

Critical Review of Extended Agile Process Model

Rajesh H. Kulkarni, P.Padmanabham, Vishal Nayakwadi

Abstract—Concrete efforts to integrate Software Engineering and Human Computer Interaction exist in the form of models by many researchers. Many efforts in this regard lead to various proposals of smooth integration of SE (software engineering) processes with HCI (human computer interaction) for product development. One such effort is extended Agile process model. This paper presents a critical review of extended Agile Process model. It also suggests means to bring nearer the two diverse communities of SE and HCI. The objective of closing the gap between SE and HCI is to increase product usability.

Index Terms— Agile process model, Human Computer Interaction, Software Engineering, Usability.

1 INTRODUCTION

The Agile process model (ASP) [1, 2, 3, 4] never indicates the mere development of application in a faster sense. But, it pays utmost attention on the speed as well as the flexibility of adaptation that is required, when the product, the process or the environmental settings undergo a transformation. Hence, ASP does not refer to only one approach. It encompasses the entire number of approaches, which relies on the notion behind coherency. When ASP is compared against the traditional models, it is found that the core idea of ASP exhibit variation from that of the traditional models.

The ASP design highly concentrates on specific attributes of software process and they are the process involving concurrency as well as asynchronous nature, process for object-oriented development, process leading to distributed development and user-centric process. If the higher level-systems are considered, the activities pertaining to the development phase are found to be naturally asynchronous and concurrent. This implies that the advancements in design show a discrepancy between sub-systems or their portions. In addition, the conventional linear synchronous process does not comply with the features of the software processes. As a process supporting Object-Oriented Development, the ASP reveals two frequent attributes in the process models' structure, namely, the iterative micro process and the incremental macro process. The distributed development, followed by the globalization of the software development organization is owed to the boundless economy.

From the period of mid 80's, the geographical distribution of the labour force engaged in development has increased in leaps and bounds that it has exceeded sixty percent. ASP can be deemed as user-centric because the users in huge organizations serve as the contributors to each and every single team and are not secluded.

The process model of ASP system can be thought as an integrated system, which allows the ASP model to be handled. The ASP system's structure and its dynamic nature are more complicated and hence, the incorporation of the software process model is extremely desired in the ASP system. The structure associated with the process involved in the ASP system is mainly concerned with process management, which includes tasks like, defining and enhancing the development processes and having a control over the process execution. An increasingly appropriate representation of the ASP process is required for achieving the ASP management. When the process dynamics of the ASP System is taken into account, the ASP system demands two levels of management structure to carry out the execution procedure. They are the simultaneous allotment of workload to numerous distributed teams for accomplishing several enhancements and the management of the workloads across more number of releases.

The implementation and the functioning of the ASP 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.

- Rajesh H. Kulkarni is currently pursuing Phd degree program in computer engineering in J.N.T University, Huderabad and working in BSCOER Pune, India. E-mail: rkpv2002@gmail.com
- P.Padmanabham is currently Director Academics in BIETin J.N.T. University Hyderabad, India., E-mail: padmanabham46@gmail.com
- Vishal Nayakwadi is working in BSCOER, Pune, India E-mail: nayakwadi89@gmail.com

2 BACKGROUND

The agile software development approaches came into existence in 1990s and it was developed with an intention of overcoming the issues that the conventional software development methods has posed like, inflexibility and the lack of means to account for the user's varying needs [1]. Conboy has given a formal definition for agility as, "the continual readiness of an ISD (information systems development) method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment" [5]. There are so many methods having relation to agility are available [6]. Scrum and Extreme Programming (XP) are the two agile methods, which have been under utilization to a greater extent. Each one of the agile methodologies will possess a different means of execution and different goals. This can be understood from the following example. XP concentrates more on the activities of the software development team, while Scrum aims to bring about the enhancement in project management through the rapid revealing of the troubles associated with the project [6]. The agile software development outweighs the other conventional schemes due to the fact that the users are allowed to communicate with the developers directly [7]. The reports of Standish Group may be divisive [8], but this extensively cited group has depicted that "software applications developed through the agile process have three times the success rate of the traditional waterfall method and a much lower percentage of time and cost overruns" [9]. The agile practices permit the software development teams and the customers to communicate and work together, so that the project can be able to yield improved results along with the achievement of customer satisfaction. The customer satisfaction can be accomplished because the software developers are accustomed to the altering needs of the customer.

3 LITERATURE REVIEW

In the literature review, numerous methodologies that include various processes concerned with agile process modelling are discussed. These approaches give a major description about the previously available best methods with regard to their features along with the knowledge gained after the application of those approaches in the industrial field. This section summarizes each and every research work in a brief manner.

R.Kulkarni et al [31-34] efforts to integrate SE and HCI lead to design of TEIM model which still requires rigorous validation. In 2002, Abrahamsson et al. have provided a technical report at VTT that deals with the development of agile software [10]. This report encloses the idea behind developing the agile software and the various procedures as well as the functions involved. A total of ten agile development methodologies were compared in accordance with the various phases that they aid and the degree of profi-

ciency that is highly demanded. They have unreliably witnessed that the agile methods have increased effectiveness and the ability to suit well in plenty of circumstances [39-42]. Yet, only a smaller range of empirically validated studies satisfy these features. In 2003, they have offered a comparative study on nine other agile methodologies [14] and have stated that the empirical support is still inadequate for these methodologies.

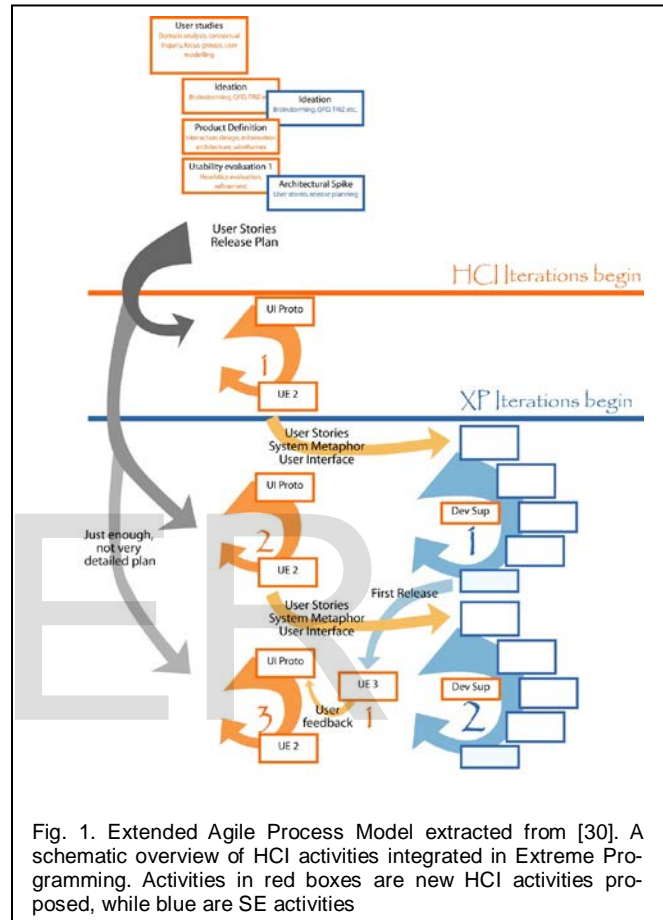


Fig. 1. Extended Agile Process Model extracted from [30]. A schematic overview of HCI activities integrated in Extreme Programming. Activities in red boxes are new HCI activities proposed, while blue are SE activities

In 2004, Cohen et al. have narrated how the agile development has been developed for years before and depicts few sources regarding the other domains. In addition, it describes about the relationships that exist among the agile development and the Capability Maturity Model (CMM) [15]. They have also stated the novelty of the proposed method in relation to the chief agile methods and their features. The practice conditions were also depicted and they were derived from a conversation of 18 experts over online. These experts were busily engaged in making the reports of diverse agile development approaches. Additionally, they have conversed about the various subjects regarding the agile development's preface as well as the project management. Moreover, they have dealt with the testing and review of agile development using seven case studies. The authors imagine that the agile methods would not prohibit the conventional ones, but merging of the ag-

ile methods would take place very soon in a similar way as the object-oriented approaches were merged. Further, they think that a symbiotic link would exist between the traditional and the conventional approaches, wherein, the choice of the process is decided using the count of people functioning on a project, the field of application, importance and novelty. Erickson et al. [16], in 2005, have expressed their research views pertaining to XP, agile software development and agile modelling. The achievement of XP was proved through the investigation of case studies and self-knowledge. Pair programming in XP was examined with a good number of research works and iterative development was also reviewed. Erickson et al. have suggested that it is necessary to make a study in detail about the main practices involved in XP, so that it would be easier to locate the functioning practices from the rest [46, 47]. In addition, they address the issues associated with the agile software development approaches using standards like, ISO and impose further study in this domain because the investigations made on agile modelling is far lower than the researches carried out on XP.

The XP and the Scrum are the two widely utilized models of agile process. Dyba and Dingsoyr [19] have classified the examinations of the models in agile process into four groups of subject matters. They are the introduction and adoption, insight of agile methods, human and social factors and comparative analysis. The authors have also stated that the agile development and an alternative were compared. The authors reported that some form of comparison of agile development was done against an alternative. Khramov in [20] have performed an investigation on the situations, where the XP practices may present better quality codes, but fail to yield better production. Benefield [21] have shown how Scrum influences the Yahoo! development teams. Ji and Sedano [22] have used a number of work products to make a comparison of the effects produced after changing from traditional development to agile development. In [23], a group of methods for co-located teams of varying sizes and criticality through Clear, Yellow, Orange, Red, Blue was used for representation purposes. The largely agile method, known as Crystal Clear, show interest on communication in small teams that are involved in software development and it is of less significance. Clear development can be accomplished using seven traits, namely, requisites for the technical environment, regular delivery, osmotic communication, personal protection, reflective enhancement, concentration and trouble-free access to expert users.

The development of A-Pro-PD [28] was carried out as a portion of Pro-PD and it aims to define a nonspecific agile process model for PD. This research has its introductory phase as a broader literature review and the chief objective of this research was to locate the basic practices involved in the agile as well as the PD methodologies. The results that were yielded at the initial stages were evaluated again with

a chain of iterative workshops for about four months. At these workshops, the feedbacks were gathered along with the proofs from the persons skilled in SPL and agile methodologies. This iterative development phase, which consumes a period of about four months have caused the version one of Pro-PD to be developed. The projects were split into three phases in [24] and they are, pre-project, project life-cycle, and post project. The inspiration for DSDM has emerged from nine principles and they are users' participation, strengthening the project team further, delivery in a regular fashion, improved concentration on the present day business requirements, development in terms of iterations and increments, means for reversing the alterations, setting up a larger scope prior to the initiation of project, conducting tests in the entire life span and communicating with more efficacy. A combination of model-driven and agile development was dealt in [35] and it gives more importance to the initial project model, splitting of the work in terms of features and offering an iterative design for each and every feature. All the iterations associated with each and every feature involves two phases, namely, the design and the development.

Martin McHugh [25] has introduced a hybrid SDLC, named as the AVModel. This model was a blending of the agile as well as the plan driven development practices and can be applied during the development of regulatory compliant software. The issues (for instance, the rigidity) that the medical device software development organizations were facing due to the plan driven SDLC have caused the emergence of this hybrid SDLC. Though the advancements in this model are continuous, it is still advantageous to the organizations that presently develop medical device software and it also has the capability to include appropriate practices additionally. Project management was concentrated more in [27], where planning in prior was little complex using techniques that tend to achieve empirical process control because the feedback loops play a vital role in those mechanisms. A self-organizing team is responsible for developing the software in incremental steps, known as "sprints", and the entire development process begins with planning and concludes with a review. The features, which need to be implemented in the system, undergo registration in a backlog. It is the product owner, who determines the backlog items to be created in the subsequent sprint. The coordination in the work will be accomplished through a stand-up meeting that is held every day. The scrum master, who is one among the team members, takes the responsibility to provide the solutions to the issues that cause the team from not working properly. The Toyota production system was developed in terms of software through the utilization of the rules from lean production [26]. The various principles adopted here were seven in number and they are get rid of the waste, augment the learning tendency, make a decision in a delayed fashion, deliver in a rapid sense, make the team powerful, empower integrity and view the entire thing.

Soren Debois [29] have addressed a current industrial project of Exformatics A/S. Here, the declarative DCR Graphs notation was utilized for modelling and to carry out the implementation of the funded application process of a Danish foundation, in accordance with the agile process. They have given a discussion on the various merits and demerits the process has rendered after its execution, which includes both modelling as well as implementation. The troublesome issues were revealed to the DCR Graphics notation at last, the objective of which is to account for the issues specified in the case study through the promoting of the declarative agile method. Excellent development practices were listed in [28] and there were twelve practices in all. Those twelve practices were the planning game, small releases, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-h week, on-site customers and coding standards. The modified version of "XP2" encloses the following "primary practices": sit together, whole team, informative workspace, energized work, pair programming, stories, weekly cycle, quarterly cycle, slack, 10-minute build, continuous integration, test-first programming and incremental design. In addition, there were totally eleven "corollary practices".

Xiaodan Yu and Stacie Petter [5] have made a study on the shared mental models theory. This study was carried out mainly to investigate the various practices involved in the agile software methodologies and to know the way the software developers operate as a team in an effective manner for accomplishing the tasks. The theoretical background of certain agile practices was examined via the lens of shared mental models theory. With the utilization of the shared mental models theory, a thorough analysis was made on three of the agile practices from Xtreme Programming and Scrum such as, on-site customer, system metaphor and stand-up meeting. This analysis makes clear of the way the agile practices help in enhancing the collaboration, when the software development process was carried out. The outcomes have revealed the method the agile practices have used to support the shared understanding as well as the improved collaboration among the team of software developers. This theoretical examination has expressed the significance of agile practices, when the shared mental models or shared understanding was created within the software developers and the customers taking part in software development teams. Few of the agile practices were found to be beneficial in generating a shared understanding with respect to the tasks to be finished. But, the rest of the agile practices have produced shared mental models in relation to the processes as well as the communications within the team. The necessary results could be extracted from the agile software development approaches, if the agile practices are checked now and then to ensure that they improve the shared understanding of the team. Three agile practices were employed to make sure about the enhancement of collaboration in the working environment of software developers and the associated customers.

4 CRITICAL REVIEW OF EXTENDED AGILE PROCESS MODEL

A.Joshi et al in [30] have proposed a modified Agile Process Model with an intention to reduce gaps between software engineering and HCI process. The knowledge about the extended agile process model can be gained with the splitting of the processes into four portions, namely, the pre-iterations phase, the first HCI iteration, the second HCI iteration and the iterations progressing after it. . The pre-iterations phase involves the execution of the HCI activities of user studies, usability evaluation, ideation and product definition. The First HCI Iteration Agile process models never describe the requisites at once. Agile processes allow the development to be distributed in smaller iterations and the bits of working code resulting from the first iteration are delivered. This demands the delivery of the user interfaces corresponding to those bits delivered from the first iteration in portions in the first iteration itself. When the first iteration of XP begins during the Second HCI Iteration, the HCI team is made to operate on the UI design of the user, its prototyping and evaluation that are preset to be performed in the second iteration. But, when the first iteration of XP gets finished during the second iteration, plenty of UI designs would be created. This methodology is ready to incorporate diversity in the process models, so that the requirements of a wide range of products, the capability of the team and the contexts regarding the project can be met. Improved integration is achievable, if the HCI deliverables exist prior to the situation, where the SE process model demands the associated decisions. The data from eleven industrial projects that have employed agile processes have been examined to identify the influence of integrating HCI activities with the agile process models, which help in attaining the usability objectives.

If the extended agile process model is considered, the means to incorporate the dynamic generation of sub-processes within the HCI model is recommended. The inclusion of the sub-processes to the model allows the processes to be formally verified and the model changes to Turing-expressive. The suitable design tools enable the modelling notation to be visualised in a graphical manner. With this type of visualization, the models become increasingly lucid and cause the business users to robustly adapt the processes on their own at the end. Setting up ownership and managing the associated processes leads to business development without making alterations in the IT systems that aids the processes. Incorrect course of business processes can be eliminated, if the control as well as the monitor phases is considered at each and every single stage of the process. The necessity of IT persons and the maintenance involving higher cost as well as larger duration have been the reason for why the HCI actions with the agile process models assist the needs.

Specific changes required in extended agile process model are in Pre-Iterations Phase, user studies, ideation, product definition and usability evaluation are performed in the extended agile process

model. Here, intelligent decision can be an additional module which can be included to automatically derive the decision rule about the current trend using advanced intelligent tools.

In the first The First HCI Iteration, piece of working code is delivered as per the extended agile process model. The distributed design of the working process model is a major challenge posed by the extended agile process model. Here, objective evaluation is an additional component which can be included in the first HCI iteration to evaluate the first output about the reach ability with the overall objective in the whole extended agile process model.

In the second HCI Iteration, HCI team members are continuously participated in the development team meetings, review ongoing work and provide clarifications or changes as required to quickly integrate the UI design deviation. The metric-based reach ability measures can be included in this iteration to track the overall progress of current iteration within the software development cycle.

In the Third HCI Iteration, time and resource management are important parameters to be considered as two teams should manage the three versions at a time and also, the current development progress. In this iterative phase model, dynamic and systematic process handling should be effectively included as sub-processes to model the iterative process flow. Also, evaluation of each and every iteration should be visualized to the team in visualized way to further improve the development cycle.

5 RESEARCH GAPS AND FUTURE DIRECTION

The agile software development may be more familiar, but the organizations face severe difficulties in employing them. If these agile practices are ignored or their implementation is not appropriate, the entire project duration will fail to develop or continue its collaboration. Though countless agile software development and practices are available, it is better for an organization to use a few. The agile software development can be successfully implemented, if the desirable quality of each and every software development practice is perceived well. But, the empirical researches have shown that there are also problems in recognizing the advantages of agile software development. The subject norms as well as the training are mainly responsible for affecting the agile software development, when it is being adopted. Few researchers have investigated the agile software development and have come to a conclusion that just a part of the organizations using agile methods was able to satisfy the facts defining the agile software development. If the agile practices are applied with the theory, the software development team and the customers can have improved collaboration to each other.

More number of enhancement activities is further demanded to achieve the technical infrastructure, which helps the processes to a larger extent. The enhancement actions should be done in such a way that the processes like documentation, requirement change

management, project management, configuration management, requirement elicitation and process establishment work in coordination with the currently available techniques. Making use of these techniques can lead to additional researches in the fields, where the individuals are allowed to communicate with each other and also with the processes or tools. Moreover, the researches on the control of working software on the comprehensive documentation, the group effort of the customers on contract negotiation and ensuring changes at whatever time the plan requires. Plenty of inferences can be made regarding the research from this systematic review. This review states that a number of empirical studies on agile development approaches is highly required to perform a research work. For the past few years, agile development is of utmost interest in the software industry. This is because, the researchers are attracted towards the way the agile development has been achieved as well as the influence of it, when utilized. The review also expresses that further studies have to be made on the circumstances, where the practitioners can apply the agile development. Action research allows the collaboration to be built between the researchers and the industry. Hence, action research can be more appropriate to be applied in the domain of agile software development.

6 CONCLUSION

In this paper a critical review of extended agile process model is done. We identified the need for dynamic generation of sub-processes within HCI process. Intelligent decision module is recommended in the pre-iteration phase. Objective evaluation in the first HCI iteration considering distributed environment is recommended. In the second HCI iteration metric-based reachability measures are suggested. Visualization in every iteration is recommended for overall development of lifecycle.

ACKNOWLEDGMENT

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

REFERENCES

- [1] MiXio Aoyama, "Agile Software Process Model", in Proceedings of Twenty-First Annual International Conference on Computer Software and Applications, pp. 454 - 459, 1997.
- [2] R. McCauley, "Agile development methods poised to upset status quo", SIGCSE Bull, vol. 33, pp. 14-15, 2001.
- [3] O'Leary P, Ali Babar M, Thiel S, Richardson I, "Product derivation process and agile approaches: Exploring the integration potential", in Proceedings of Second IFIP Central and East European Conference on Software Engineering Techniques. Wydawnictwo NAKOM: Poznan, Poland, 2007.
- [4] Xiaodan Yu, Stacie Petter, "Understanding agile software development practices using shared mental models theory", Information and Software Technology, Vol. 56, no. 8, pp. 911-921, August 2014.
- [5] K. Conboy, "Agility from first principles: reconstructing the concept of agility in information systems development", Information Systems Research, vol. 20, pp. 329-354, 2009.
- [6] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile Software Development Methods: Review and Analysis, VTT Publications, 2002.
- [7] J. Highsmith, A. Cockburn, "Agile software development: the business of innovation", Computer, vol. 34, pp. 120-122, 2001.

- [8] R.L. Glass, "The Standish report: does it really describe a software crisis?" *Communication ACM*, vol. 49, pp. 15-16, 2006.
- [9] S. Group, 2011 CHAOS Report, 2011.
- [10] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile software development methods: review and analysis, VTT Technical report, 2002.
- [11] D. Cohen, M. Lindvall, P. Costa, An introduction to agile methods, in: M.V. Zelkowitz (Ed.), *Advances in Computers, Advances in Software Engineering*, vol. 62, Elsevier, Amsterdam, 2004.
- [12] J. Erickson, K. Lyytinen, K. Siau, Agile Modeling, "Agile software development, and extreme programming: the state of research", *Journal of Database Management*, vol. 16, no. 4, pp. 88-100, 2005.
- [13] P. Krutchen, *The Rational Unified Process: An Introduction*, third ed., Addison-Wesley, Boston, 2003.
- [14] P. Abrahamsson, J. Warsta, M.T. Siponen, J. Ronkainen, New directions on agile methods: a comparative analysis, in: *Proceedings of the 25th International Conference on Software Engineering (ICSE'03)*, IEEE Press, 2003.
- [15] M.C. Paulk, C.V. Weber, B. Curtis, M.B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Boston, 1995, ISBN 0-201-54664-7.
- [16] J. Erickson, K. Lyytinen, K. Siau, Agile Modeling, "Agile software development, and extreme programming: the state of research", *Journal of Database Management*, vol. 16, no. 4, pp. 88-100, 2005.
- [17] T. Dyba, E. Arisholm, D. Sjøberg, J. Hannay, F. Shull, "Are two heads better than one? On the effectiveness of pair-programming", *IEEE Software*, vol. 24, no. 6, pp.10-13, 2007.
- [18] H. Erdogmus, M. Morisio, M. Torchiano, "On the effectiveness of the test-first approach to programming", *IEEE Transactions on Software Engineering*, vol. 31, no. 3, pp. 226-237, 2005.
- [19] T. Dyba, T. Dingsoyr, "Empirical studies of agile software development: a systematic review", *Information Software Technology*, vol. 50, pp. 833-859, 2008.
- [20] Y. Khramov, "The cost of code quality:", in *Proceedings of the AGILE Conference*, pp. 119-125, 2006.
- [21] G. Benefield, "Rolling out Agile in a large enterprise", in: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, 2008.
- [22] F. Ji, T. Sedano, "Comparing extreme programming and Waterfall project results, Software Engineering Education and Training (CSEE&T)", in *Proceedings of 24th IEEE-CS Conference*, pp. 482-486, May 2011.
- [23] A. Cockburn, *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley, 2004, ISBN 0-201-69947-8.
- [24] J. Stapleton, *DSDM: Business Focused Development*, second ed., Pearson Education, 2003, ISBN 978-0321112248.
- [25] Martin McHugh, Oisín Cawley, Fergal McCaffery, Ita Richardson, and Xiaofeng Wang, "An Agile V-Model for Medical Device Software Development to Overcome the Challenges with PlanDriven Software Development Lifecycles", in *proceedings of 5th International Workshop on In Software Engineering in Health Care (SEHC)*, Pp. 12-19, 2013.
- [26] M. Poppendieck, T. Poppendieck, *Lean Software Development - An Agile Toolkit for Software Development Managers*, Addison-Wesley, Boston, 2003, ISBN 0-321-15078-3.
- [27] K. Schwaber, M. Beedle, *Agile Software Development with Scrum*, Prentice Hall, Upper Saddle River, 2001.
- [28] K. Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2000, ISBN 0-201-61641-6.
- [29] [29] Søren Debois, Thomas Hildebrandt and Morten Marquard, "A Case for Declarative Process Modelling: Agile Development of a Grant Application System", in *proceedings of 18th International Conference Workshops and Demonstrations on Enterprise Distributed Object Computing*, 2014.
- [30] A. Joshi, N.L.Sarda, "Integration of Human-Computer Interaction Activities in Software Engineering for Usability Goals Achievement", Thesis Submitted for the Degree of Doctor of Philosophy, INDIAN INSTITUTE OF TECHNOLOGY BOMBAY, 2011.
- [31] R.Kulkarni, P.Padmanabham, "Validating Utility Of TEIM: A Comparative Analysis", *IJACSA*, Volume 4 No. 1, January 2013, U.S ISSN: 2156-5570(Online), U.S ISSN: 2158-107X (Print), 2011
- [32] R.Kulkarni, P.Padmanabham, "TEIM-The Evolved Integrated Model of SE and HCI", *UNIASCIT*, Vol 2 (3), 2012, 301-304, ISSN 2250-0987
- [33] R.Kulkarni, P.Padmanabham, Varsha Sagare, Vipul Maheshwari, "Usability Evaluation of PS Using SUMI (Software Usability Measurement Inventory)", *IEEE- International Symposium On Knowledge-Intensive Software Engineering (KISE'13)*, 22-25 August 2013 Co-located with ICACCI-2013.
- [34] R.Kulkarni, P.Padmanabham, M.S.Namose, Vipul Maheshwari, "Improving Software Quality Attributes of PS Using Stylecop", *Global Journal of Computer Science and Technology(GJCST)*, Volume 13 issue 8, 2013