Critical Review of Extended Waterfall Model

Rajesh H. Kulkarni, P.Padmanabham, K.K.Baseer

Abstract—Software product quality improvement is a desired attribute and a strenuous effort is required to achieve that.Usability is also a desired attribute as it helps in identifying how effectively user achieves product goals.Concrete efforts to integrate Software Engineering and Human Computer Interaction exist in the form of models by many researchers. Better user experience is an oft expressed and desired quality of the products designed nowadays.Many efforts in this regard lead to various proposals of smooth integration of SE (software engineering) processes with HCI (human computer integration) for product development. One such effort is extended waterfall process model. This paper presents a critical review of extended waterfall model. It also suggests means to bring nearer the two diverse communities of SE and HCI.

Index Terms— Usability, SE, HCI, Extended Waterfall Process Model.

1 INTRODUCTION

The model of waterfall process refer Figure 1 in the field of software engineering [1] has been playing a chief role from the past. This model was put forth in 1950s and later, it has turned out to be a renowned model in 1970s. Its structure takes the form of a cascade involving phases and it can be viewed that the output from one phase serves as the input to the subsequent phase. If a single phase is considered, there will be a group of actions that several people may carry out in a simultaneous manner. The idea behind the waterfall process constitutes the following. (i) A linear and sequential path may exist between the consecutive phases, (ii) The standards may be maintained at places, where the outputs or deliverables could be generated or (iii) Few techniques, which aim to achieve the necessary output, may be suggested. The waterfall model imparts numerous benefits.One such benefit is that it could legibly recognize the number of related phases through which a process in software development should traverse.

These phases may own varying arrangements along with few changes in the scope as well as the significance and hence, there is a possibility for these phases to occur in more than one process models. Moreover, the phases involve the following: (1) drawing out the requirements for specifying how the system should look like at the end, (2) examining the requirements for sorting it out into functionality group and non-functionality group (for instance, the features), (iii) making an architectural design for the system as well its corresponding elements, (iv) programming and executing the system, (v) substantiating the performance of the system and its associated elements and (vi) subjecting the system to work in an operational background [2].

- *Rajesh H. Kulkarni is currently pursuing Phd degree program in computer engineering inJ.N.T University, Huderabad and working in JSPM NTC Pune,India. E-mail: rkpv2002@gmail.com*
- *P.Padmanabham is currently Director Academics in BIETin J.N.T. UniversityHyderabad,India,. E-mail: padmanabham46@gmail.com*
- *K.K. Baseer is Phd Scholar in JNIAS-JNTUA, Anantapuramu, India, India*
  E-mail: kkbasheer.ap@gmail.com

Waterfall model is classic SDLC model. This model is also called as a heavyweight process. The lifecycle is long say six months. This classic model consists of the phases: Requirement, Design, Analyze, Code, Test,and Maintain. There are other models such as Agile and RUP. In Agile Development the lifecycle is very small say two weeks. Requirements are freezed. A scrum call is issued for not more than fifteen minutes where what today's tasks to be done are discussed and scrum master takes the onus for that. It is also called as a light weight process because of shorter span of lifecycle.In practice none of the models are followed to the tee. Every project in itself is an experience. If bracketed each project is a representative of its own unique model. The reason being Requirements are given yesterday night and the deadline for deliverable is within a week. It is a chaotic scenario. Most of the time is spent on negotiations between client and vendor.As per Carroll et al. "Human-computer interaction (HCI) is an area of research and practice that emerged in the early 1980s, initially as a specialty area in computer science embracing cognitive science and human factors engineering. HCI has expanded rapidly and steadily for three decades, attracting professionals from many other disciplines and incorporating diverse concepts and approaches. To a considerable extent, HCI now aggregates a collection of semi-autonomous fields of research and practice in human-centered informatics."[18, 19].

The implementation and the functioning of a system will be unfeasible without an integrated software engineering environment, which is capable of offering stability to handle the process as well as the product. There are three layers, namely, the collaboration layer, the information layer and the operational layer. The collaboration layer relies on the process management techniques for assisting the combined task. The information layer is one among the three layers supporting the ASP system. Yet, the traditional product management systems have their centre of attraction towards the source codes or else on excellently organized design details like, the transition diagrams and the data flow diagrams. The operation layer acts as the ASP system's infrastructure.

2 BACKGROUND

The domain of software development has undergone a tremendous progress in the past forty years of research. In 1970, Winston Royce's has granted a popular publication on the present day waterfall scheme [3].It is this approach that has brought about the transformation in the area of software development, in which the software development is imagined to be extremely complicated and user-specific that a number of shortcomings may arise with improper arrangement.The waterfall process model serves as the conventional and typical means for developing the software.In the Waterfall process model, the significant phases taking part are Communication, Planning, Modeling, Construction and Deployment.Further, this model comprises of iterations in the form of system development life cycle (SDLC) phases.The software developers are rendered with plenty of benefits from the waterfall model.The first benefit is that the staged development cycle imposes control because a phase will definitely have a beginning and ending, which allows the vendor as well as the clients to fix goals to determine the advancements in a decisive way.The time spent and the effort applied gets degraded because the requirements and design are demanded prior to the execution of a single line of code, thus deviation from the schedule is prohibited.Acquiring the requirements and design at the initial stage grants several benefits. One such is the enrichment gained in the quality. In addition, the flaws that are likely to occur can be disclosed and rectified in the design phase itself. Hence, performing flaw detection and correction in the testing phase, where the entire number of elements have undergone integration and locating certain errors can be difficult, are eliminated. The two phases in the beginning result in a formal specification and thus, the function of waterfall model is to assist in conveying the knowledge to the team members distributed in various places. The application of waterfall process is feasible, if and only if the complete requirements exist for re-engineering the presently available SW. The commercial projects cannot use this model practically due to the presence of iterations among the phases. Moreover, the price and duration spent for system advancement and implementation is also large. One more hindrance comes from the working version because of its existence at the time of implementation. Failure can surely occur in cases, where the requirements of the customer are ambiguous-natured. Gathering the entire requirement details is highly unfeasible in an earlier stage in the project and this model works only after the requirements are furnished [4].

Critic of Waterfall model and SE/HCI gaps derived from [10]:

- product takes too long to develop with the waterfall model(hanna,1995)
- real projects rarely follow the sequential flow that the model proposes, even with theiterations (Pressman, 2005 p. 79)
- treat it as strictly linear model and iterations are regarded as wasteful
- waterfall model with feedback loops leads to idleness of team members
- Kroll and Kruchten say that the waterfall model with feedback loops leave a lot of team members idle for extended periods, that it still defers integration of code and testing until it is very late and when problems are harder to resolve, and hence is poor at managing risks (Kroll, et al., 2003 pp. 6, 50).
- SE is relatively immature in the requirements area (Suitcliffe, 2005).
- Language for communication is an issue. Improved communication is possible through prototypes, sketches, mock-ups, simulations, and scenarios er (Gulliksen, et al., 2003).
- Communication techniques are basic. The oft-used phrase "requirements gathering" conjures an image in the mind's eye that requirements are like flowers fallen overnight under a parijatak tree, and one needs to only gather them for the morning pooja .
- Instead of asking users what they want and putting it into the design blindly, it is much better to directly watch what users do, identify design problems and opportunities and then design a system that solves the problems and realises the opportunities.
- Conscious design decisions can be taken during communication phase following HCI design process.
- SE literature does not acknowledge that that requirements specifications already specify HCI design decisions.
- HCI aspects are considered superficial and SE literature does not mention or demonstrate the importance of considering alternative HCI designs.
- There are major gaps of communication between the HCI and SE fields: the architectures, processes, methods, and vocabulary being used in each community are often foreign to the other community." (IFIP WG 2.7/13.4 on User Interface Engineering, 2004).. There is a need to have shared processes, common techniques, nomenclature, checkpoints, and measures for success (Jerome, et al., 2005).
- The HCI practitioners must have SE process support before they can deliver good quality usable software.
- There is a deep- rooted myth that usability is not a central topic of SE.
- SE and HCI practitioners rarely collaborate and speak different language.
-

Critic of Waterfall model derived from [11]:

- Time and cost estimation is extremely difficult
- Upfront requirement capture and design is not realistic and suitable for real world
- Assumption that design can be easily translated into products is not feasible
- Division of responsibilities is not efficient

Critic of Waterfall model derived from [12]:

- The waterfall model is not appropriate for many practical project situations.
- Software development is less linear than the waterfall model suggests.
- There's can be a disconnect between organizational process mechanisms and the reality of the project.
- Cultural factors within the organization can inhibit resolving the disconnect between the organizational process mechanisms and the realities of the project.
- An imposed process model can start your project out at a disadvantage

3 LITERATURE REVIEW

In the literature, the discussions on various phases involved in software engineering like, communication, modeling, planning, construction and deployment [5] can be seen.Hutchinson et al. [6] have dealt with a development process model containing four stages, which depends on the components.The implementation of this model was naturally found to be more complicated. Instead of employing house development, the integration of the off-the-shelf components with the freshly developed components was the central theme behind this model and it has failed to make use of the repository.Costabile [7] (see Figure 3) has discussed the means with which the HCI activities and the waterfall model can be integrated.Constabile's was the first process-based effort for revised waterfall model which was the source of inspiration for Anirudha Joshi's extended waterfall model with subtle differences.Constabile revised traditional waterfall model of software engineering by incorporating usability activities. This he believed will lead to user-centered design which is the need of the hour. Constabile's take on this revision is being put as is from [10],The shadowed boxes in the figure indicate some activities to be performed in order to shift from the system-centered design typical of the classical waterfall model to user-centered design that may lead to designing usable systems.

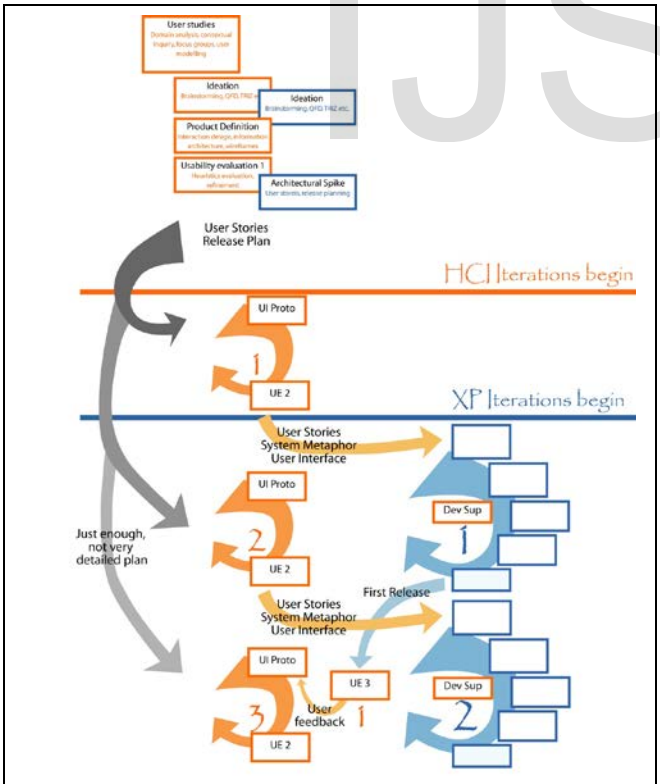

Fig. 1. Extended Agile Process Model extracted from [30]. A schematic overview of HCI activities integrated in Extreme Programming. Activities in red boxes are new HCI activities proposed, while blue are SE activities

system, the tasks they perform and the environments in which they work.

The best way to collect this information is by visiting users at their workplace, observe the way they carry out their tasks, and talk to them through different types of interviews [10]. The user analysis will also affect the general design of the system architectural design and detailed design should explicitly include the design of the user interface, which is not anymore deferred to the end of the system development.

Indeed, the user interface is the most relevant part of the system from the users' point of view, and its design should be discussed by the designers directly with the users since the very beginning of the life cycle. As indicated in the shadowed box 2, use scenarios [10] that are a sequence of steps describing an interaction between a user and the system may help figuring out the design of a usable interface. Constabile says, "The current need is product design should suit user needs globally. Hence design approach should be user-centered. The basic principles of user-centered design are: 1) analyze users and task; 2) design and implement the system iteratively through prototypes of increasing complexity; 3) evaluate design choices and prototypes with users. User-centered approach requires understanding reality: who will use the system, where, how, and to do what." [10].Joshi and Sarda [8] have exploited IoI for integrating the HCI activities and the waterfall process model.They have advised to involve the investigation of the competitive product analysis, contextual user studies and user modelling, product definition/information architecture/wireframes with a multidisciplinary team, ideation with a multidisciplinary team, enrichment of product description and formative usability assessment in the communication phase.In addition, they have insisted to make an inclusion of comprehensive user interface prototyping, enrichment of prototype and formative usability assessment of the user interface in the modelling phase of the waterfall process model.At the time of construction phase, the usability team should perform re-assessments and aid the development team. Moreover, a summative usability assessment should be performed, if a fairly large fidelity version is found to be present. Anirudha Joshi et al.[9] have recommended two metrics for describing the influence of integrating HCI activities and SE processes.Usability Goals Achievement Metric (UGAM) refers to a product metric, which gives the measure of the degree to which the product design satisfies the objectives of user skills.On the other hand, Index of Integration (IoI) points to a process metric that is capable of measuring the level to which the HCI activities and the SE processes get integrated.These two metrics have been produced from the standpoint of an organization and hence, they can find applications in vast quantity of projects or products.In addition, a trial on how the metrics can be utilized without difficulty in an industrial environment has been dealt.Since the two metrics have higher correlation to each other and allows a successful integration of HCI and SE processes, plenty of other methods can also use them in an identical sense.The assessment of these two metrics has employed three studies. The two metrics were evaluated in three self-governing studies and they are, a classroom-dependent evaluation consisting of two sets of students, a qualitative feedback from three industrial projects and a quantitative evaluation with 61 industrial projects.The two metrics allow the process to be more organized, since they are beneficial and less complex.Seffah et al speak about the dichotomy that decouples usability from software develop-

ment life cycle [13, 14]. The dichotomy arises from the diverse psyche of usability practitioners and software developers as mentioned in [13, 14]:

"• We, the engineers, the real designers of software, can build reliable and safe software systems with powerful functionalities. The usability people, the psychology guys, can then make the UI more user-friendly.
• We, the usability professionals and interaction designers, should first design and test the interface with end users. Then, developers—the functionality builders—must implement the system that supports the user tasks."

They speak about cross-pollination, educating usability and Software developers to work together and the need of developing computer assisted usability engineering tools.
Len Bass et al speak about "benefits of linking usability to architectural design" [38]. The figure 2.5 elaborates those benefits.
Ferre et al speak about bridging the gap between usability and software engineering practices [61, 63].
Important features of Star Life cycle as per Hix and Hartson (see Figure 4), are that "evaluation is at the centre of activities, no particular ordering of activities; development may start in any one and it is derived from empirical studies of interface designers" [20,21].
Usability is defined in Part 11 of the ISO 9241 standard (BSI, 1998) as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"[6]. As per Constabile "Effectiveness is defined as the accuracy and the completeness with which specified users achieve specified goals in particular environments. Efficiency refers to the resources expended in relation to the accuracy and completeness of goals achieved (it is similar to the productivity factor that characterizes quality in use in the ISO/IEC 9621-1).Satisfaction is defined as the comfort and the acceptability of the system for its users and other people affected by its use."[10]. The Star life cycle concept, derived empirically from extensive observations of real world developers [Hartson and Hix, 1989], is an evaluation-centered iterative usability engineering process for top-down, bottom-up development, or inside-out development. The primary goal is to support continual evaluation and iteration during user interaction development, including much tighter, smaller loops of iteration than imagined in the spiral methodology (see next section). The Star life cycle minimized the number of ordering constraints among development activities. For example, developers did not have to specify all requirements before working on design. In fact, developers sometimes started by exploring design possibilities — perhaps by using a rapid prototyping tool and doing walk-throughs with clients and users — and, in the process, learned a lot about requirements. The resulting evaluation-centered life cycle concept,[adapted from Hix&Hartson, 1993], was termed the star life cycle, because of its shape. The points of the star are not ordered or connected in a sequence. This means that a user interaction developer can theoretically start with almost any development activity and move on to almost any other one. The various activities are highly interconnected, however, through the usability evaluation process in the center; results of each activity are evaluated before going on to the next activity. In general, a different kind of evaluation is required after each different activity in this life cycle. Conventional life cycles lean toward independent performance of each development activity, whereas the star life cycle supports interdependent, but distinct, activities. Evaluation centeredness is a very important property of the Star Life Cycle, which (LUCID/Star)*inherits. In (LUCID/Star)*each product evolution goes through a cycle of four basic activity types derived from the LUCID framework. Cycle completion is dependent on the results of a final Evaluate activity and satisfaction of the cycle exit criterion. Most of the time these two things will be the same, meaning that the exit criterion is based on attaining some predetermined result from the final Evaluate activity. This implies that each evolution of the product could be evaluated possibly many times before moving into the analysis of the next evolution.

4 CRITICAL REVIEW OF EXTENDED WATERFALL PROCESS MODEL

After personally observing the two (SE+HCI) communities the conclusion derived is they are poles apart in their thinking, language, choice of tools. Our belief is success of extended waterfall model is dependent on systemic efforts to bring these communities together or at least reduce the gap and bring them a bit closer. The effort can be in the office space allocated to two diverse communities may be nearer. HCI contribution in a project may be highlighted and apprised to software community. As agile needs top-down support same is the case with merger of these two communities. Executive and management support should also be derived. Software industry has to understand and recognize significance of design goals.Efforts like bringing in Usability expert for greenhouse project can show a positive impact. He will be instrumental in data gathering and understanding the user requirements. This experience for software community will lead to a conclusion that usability is a very much in thing and having a usability person in the software development team is very much desirable.Industry software experts opine that there is scope for merger of requirement gathering process and HCI by bringing in a Usability expert during requirement phase. Business Analyst may question the wisdom of having a usability man when BA is there. One solution is Structure the processes in such a way that document should have inputs from both the BA and Usability man. There is no role or minimal role of HCI/Usability during maintenance phase of SDLC.There is scope for creating processes and documents wherein the requirements are user/client driven. There is scope for creating a template on the lines of Volere which includes HCI/Usability components embedded in requirement template.A senior project manager in a software company shared his experience saying, "Requirements are given by the client. Development team works on it and as per the deadline produce deliverable. Most of the times deliverable was not as per client specification and it was returned back. Then again the requirements were resent and this iteration went on till frustration level. Now after inception of Usability in SDLC requirements are clearer."In academics if HCI community could lobby hard and convince Pressman to dedicate one chapter at least for this new modelthen there will be a huge impact. Already we see effect of adding usability chapter in

pressman's text. Project groups should be constituted in such a way that HCI will complement the SE effort and get recognized.In the Extended waterfall process model (see Figure 2)[5], the processes exploit five phases, namely, communication, planning, modelling, construction, and deployment.Few works related to these phases are described below.Communication phase: Pressman [5] has stated that there are two activities in the communication phase and they are the project commencement as well as the collection of requirements.In the case of requirements engineering, activities like inception, elaboration, elicitation, specification, management, negotiation and validation takes place.Of all these activities, the requirements elaboration, validation, specification and management play a vital role in transforming the informal depictions into formal technical models of software attributes, tasks, limits and quality conditions, in addition to handling them in the whole development procedure.Yet, these are interior activities in software engineering.On the contrary, the requirements elicitation and the requirements negotiation are external to software engineering because they are responsible for accomplishing the communication between the users and developers, which in turn are subject to numerous alterations.Planning: Though the HCI activities are integrated to the planning phase of the waterfall model, the HCI metrics are recommended while assessing, organizing and following the HCI activities.Modelling: This phase makes a change from reality to virtual and in waterfall model, it owns the actions related to analysis as well as design. For best understanding, the "requirement analysis" and the "software design" are named again.Requirements analysis activity refers to the transformation of the requirements into an intermediate analysis model, which indicates the activities, information and purpose.On the contrary, the analysis model lies between the reality and the virtual environment, denoting a design model.Requirements analysis shows increased interest towards the formal description of a hierarchy of domain classes mirroring the actual world, their tasks, the events influencing them and the bond existing among them. The UML language is used to represent the models in a formal sense.Construction: Coding and testing forms the main activities of this phase and they have been widely studied in Pressman and other SE literature. Deployment: The HCI design team takes effort in gathering the benchmark data from the domain of usability metrics, which include the assessment of HCI effort, user performance along with the return on investment on HCI activities, so that it will be well-suited for the study-activities of the user in the near future.During the analysis of the extended waterfall process model, additional occupations that involve the working of individuals as teams have been created due to the team's effectiveness. Software is much demanded in several fields because of the popularity of the computers.There will be some kind of dissimilarity or uniqueness between these fields. Hence, the software development process model should be constructed in a way that it is appropriate to be applied for a specific issue through the analysis of the features related to that particular field.Software development is required compulsorily and it should be of improved quality. The present-day software extended waterfall process theory emphasizes that the people should never be more particular about their work than just get pleased with the attainment of a universal model.Non-universal models have to be constructed for executing it on a specific issue of a specific field of interest. In this way, the pertinence of the model can be enriched.The history of software development states that the extended waterfall process model has been introduced in the same manner. Though the extended waterfall model has been used prevalently, issues have aroused due to the vagueness and continuously varying capability of the user's requirements. The waterfall model has found difficulty in tackling the new troubles.So, the extended waterfall process model was proposed after examining the waterfall model and now, the new problems can be evaded away. The problems include vagueness, unpredictability, imperfection and the inability of natural language to define the user requirements.These problems will result in poor software quality. The extended waterfall model has proved that it was capable of eliminating these problems in a mathematical way. In addition, the extended waterfall process model is highly demanded to develop software, since the current technologies can be optimally utilized.Moreover, the process of developing software has strong connection with the techniques as well as the tools involved in constructing the model, particular methodology and the tools aid a particular model. If the people make use of innovative technologies as well as tools to accomplish the software development, new software development process models emerge automatically based on the novel techniques and tools.

Specific Changes Suggested

Specific changes required in extended waterfall process model: Communication is the first step in extended waterfall process model. In this phase, user studies about the project and the evaluation strategy are two intermediate components.Through this phase, requirements specification is formulated after defining the product specification. The definition of standard protocol for communication is missing in this communication phase.The escalation and the inquiry service process module are important for this phase which should be additionally included to further improve the communication phase. In communication phase, delay or communication gap is major factor which can be also major problem in extended waterfall process model.This can be effectively handled with optimally designing the policies for communication itself.Planning is an important component for the software development cycle. Here, estimation and scheduling played a major role to track the progress. The estimation is much important to formulate the process and schedules. In extended waterfall model, estimation is usually done with usual statistic dependent procedures but intelligence estimation can be effectively included.Also, comparative and relative prediction is much needed component in planning phase of extended process model. This additional component is missing in the extended process model. Also, scheduling and tracking the process flow can be further strengthened with effective intelligence methods. Along with three major components, incremental estimation can be a further component to be included for automatically identifies the behaviour of the system over the period of time. Modelling is the third component of extended waterfall process model. Prototyping, usability evalua-

tion, requirement analysis and software design are major steps included in this modelling phase. The usual procedure of modelling is with the metrics which can screen out the current progress of the system. The modelling phase is more related with mathematical design of the complete process.Statistical test can be included as addition step in the modelling phase of extended waterfall process model because theoretical validation in modelling is completely missed in extended waterfall process model. In construction step of extended waterfall process model, development and validation are two major components in this phase. Resource management is a critical factor in development cycle. This can be included as additional component in the extended waterfall process model to obtain better results. In Deployment, delivery, support and review are the important components in the deployment phase. Here, analysis is much important for analyzing our strengths and weakness to further improve the quality. This particular step is missing in the current extended waterfall process model.

5 RESEARCH GAPS AND FUTURE DIRECTION

In software engineering, waterfall model has been an essential task for years before and it was introduced with the aim of controlling the software development process.But, this model offers increased inflexibility. The lack of flexibility in the cascade and linear model was not that much easy, if the usage of feedbacks as the fundamental elements are neglected in the initial stages.Basically, the feedbacks involved in the waterfall model are not cost-effective. A process is desired to have the nature of being flexible as well as adaptable, until the system requirements are found to be not changing with increased legibility level.In addition, the system should have the ability to be employed in a closed and non-dynamic operational situation. As an illustration, consider a wide variety of agent applications that concentrate more on open systems with multiple agents. These systems involve the communication between numerous agents and thereby, create a sprouting behaviour and scattered intelligence that can be viewed in a dynamic operational environment.Hence, the disadvantage of waterfall model is that it never expects the modifications in the requirements and also, the software development is devoid of planning. But, these demerits are more essential in several situations nowadays.The Waterfall model offers less efficiency because even the experts in the software development process cannot decide the parts of the model to be more concentrated. Hence, the performance of the software waterfall process has to be enriched to increase the efficiency as well as the benefits of the software process model. Allowing a software project manager to monitor the activities performed at every single stage or phase of the Software Development Life Cycle (SDLC) can accomplish them and offers a small percent of the entire productivity. The increase in performance and productivity along with the reduction in the effort can be attained, if a set of high level activities are tracked. A list of activities that are subjected to elimination, disregard or entrust forms the rest of the percent in the entire productivity level. Hence, the enhancement in the Waterfall process model can be enhanced and made effective through the increased level of concentration on the specified activities.The waterfall model also suffers from few of the familiar shortcomings [1]. The drawback of more concern is that the requirements are deemed to be stable and known prior to the initiation of the project. In real world situations, the requirements undergo modifications and progress. To handle these varying requirements, the organizations utilize a change management scheme at the time they apply the waterfall model to the project. The use of "Big Bang" scheme has been imagined to be a chief drawback. With this scheme, the whole software can be provided at once the process gets completed. This is highly dangerous because the users are left unknown about how the finished product would be. An iterative development model allows these two key drawbacks to be eliminated. Almost all the waterfall process models employ interoperability, reusability, decreased complexity, upgradability, less expensive, efficient time reduction, efficacy, enhancement in quality and reliability. Reuse of the components is possible, if the present models are examined. Re-engineering gets rid of the difficulties associated with a particular field of research and the reusable components are to be stored in the repository. The reusing methodology can also be applied to create status components and a repository is used to save them as well. A new program can be generated, if these reuse components are exploited.The software development processes in recent times are not cost-effective as well as time-effective and in addition, they are at a long distance from feasibility and reliability. Hence, the researcher attempts to optimize the software development process, which has found utmost importance in the development phase. This problem arises due to the following reasons: (i) Huge software price, (ii) Devoid of reliability as well as feasibility, (iii) Time consuming process (iv)No communication with the customer and (v) The complex nature of the present development procedure. The entire development time can be decreased, if the components and some sort of re-use technology are utilized often. The reason behind the reduction in time is that the software exploitation is reduced in imparting the required functionality through controlling the already available software. The use of components and reuse technique can also be helpful in lessening the period of delivery in cases, where the fundamental development process model looks similar to waterfall without iterations. Extensive study is additionally required on the difficulties met, while organizing and observing the project. The software development process model is also constructed with the thought of how the complex natured software development activities can be managed. Hence, it is necessary to design the development procedure in a very simpler manner.Cutting down the complexity would allow the software development process model to undergo enhancements and paves way for research in the future.

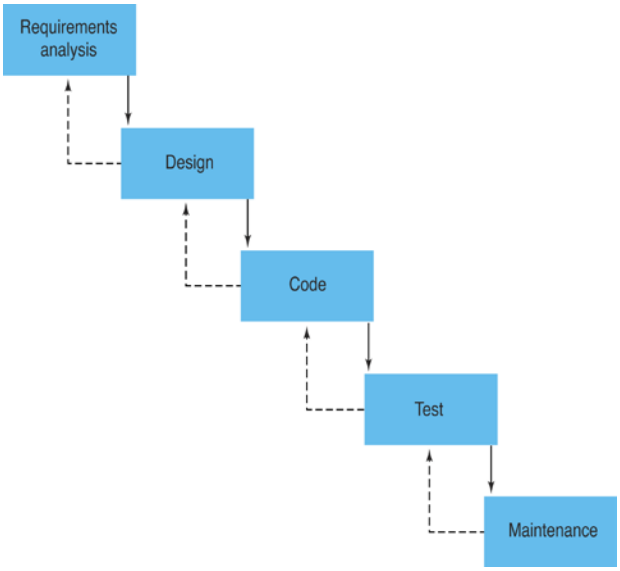

Figure 1. Waterfall Model [5]

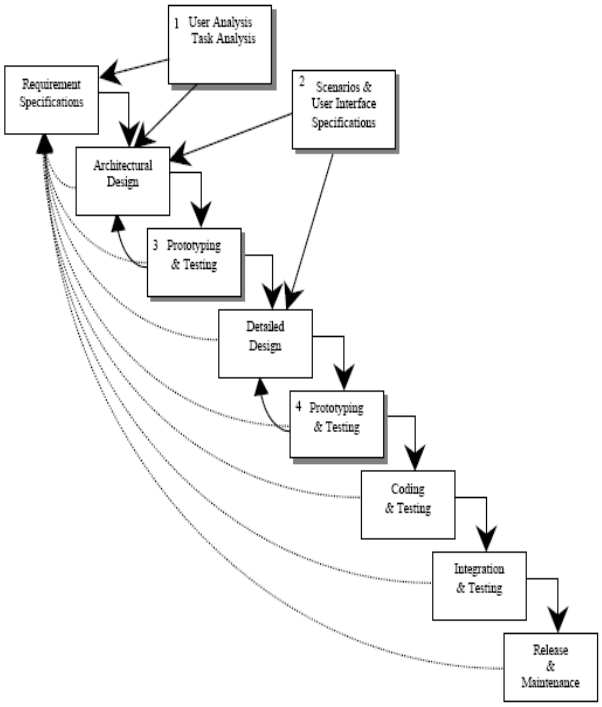

Figure 3.Revised Waterfall Model [7]

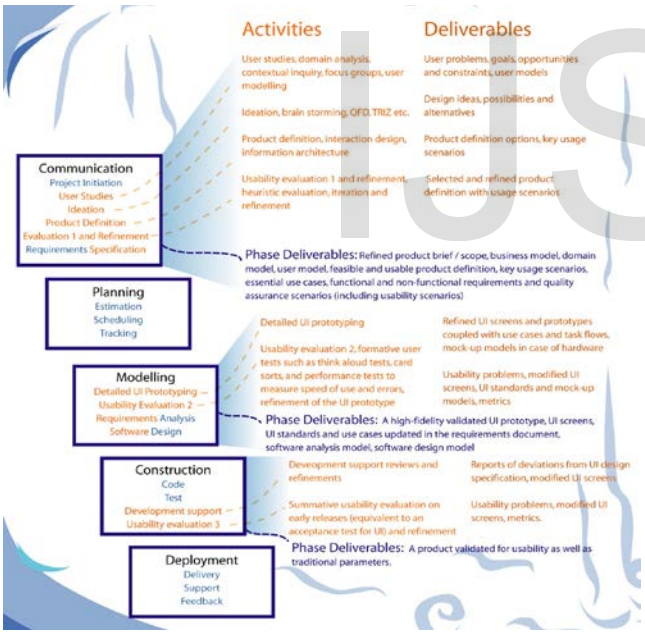

Figure 2. Extended Waterfall Model [8, 10]

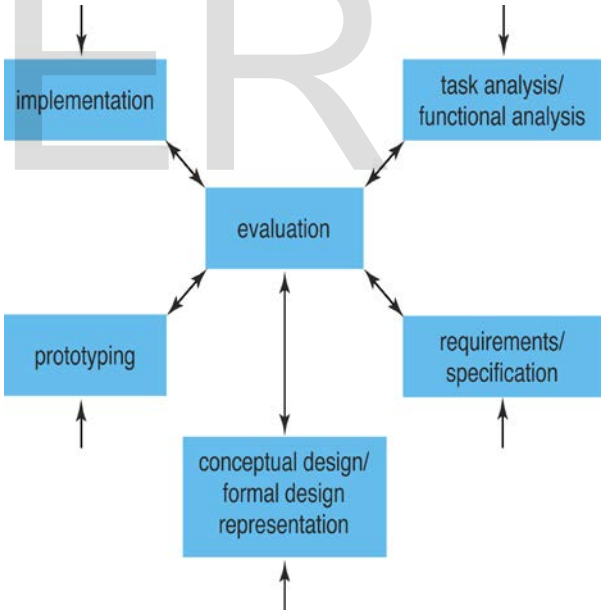

Figure 4.Star Life Cycle [7]

Acknowledgment

The authors wish to thank Dr.Anirudha Joshi, IDC, IIT Mumbai for all the support and guidance.

REFERENCES

[1] Joachim Schramm, Patrick Dohrmann, Andreas Rausch, Thomas Ternité, "Process Model Engineering Lifecycle: Holistic Concept Proposal and Sys-

tematic Literature Review", in proceedings of 40th Euromicro Conference on Software Engineering and Advanced Applications, 2014.

[2] Luca Cernuzzi, Massimo Cossentino, Franco Zambonelli, "Process models for agent-based development", Engineering Applications of Artificial Intelligence, vol. 18, pp. 205–222, 2005.

[3] W. Royce, "Managing the development of large software systems: Concepts and techniques", in: Western Electric show and Convention Technical Papers, 1970.

[4] Laplante Phillip A, Neill Colin J., "The demise of the waterfall model is imminent and other urban myths", Queue ACM Press, vol. 1, no. 10, 2004.

[5] Pressman, R., "Software Engineering – a Practitioner's Approach", 6th ed. McGraw-Hill, 2005.

[6] John Hutchinson,GeraldKotonya, Ian Sommerville, Stephen Hall, "A service model for component-based development", In: Proc of 30th EUROMICRO Conf, 2004.

[7] Costabile, M.F., "Usability in the software life cycle". In: Chang, S.K. (Ed.), Handbook of Software Engineering and Knowledge Engineering, vol. 1. pp.179-192, 2001.

[8] Joshi, A., Sarda, N.L., "HCI and SE: towards a 'truly' unified waterfall process". In: HCI International '07, 2007.

[9] Anirudha Joshi, N.L. Sarda, Sanjay Tripathi, "Measuring effectiveness of HCI integration in software development processes", The Journal of Systems and Software, vol. 83, pp. 2045–2058, 2010.

[10] Anirudha Joshi,"Integration of Human-Computer Interaction Activities in Software Engineering for Usability Goals Achievement", Thesis Submitted for the Degree of Doctor of Philosophy,IDC, IIT Mumbai, 2011

[11] http://www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development/

[12] http://shapingsoftware.com/2008/06/22/constructive-criticism-of-the-waterfall-model/

[13] Seffah A. et al,"Human-Centered Software Engineering-Integrating Usability in the Software Development Lifecycle", Human–Computer Interaction Series, 2005, Volume 8, I, 3-14, DOI: 10.1007/1-4020-4113-6_1

[14] Ahmed Seffah, Eduard Metzker, "The Obstacles and Myths of Usability and Software Engineering", COMMUNICATIONS OF THE ACM December 2004/Vol. 47, No. 12.

[15] Hix and Hartson,"Human-computer interface development: concepts and systems for its management",ACM Computing Surveys(CSUR), Volume 21 issue 1, March 1989

[16] Hix, D. &Hartson, H. R.," Developing user interfaces: Ensuring usability through product and process.", New York, NY: John Wiley & Sons (1993)

[17] www.Volere.co.uk.

[18] Carroll, J. M. HCI Models, "Theories and Frameworks: Toward a Multidisciplinary Science", San Francisco: Morgan Kaufmann publishers. ISBN: 1-55860-808-7. (Ed.) (2003).

[19] Carroll, John M. Human Computer Interaction - brief intro. In: Soegaard, Mads and Dam, RikkeFriis (Eds.). "The Encyclopedia of Human-Computer Interaction, 2nd Ed.". Aarhus, Denmark: The Interaction Design Foundation, (2013). Available online at http://www.interactiondesign.org/encyclopedia/human_computer_interaction_hci.html

[20] Hix and Hartson,"Human-computer interface development: concepts and systems for its management",ACM Computing Surveys(CSUR), Volume 21 issue 1, March 1989

[21] Hix, D. &Hartson, H. R.," Developing user interfaces: Ensuring usability through product and process.", New York, NY: John Wiley & Sons (1993)