

Optimized Quality Model for Agile Development: Extreme Programming (XP) as a Case Scenario

Atika Tabassum

Dept. of Software Engineering
Bahria University Islamabad,
Pakistan

Dr. Shahid Nazir Bhatti

Bahria University Islamabad,
Pakistan

Aneesa Rida Asghar

Bahria University Islamabad,
Pakistan

Iqra Manzoor

Dept. of Software Engineering,
Bahria University Islamabad, Pakistan

Dr. Imtiaz Alam

Dept. of Electrical Engineering
Bahria University Islamabad, Pakistan

Abstract—The attributes of quality are that it is complex taxonomy, it cannot be weighted or measured but can be felt, discussed and judged. Early assessment and verification of functional attributes (requirements) are supported well by renowned standards while the nonfunctional attributes (requirements) are not. Agile software development methodologies are of high repute as the most popular and effective approaches to the development of software systems.

Early requirements verification methodologies in Agile Software Engineering are well focused in this way and hence mainly researched have achieved in functional requirements. For early quality aspects (attributes) in order to bring quality in our design and hence development process, it is very important to consider nonfunctional requirements quality metrics (attributes). A comprehensive work is also being done to propose and validate (using iThink) different quality models which could make sure the quality of agile software products being developed, which will be though available in detail in the literature review (section II). Yet a generic and standard quality metrics model is missing in this for the agile software practices in all, which off course is further needed to make sure that the agile product being developed, will surely accomplish quality characteristics as decided by the stakeholders as well as the mentioned quality standard they are addressing. In this work we have proposed a quality metrics model that fulfills the desired quality attributes exist in ISO/IEC (Quality standards, ISO 9126, ISO 25000) in early requirements, we validated this by performing simulations in iThink technology that also ensures that the quality of item being produced to meet the described criteria.

Keywords—Agile Software Engineering (ASE); Agile Software Development (ASD); Extreme Programming (XP); ISO; ISO 9126; ISO 25000

I. INTRODUCTION

Extreme Programming is one of a software engineering practice, articulated in 90s by Ward Cunningham, Kent Beck, and Ron Jeffries [17], the most important and noticeable among several methodologies. XP is different from traditional methodologies in a way that it emphasizes on adaptation than on prediction. In XP programming it is believed that it is more realistic to adapt different changes that appear during the whole software development process rather than specifying all the requirements at the beginning. XP provides a way for

improvement and new style for development. XP aims at lowering the cost of change. The process of Extreme Programming starts with planning and then there are four steps that are followed in all iterations: designing, coding, testing, and listening. Though in the software development process, requirement management and requirement engineering practices are also very important & very critical [18].

The product quality that is being established by adopting the extreme programming methodology is also very important. Different quality models have been proposed that try to cover all the important and critical quality attributes [9] [4]. Here we have proposed a quality model specifically by considering the activities of extreme programming. The details about mentioned quality attributes, model and need for the model in this is highlighted in detail in the literature review section (section II), some limitations are given in section III, further the details about the survey that has been conducted to calculate the effect of different quality attributes are given in section IV and then a model is proposed in section V.

II. LITERATURE REVIEW

A. McCall Quality Model

This very first quality model was offered by Jim McCall [12], the quality characteristics of this model were categorized into three major groups. These three groups contain 11 quality attributes. First is product revision that consists of flexibility, maintainability and testability. Second is product operation which is dependent on honesty, reliability, precision, effectiveness and usability. And third is product conversion that contains the attributes reusability, transferability and interoperability. The sole aim of this model was to minimize the gap between the developers and the end users by emphasizing on the dimensions of quality. This model is suggested for generic systems, and hence the attributes that are specific for different domains are not clearly spoken in the space of the mentioned model [9].

B. Boehm Model

Boehm's quality model portrays a numerous leveled structure of attributes, each of which adds to the total quality.

Boehm's quality model [3] is fundamentally the expansion of McCall Quality model.

Utility characterizes how effortlessly, dependably and effectively programming item can be utilized, practicality characterizes how effectively modifiable and retestable the product item is and portability characterizes how the product item can be utilized after change has happened as a part of a domain [11]. Boehm distinguished seven quality components: Portability, Reliability, Efficiency, Usability, Testability, Understandability, and Flexibility [9].

C. FURPS Model

FURPS model [4], classifies attributes into two unique requirements, for example, Functional Requirements (F) which is characterized by predictable input and output and Non Functional Requirements in which U represents Usability, R represents Reliability, P represents Performance (incorporates practical prerequisites) and S represents Supportability (incorporates reinforcement, essential of plan, execution, interface) [9]. One drawback of the FURPS model is that it doesn't reflect the portability perspective, which might be a vital basis for application advancement, particularly for programming based frameworks [11].

D. Dromey's Quality Model

Dromey's proposed an operational structure [15] [9] [4] for evaluating Requirement assurance, plan and practice stages. The structure involves three models, i.e. Requirement quality model, Design quality model and Implementation quality model. The top item properties for this model incorporate [11]:

- First one is correctness that checks that no defacto standard is affected and also checks the usefulness of quality characteristics.
- Second is internal measure that how well a module has been situated by future use, with proficiency, practicality, and dependability as quality characteristics.
- Third is contextual that manage the outer effects on the use of a constituent, with quality attributes in dependability, viability, reusability, and compactness.

E. IEEE Quality Model

IEEE Quality Model is mainly standard for software maintenance [6]. This model offers a process for handling and executing software maintenance actions. Standards like quality assurance, confirmation and authentication, software formation controlling in which linked processes are well-defined [9]. This model represents numerous proportions of qualitative features and signifies features that are Reliability, Functionality, Usability, Efficiency and Maintainability [9] [14].

F. ISO 9126-1 Model

ISO 9126 is a universal model for the improvement of programming [11]. ISO 9126 states and figures the product item quality regarding inner and outside programming qualities and their relationship to properties. The ISO 9126-1 quality model [9] [14] is additionally ordered into two groups. Initial one is Quality being used traits and the second that contains internal quality properties and External Quality properties.

Quality being used traits is those properties that can be evaluated just when programming is satisfied and conveyed to the end client. Then again inner quality properties are those that can be measured even without executing the item, though outer quality characteristics are those that can't be measured without executing the item. ISO 9126-1 quality model contains these properties so that the item can meet the guidelines of quality.

G. Ghezzi Model

Ghezzi C. et al. [7], characterizes that center qualities manage the structure of programming which benefits the product engineers to achieve those outer qualities for which programming clients have a ton of concern furthermore conveyed both internal and external characteristics of programming which are Reliability, Maintainability, Reusability Usability, Flexibility, Portability, Accuracy, and Integrity [14].

H. Other quality models/frameworks

Richard et al [8] discussed the 24 quality attributes specifically focusing on extreme programming. In XP a customer take part with the development team so specification is not a single document. It consists of user stories, acceptance tests written by customers and unit test written for each module. Author basically applied those 24 quality attributes that were proposed by Davis [9] for a quality SRS, on a specification created with XP. However extreme programming process achieves higher values in nine attributes and drops the values in two. The most positive results were in ambiguity and understandability because in extreme programming customer is present at all times to answer every question [8].

M. Usman et al. [9] proposed a quality metric model for agile development. Eight quality attributes were considered most important for agile development. Those were availability, flexibility, testability, scalability, performance, portability, understandability and usability. And the influence of these quality attributes among individual phases of SDLC was also evaluated. Results emphasized that flexibility is the best quality attribute among all attributes and then portability and understandability. However it was suggested at the end that more quality attributes can also be included in the quality model that are maintainability and modifiability and their influence on software development lifecycle.

Robert et al. [10] presented the influence of agile development on quality inside the administrative, procedural and traditional back ground and provided business standard approvals to reduce such influences. It was suggested that IT organizations should practice quality techniques and quality frameworks like AAIM, AQT and ASSF. The agility characteristics that were described are flexibility, speed, thinness, learning and receptiveness. However this work largely concentrated on the surface of administrative and traditional impacts. More research examination and survey are needed to increase the knowledge base linked to this matter.

Deepshikha et al. [11] proposed a framework that contained some steps for the improvement of different quality models. Here author's basically compared different quality models and found the main alterations between these models. It was found

that in the 17 features only one quality feature is same in all models and that is reliability. Similarly, there are merely three features (i.e. portability, efficiency, usability) which are fitting to four quality models. Two features are mutual only to three quality models and that are functionality and maintainability. Two features fit in to two models and that are testability and reusability. And, nine features are presented in only one quality model. At the end some comments were given to these models. It was said about McCall model that it is a general model based only upon the finding of a person's questionnaire. FURPS was built for a company so it is a special purpose model, however it was suggested that a new model can be built from these quality models [9] [4] [7].

Inderpal et al. [12] compared different characteristics of different quality models and it was found that maintenance cost depends so much upon the quality of a product. Author compared different quality models and it was found that reliability is a common characteristic of all quality models. It was also suggested that different quality features can be executed with relative to cost, schedule and modification. However it was suggested at the end that it's very significant to have very decent description of software requirements to have greatest results. It should be noted that different phases that are involved in a process should be done in proper way to have quality.

Ranbir et al. [15] presented a survey of different quality models [3] [4] [12] [6] and also did their comparative analysis. To do the comparison data was collected from various

organizations through questionnaire and also through different published articles. Interviews were also taken from various students. The questionnaire contained different quality attributes from different quality models and their suitability was measured and analyzed. Analysis was basically done by using different tools that help for analyzing different things and then results were presented in the form of comparison table. However, it was suggested at the end that all these quality models are working well but still there is a need a software quality model that can be applied during whole software development lifecycle.

Sanjay et al. [14] depicted various quality models and their investigative assessment, decided programming capability and its subjective attributes all the more plainly. Diverse programming quality models were suggested for programming applications by different scientists. The ISO 9126-1 model which in actuality joins the consequences of various different models has been thought as the most recognizable model and this has been broadly recognized and acknowledged as an elementary model in range of business and research. This was a comprehensive study to process the diverse components of various programming quality models and figure their near plausibility. Simultaneously, an investigation of various models which have been utilized to ascertain the quality will be inspected suitably.

Table I comprehensively highlight the comparison of different quality models/ frameworks ascribed in detail in literature in section II.

TABLE I. COMPARISON OF DIFFERENT QUALITY FRAMEWORKS [9] [11] [7] [4] [10] [3]

Quality Attributes	McCall	Boehm	FURPS	Ghezzi	IEEE	ISO 9126	Dromey	Robert et al	M. Usman et al	Micheal et al
Correctness	✓									
Efficiency	✓	✓	✓	✓	✓					
Flexibility	✓							✓	✓	
Reliability	✓	✓	✓	✓	✓	✓	✓		✓	
Testability	✓	✓							✓	
Maintainability	✓	✓	✓	✓	✓					
Portability	✓	✓		✓	✓				✓	
Functionality			✓		✓	✓	✓			
Modifiability		✓								
Performance			✓				✓		✓	
Interoperability	✓									
Security	✓									✓
Usability	✓	✓	✓	✓	✓	✓	✓		✓	
Robustness										
Accuracy				✓						
Integrity				✓						
Reusability	✓			✓				✓		
Supportability			✓							
Scalability									✓	
Process Maturity							✓			

III. LIMITATIONS OF DIFFERENT QUALITY MODELS

A few limitations have been observed in various quality models. These are followings:

One of the most significant offerings of McCall model for software quality is describing the association between SQ dimensions. Though, this model did not reflect one of the main features of software quality that is

functionality. By description, functionality is what a product can do for a user. So functionality is key factor to assess the software so that it would meet the expectations of user.

Boehm proposed a quality model based on the needs of users but did not provide any recommendations for evaluating the software quality features in that model. There can be additional research done on the depth of

SQ dimensions. Hence this research displays a need to assess the software quality.

One of the major drawback of FURPS model is that it does not reflect one of the significant quality attribute portability. Portability is the ability of the software to work in dissimilar or diverse situations and environments. And user's environment might keep changing and therefore software also needs to adjust to new computing environment. Especially in agile portability can be a main attribute and for that reason cannot be ignored.

ISO-9126 appears to be more precise, comprehensive and does not fall short as other models do. But, it has not delivered the clarity of how some specific software quality attributes can be measured. This can, however, be the best model in comparison to the other proposed models.

Dromey's model tried to enhance the understanding of the association between features and sub-features. So this model could not emphasis on how to measure the software quality. It has recognized the relationship between quality features and sub-features. This research therefore studies this relationship and evaluates the software quality.

IV. RESEARCH METHODOLOGY

Taking this comprehensive research literature into account and research work done into the quality metrics in agile software engineering (XP etc.) it has been observed that the quality aspect is the most important aspect in every methodology that is used for the development of software products. Further that for bringing improvement in the quality of the product, different models have been suggested and used in the industry. Those models are helpful for achieving high quality software products but we precisely we are unable to identify in the mentioned literature and others about a model that is specific for agile methodologies and especially and precisely about the extreme programming. As we know that extreme programming is different from other previous methodologies [17], there must be a standard quality metric model that fulfills the criteria for the activities and quality attributes (metrics) of agile methods especially extreme programming. Those quality attributes that contain highest influence on the activities of extreme programming must be analyzed and measured.

A survey has been conducted through different software organizations (at Software Technology Park). The methodology used for this scenario to conduct this survey (Qualitative analysis) was that of set of questionnaires and face to face interviews (mostly open-ended). In this almost 20 different software houses/ software companies have been contacted and hence covered via this survey. The frequently asked questions were as follows:

1) Do you think that choosing the most suitable quality model is a real challenge when you are working properly in agile?

2) In order to ensure high quality product do you think the developer must concentrate on the quality of the process?

3) Is there a need of a quality model specifically for extreme programming (agile)?

4) Are there any attributes in different quality models that not need to be addressed in agile development?

Almost all the companies/software houses agreed that selecting the most appropriate model is a challenge because all those models are developed for traditional methodologies. Most of these models appear to be fully adopted in large scaled organizations or enterprises. Maybe these models could accommodate small companies or shorter version of businesses. All of the quality models emphasize too much on documentation which agile proves to be resisting due to development at a very high pace.

Almost 80% of the organizations think that there is a need of a quality model specifically for extreme programming (agile) because all the previously developed models contain so many attributes that are not needed in extreme programming (agile) so we can say that those models are complex for agile development. Secondly most of these models appears to be fully adopted in large scaled organizations or enterprises. Maybe these models could accommodate small companies or shorter version of businesses.

It is known that XP is different from all other methodologies in many ways. It basically involves user stories, customer availability, pair programming, small releases and iterations, continuous integration, unit and integration testing, acceptance testing and customer feedbacks. On the basis of these steps, we can try to define that which quality attributes are most important for XP and we can make a new quality model specifically for the products that are developed through XP approach.

Quality product is always very important and it is understood that in case to accomplish a quality product, quality of the process must be considered. Our proposed model will basically represent all the phases/steps of extreme programming and the quality parameters that are necessary to achieve the quality product.

An evaluation criteria is defined here to measure the influence of different attributes on agile SDLC.

Very Strong Effect	Strong Effect	Average Effect	Low Effect	Very low Effect
10	8	6	4	2

Almost 25 quality attributes in our questionnaire from all the quality models are considered and evaluated against all the phases involved in extreme programming (agile).

On the basis of this survey through questionnaire it has been observed that 7 quality attributes are the most important quality attributes for an extreme programming (agile) product. Those 7 attributes have the highest influence on the quality of agile process as well as on the quality of agile product.

TABLE II. EFFECT OF QUALITY ATTRIBUTES ON EXTERME PROGRAMMING SDLC & XP PRODUCT

Quality Attributes	User stories/	Design/ Release	Implementation/Pair Programming	Unit testing	Integration testing	Small Release	Acceptance testing	Final Product	Overall Influence
--------------------	---------------	-----------------	---------------------------------	--------------	---------------------	---------------	--------------------	---------------	-------------------

	Requirements	planning							
Availability	10	8	8	8	8	10	8	10	70
Efficiency	4	4	10	10	6	6	8	10	58
Usability	2	4	6	8	6	8	8	10	52
Maintainability	8	8	10	6	6	10	8	10	66
Testability	4	6	8	8	10	8	8	8	60
Flexibility	8	8	10	8	6	10	4	10	64
Portability	2	6	8	8	8	8	8	10	58

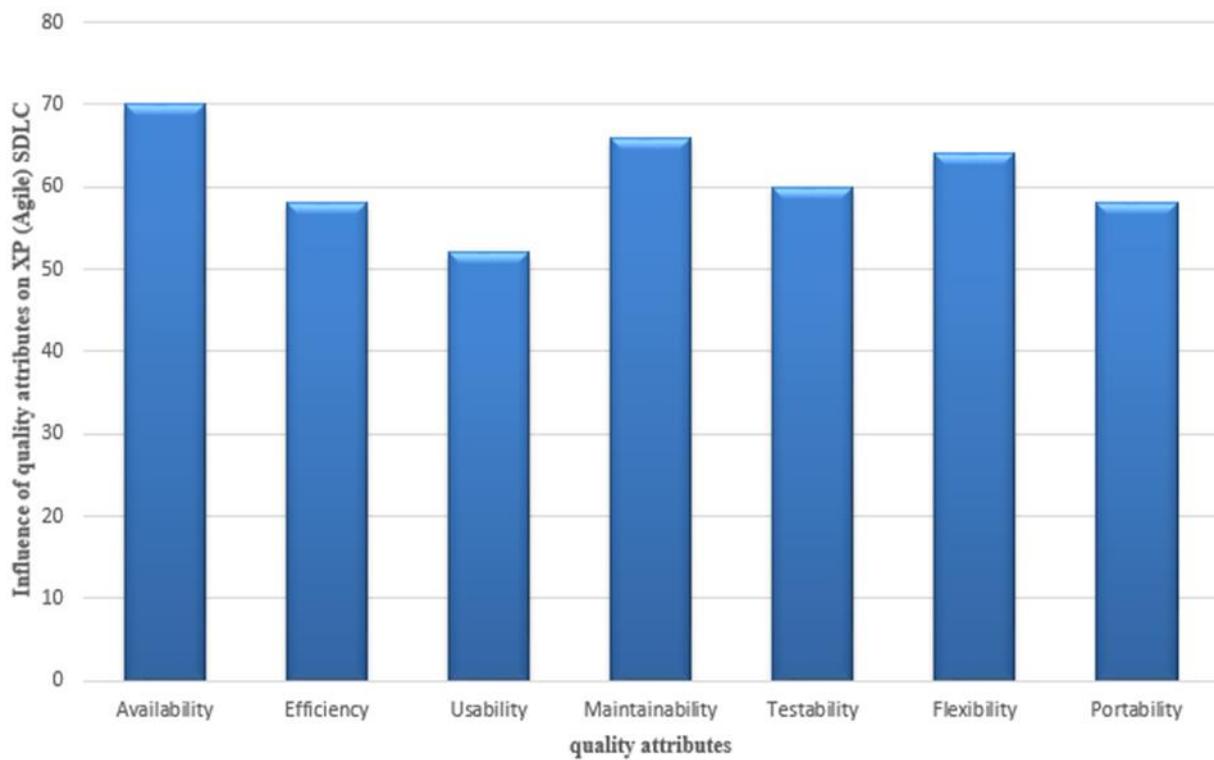


Fig. 1. Effect of quality attributes on extreme programming development process

V. PROPOSED MODEL

On the basis of these results from the survey, we have developed our quality model for Extreme Programming (Agile).

This model contains those attributes/features that have highest influence on the quality of agile SDLC and agile product. It also contains the sub features that are helpful for measuring these attributes.

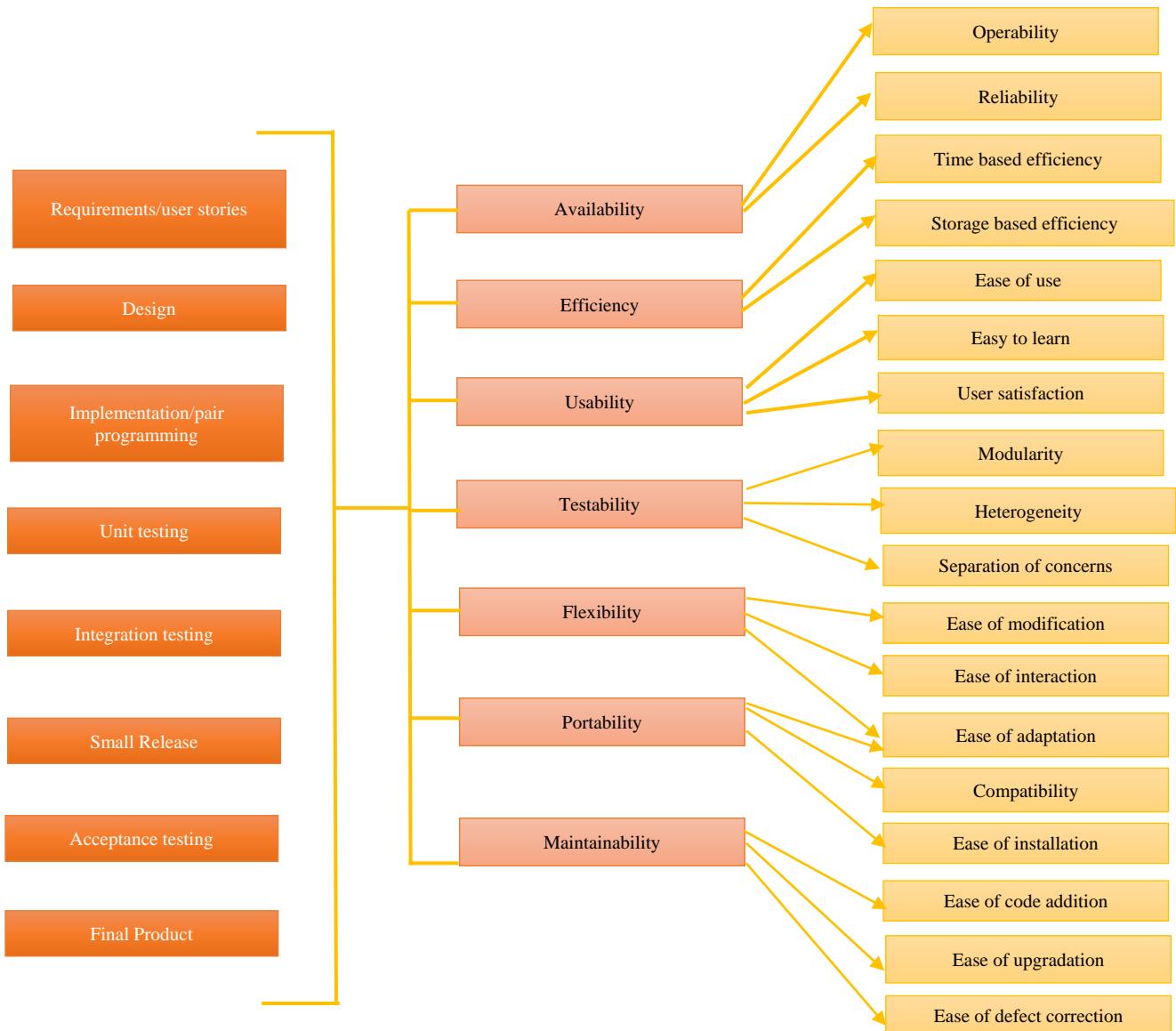


Fig. 2. Quality model for XP process & product

The quality attributes of a proposed quality model (quality model for extreme programming) are defined in a table 3.

And the sub features of proposed quality model (quality model for extreme programming) are defined in table 4.

TABLE III. DEFINITIONS OF QUALITY ATTRIBUTES USED IN PROPOSED MODEL

Availability	It refers to working ability of the application. The degree to which a system can continue to work when a major component or set of components goes down. [11]
Efficiency	A set of attributes that relate to the relationship between the level of performance of the software and the amount of resources used, under stated conditions.
Usability	A set of attributes that relate to the effort needed for use, and on the individual assessment of such use by a stated or implied set of users.
Testability	Attributes of software that relate to the effort needed for validating the modified software.
Flexibility	Flexibility refers to the ability of an application to undergo changes when required without affecting the overall application. [11]
Portability	A set of attributes that relate to the ability of software to be transferred from one environment to another. [11]
Maintainability	A set of attributes that relate to the effort needed to make specified modifications.

TABLE IV. DEFINITIONS OF SUB ATTRIBUTES USED IN PROPOSED MODEL

Operability	Attributes of software that relate to the users' effort for operation and operation control.
Reliability	A set of attributes that relate to the capability of software to maintain its level of performance under stated conditions for a stated period of time. [3]
Time based efficiency	This characteristic indicates the ability to perform a specific task at the correct time, under specified conditions. [3]
Storage based efficiency	It is the ability to store and manage data that consumes the least amount of space with little to no impact on performance; resulting in a lower total operational cost. [3]
User satisfaction	It is the degree to which a system is pleasant to use for the user.
Modularity	It is the degree to which a system's components may be separated and recombined.
Heterogeneity	It is the degree to which a system's components are separated and flexible enough so that they can be tested or used easily.
Separation of concerns	Separation of concerns (SoC) is a design principle for separating a computer program into distinct sections, such that each section addresses a separate concern.
Ease of modification	Corrections, improvements or adaptations of the software to changes in environment and in requirements and functional specifications.
Ease of adaptation	Attributes of software that relate to the opportunity for its adaptation to different specified environments without applying other actions or means than those provided for this purpose for the software considered.
Compatibility	Software that is composed of elements that can easily combine with other elements.

We have tried to validate our model with the help of a dynamic simulation tool (iThink). The graphs generated at the end (figures 4, 5 and 6) are showing that when we increase our inflows that means when we increase rate of availability, rate of efficiency, rate of modifiability, rate of portability then we see that quality of the process increases with time and vice versa.

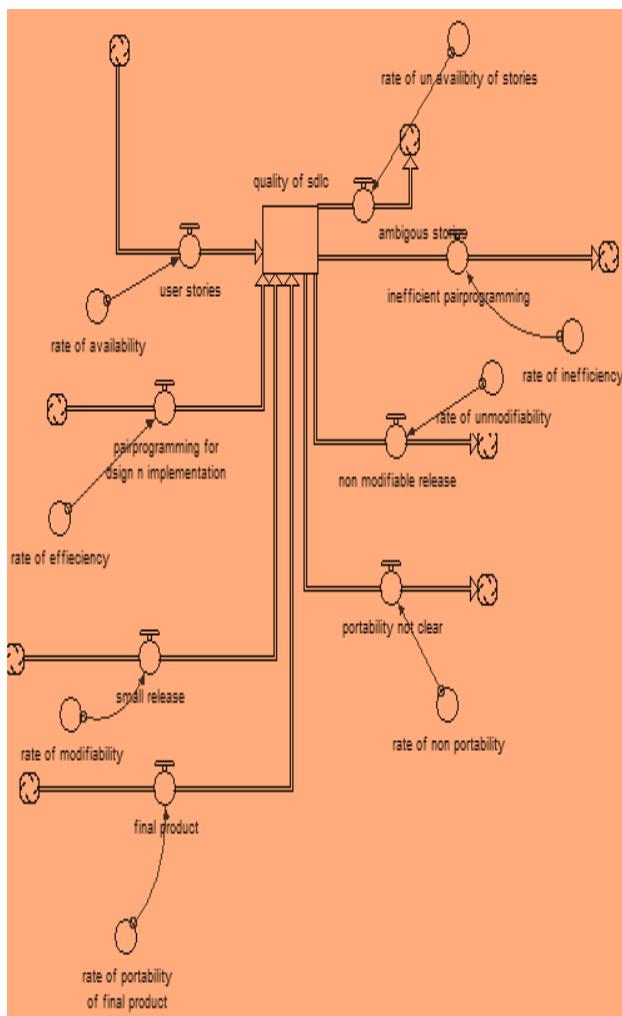


Fig. 3. Simulation validation in iThink

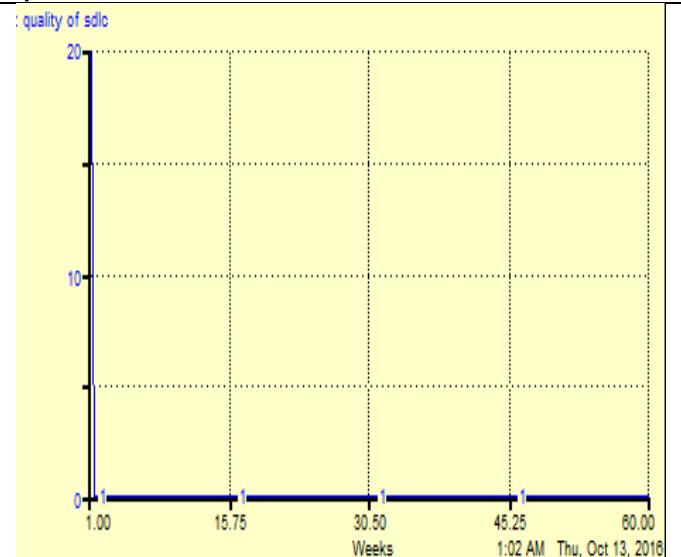


Fig. 4. Increase in Quality at early iterations

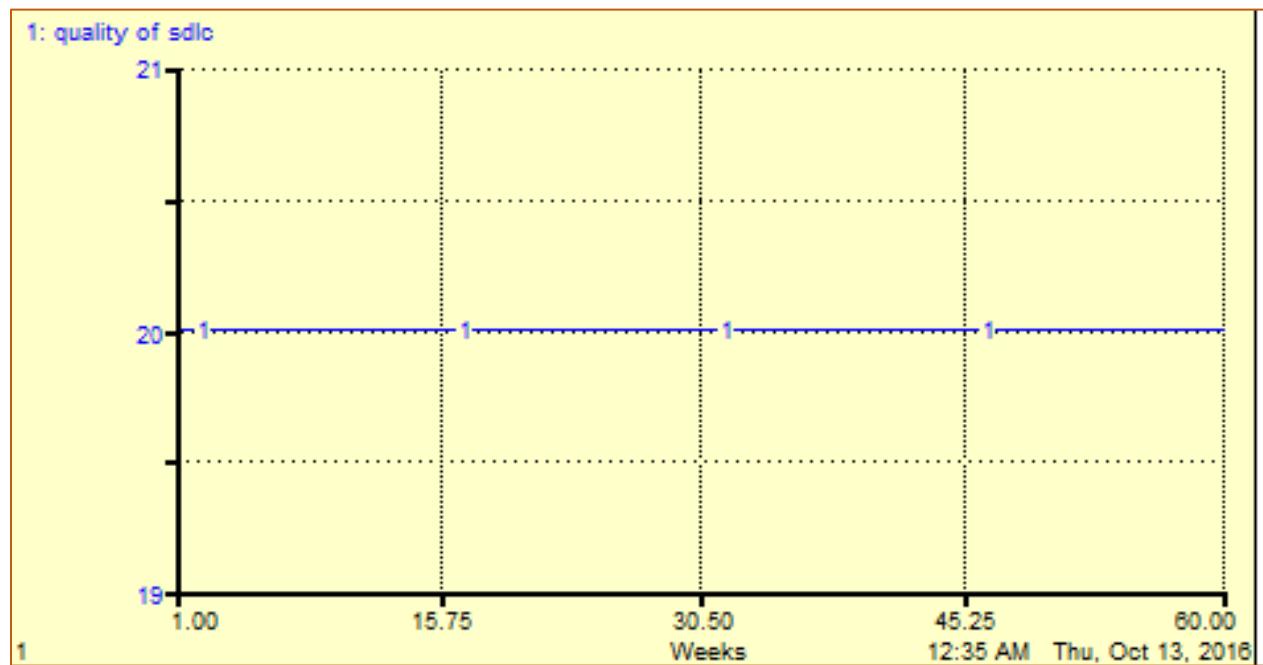


Fig. 5. To Static Quality via Attributes

And we are changing the values of different attributes then graph is changing constantly. In this way we can say that if we emphasize on a quality process and we will maintain quality steps at every development phase then quality of SDLC will continue to increase.

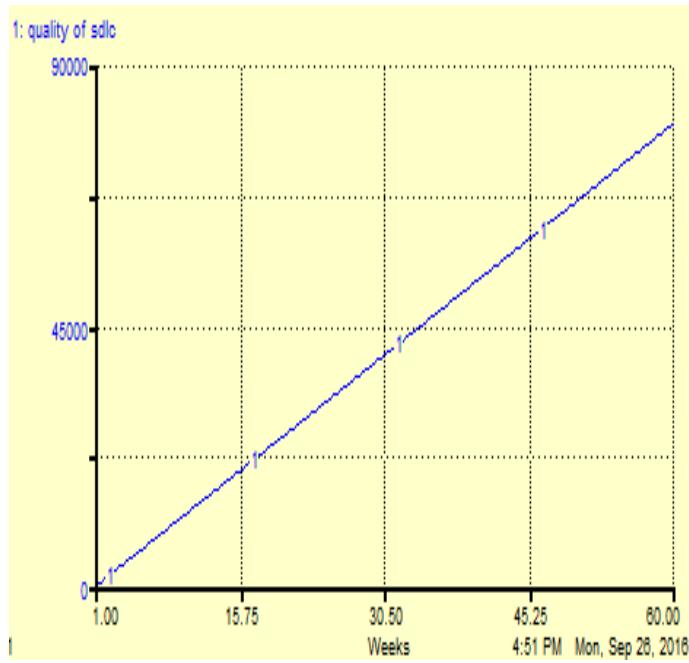


Fig. 6. Development in progress vs quality

VI. CONCLUSION

Software systems have to ensure consistent and bug free execution at a rapid pace every time they are used especially in Agile Development. Improving software quality and performance has become a priority for almost every organization that relies on the software development. Thus the quality issue related to the software's industry becomes more important, apparent and more technical also considering the user's requirements in this aspect. The following work demonstrates the need for and a detailed quality model for an XP process as well as for an XP product. Further this also highlights that if we want a quality product then we must concentrate on the quality of the process first and only by estimating quality attributes of a quality process (ISO 25000) we can achieve a quality product metrics (ISO 9126).

Improving software quality and performance has become a priority for almost every organization that relies on the software development. As software development grows more powerful the users demand are more powerful, sophisticated software. Thus the quality issue related to the software's industry becomes more important, apparent and more technical also considering the user's requirements in this aspect. To value customer trust and overall quality as defined by International Standards (ISO/ IEC 9126 & 25000), the quality metrics attributes must be taken into account in the planning and design of the software.

Further, in Quality estimation (metrics/ attributes) in Agile Development (XP) as we basically involves user stories, customer availability, pair programming, small releases etc., if we specify which quality attributes are most important for XP which was briefly highlighted using the proposed model. The following methodology (framework) proposed essentially represents all the phases/ steps of extreme programming and the quality parameters that are necessary to achieve the quality product in this way. As concluded already we have to subsist with this that in case to accomplish a quality product, quality of the process must be considered a prime initiative.

The following paper presents a comprehensive quality model for agile and specifically for extreme programming. All

those attributes that have highest influence on the quality of agile SDLC and agile product are included and mentioned in this model. The influence and effect of each quality model has also been presented in tabular form and also in the form of graph. It has been observed availability, flexibility and maintainability has the highest effect on the quality of agile SDLC and agile product.

REFERENCES

- [1] Kumar, A., Kumar, R. and Grover, P. S. (2006). A change Impact Assessment in Aspect-Oriented Software Systems, In the proceedings of International Software Engineering Conference Russia, (SECR-2006), Dec, pp. 83-87.
- [2] Capers Jones, (2009). Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies, 1st Edition, McGraw-Hill Education; 1 edition
- [3] Sanjay Kumar Dubey1, Soumi Ghosh2, Prof. (Dr.) Ajay Rana (2012) Comparison of Software Quality Models: An Analytical Approach, International Journal of Emerging Technology and Advanced Engineering.
- [4] Zainab Sultan, Rabia Abbas, Shahid Nazir Bhatti and S. Asim Ali Shah, "Analytical Review on Test Cases Prioritization Techniques: An Empirical Study" International Journal of Advanced Computer Science and Applications(IJACSA), 8(2), 2017.
- [5] Boehm, B.W (1978). Characteristics of software Quality, TRW Series of software Technology, Amsterdam, North Holland.
- [6] Aneesa Rida Asghar, Shahid Nazir Bhatti, Atika Tabassum, "Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development" International Journal of Advanced Computer Science and Applications(IJACSA), 7(12), 2016.
- [7] Scotch Ambler (2002) Agile Modeling: Effective practices for extreme programming and the Unified Process, Wiley; 1 edition
- [8] Shahid Nazir Bhatti, Deducing the complexity to quality of a system using UML. ACM SIGSOFT Software Engineering Notes 34(3): 1-7 (2009)
- [9] Aneesa Rida Asghar, Shahid Nazir Bhatti, Atika Tabassum, "The Impact of Analytical Assessment of Requirements Prioritization Models: An Empirical Study" International Journal of Advanced Computer Science and Applications(IJACSA), 8(2), 2017.
- [10] Jeff Langr (2011) Agile in a Flash: Speed Learning Agile Software Development, Pragmatic Bookshelf; 1 edition
- [11] Ghezzi, Jazayeri, C. M. and Mandrioli, D.(1991) Fundamental of software Engineering, Prentice-Hall, NJ, USA.
- [12] Richard Duncan (2001) The Quality of Requirements in Extreme Programming, The Journal of Defense Software Engineering
- [13] M.Usman Malik, M. Haseeb Nasir, Ali Javed (2014)An Efficient Objective Quality Model for Agile Application Development, International Journal of Computer Applications
- [14] Aneesa Rida Asghar, Shahid Nazir Bhatti, Atika Tabassum, "Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development" International Journal of Advanced Computer Science and Applications(IJACSA), 7(12), 2016.
- [15] Robert Imreh , Mahesh S. Raisinghani (2011) Impact of Agile Software Development on Quality within Information Technology Organizations, Journal of Emerging Trends in Computing and Information Sciences
- [16] Dr. Deepshikha Jamwal (2010) Analysis of Software Quality Models for Organizations, International Journal of Latest Trends in Computing
- [17] Inderpal Singh (2013) Different Software Quality Model, International Journal on Recent and Innovation Trends in Computing and Communication
- [18] Robert K. Wysocki (2009) Effective Project Management: Traditional, Agile, Extreme, Wiley; 5 edition
- [19] Shahid N. Bhatti, Maria Usman, Amr A. Jadi, 2015, Validation to the Requirement Elicitation Framework via Metrics. ACM SIGSOFT Software Engineering Notes 40(5): 17, USA.
- [20] R Geoff Dromey (1995) A model for Software Product Quality, IEEE Transctions on Software Engineering
- [21] <http://testerwiki.blogspot.com/2012/12/quality-assurance-activities-in-sdlc.html>
- [22] Jerzy Nawrocki, Bartosz Walter (2002) Extreme Programming Modified: Embrace Requirements Engineering Practices, Proceedings of the IEEE Joint International Conference on Requirements Engineering.
- [23] Ranbireshwar S. Jamwal, Deepshikha Jamwal & Devanand Padha (2009) Comparative Analysis of Different Software Quality Models, Proceedings of the 3rd National Conference; INDIACom-2009. Computing For Nation Development, February 26 – 27, 2009
- [24] Farrukh Latif Butt, Shahid Nazir Bhatti, Sohail Sarwar, Amr Mohsen Jadi and Abdul Saboor, "Optimized Order of Software Testing Techniques in Agile Process – A Systematic Approach" International Journal of Advanced Computer Science and Applications(ijacsa), 8(1), 2017.
- [25] Alexey G. Malishevsky, Joseph R. Ruthruff, Gregg Rothermel, Sebastian Elbaum "Cost-cognizant Test Case Prioritization" Technical Report TRUNL-CSE-2006-0004, Department of Computer Science and Engineering, University of Nebraska – Lincoln, 2006.