective chapter is included.

Chapter 2 deals with the review of classifications of software quality attributes. First, we propose the generalized classification of software quality attributes based on various available classifications. In this classification, usability has been observed to be only a quality factor. Subsequently, the generalized classification has been reviewed in usability
perspective. Therefore, we propose another classification of usability attributes in various classes and sub-classes in context of usable software. It has been observed that the modification of view and organization changes the role and behavior of quality attributes. Therefore, the comparison of these classifications has been presented consequently to show the dynamism of the attributes in different perspectives.

In Chapter 3, mainly a classification of usability requirements has been proposed. First, we have discussed about usability requirements and their importance in usability assessment process in an iterative manner. Subsequently, an attempt has been made to classify the usability requirements based on the usability issues and usability attributes. Also, we have provided viewpoints of different stakeholders of the software such as: User, Developer and Researcher. Further, we have attempted to find association of requirements of the identified stakeholders of the software by computing coefficient of association. Based on the results, the most common requirements have been identified and termed as Key Concerns (KCs). These KCs prove to be vital for further usability investigations.

Chapter 4 illustrates the computation of project influence showing importance of usability attributes in software development. We have conceptualized new terminology along with the definition of existing fundamental project parameters. Further, we have characterized projects based upon the terms defined in this chapter. A mechanism supporting a multitier structure of influence assessment and assignment of weights is discussed subsequently. This mechanism is central to the computation of project influence. To compute the project influence, an Influence Assessment Method (IAM) is proposed and steps are described in consequent section in this chapter. Next, the working of IAM with a case study using various schemes of weights of influence levels in some projects has been presented. In order to illustrate the computation, projects are simulated with reference of actual software projects. The APPENDIX 4.C at the end of the chapter consists of description of the real projects taken for the case study.

Chapter 5 leads to encompass ranking usability attributes. The approaches proposed here employ empirical methods. Thus, we have first described the empirical methods
including their classification and suitability in research, selection criteria and their importance in research. An attempt has been made to conceptualize new terms required for the said purpose with review of existing parameters. The mechanism of dependency has been tested for random weights and it has been noticed that it is not suitable in order to define dependencies of usability attributes in any software. Therefore, we refer to the mechanism defined in this chapter. The first approach is an algorithmic approach which has been established with informal and formal description of proposed algorithm “CompuRank”. Its working has been explained with a case study. The algorithm has been executed with various schemes of weights and the results are listed in APPENDIX 5.A at the end of the chapter. The algorithm may be further generalized for any number of software project parameters. Another approach namely; a survey based approach is also proposed in this chapter. It mainly deals with the survey of software professionals with varying experience in three groups to assess ranks of usability attributes. Next in the chapter, we have attempted to find any association of the rankings obtained by using algorithmic approach and survey based approach. The discussion on the results is presented in the chapter subsequently.

In Chapter 6, we have presented the definition of scenario with its importance in development related quality assessment. A Scenario Based Ranking Method (SBRM) is proposed for computation of ranks of usability attributes. Again, with the help of a case study having four cases of acceptance of scenarios, we have demonstrated the results obtained using SBRM. This method has been executed with various schemes of weights of influence levels in each case. The scenarios defined for each attributes and detailed result are attached as APPENDIX 6.A and APPENDIX 6.B respectively with this chapter. Further, comparison of ranks computed using “CompuRank” and SBRM has been performed with intent of unification. The results reveal the success as discussed at the end of the chapter.

Chapter 7 deals with proposing a Collaborative Life Cycle (CLC) that basically collaborates Classical Software Development Life Cycle (CSDLC) and Usability Engineering Life Cycle (UELC). We have first discussed the phases of CSDLC and UELC in this chapter. Prerequisite usability factors in each phase are described categorically. Further, Collaborative Development Model (CDM) has been proposed to exercise CLC. Next, we
have applied the Activity-Artifact Graph (AAG) for tailoring and verification of CLC. We have discussed necessary AAG definitions and used in order to show the conformance of CLC. For addition of activities, artifacts and for merging independent activities, we have refined the available algorithms of procedures. The resultant AAG of each process module and inter-module relationship are drawn for clear understanding of the work. The detailed activities, artifacts and relations in each phase (process module) in CSDL, UELC and CLC are listed systematically in APPENDIX 7.A, APPENDIX 7.B and APPENDIX 7.C of this chapter.

In Chapter 8, we have presented the concluding remarks on the contribution of work presented in the area of usability especially in software development environment in this thesis.