

# Improving Quality of Perception (QoP), Quality of Experience (QoE), and Quality of Service (QoS) in agile development using Cleanroom Software Engineering (CSE)

Sana e Zainab, Munazza Jannisar, Ali Javed

University of Engineering and Technology, Department of Software Engineering, Taxila, 47080, Pakistan  
Email: sana\_zanaib@yahoo.com, munazzajannisar@yahoo.com, ali.javed@uettaxila.edu

**Abstract**—Pioneering ideas from the software engineering discipline have factually affected every sphere of life. Agile software development approach has been promoted since its commencement and stipulates strategies that improve the quality of software product. To consummate fast and reliable development processes, several agile approaches are charted and are quite popular. For quality improvement and to achieve defect free system, the concept of Cleanroom Software Engineering (CSE) is ingrained into agile development life cycle. For embedding users concerns, it is important to distinguish three approaches to quality: Quality of Service (QoS), User-perceived QoP, and Quality of Experience (QoE). QoS is technology centered approach, so by using Incremental Planning of CSE, it shall facilitate the customer's clarification of system requirements and will control the technical complexity. Usage Specification and Usage Modelling will be used during the Certification phase of CSE which will help to achieve QoP and QoE, being user centered approaches. Results collected from Survey conducted, explains above mentioned factors improvement.

**Index Terms**—(QoS) Quality of Service, (QoP) Quality of Perception, (QoE) Quality of Experience, (CSE) Cleanroom Software Engineering, (SDLC) Software Development life Cycle.

## I. INTRODUCTON

Agile loom has gone from being a manifesto to wide industry focus. Most of the product majors are adopting it, which pledge to solve assorted gripe that traditional software development methodology is supposed to have created. The short, time constrained sprints with predefined goals and the overall technique that scrum proclaim helps to achieve intended artifact [16].

A commissioned study conducted by Forrester Consulting on behalf of HP, August 2012 [17]. It began in July 2012 and was completed in August 2012. About 112 IT professionals were surveyed in traditional and emerging industries about their Application development habits and practices, either in Agile or traditional way.

The purpose of this survey was to measure success, and what are the success rates. 24 most successful companies were identified and compared them with the 88 remaining companies. The conclusion of their research was that: Agile teams are 36% more prolific, team morale is 20% more improve, The involvement of business sponsors is 47% more and their results in simple, flexible applications 62% more often than the control group.

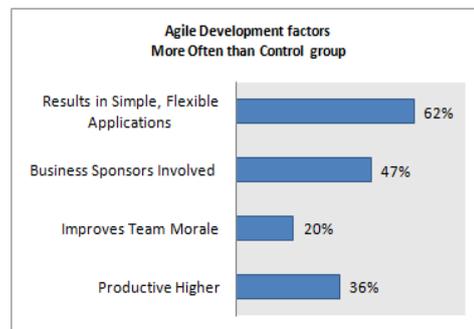


Fig 1 Agile Comparison with Control Groups

The problem encountered when some things were less often did in successful groups rather than control groups. The successful group relied of this theory that it is good to limit the work in progress, its requirements awaits the coding 81% less often than the control group, its coded modules await testing 70% less often, and system testing holds off until all coding is complete 47% less than the control group.

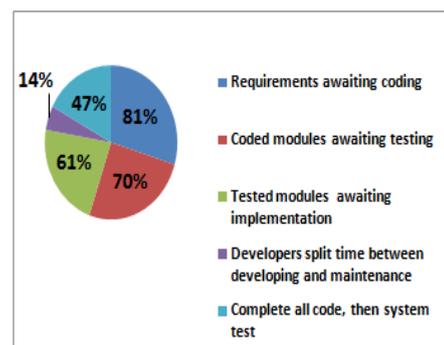


Fig 2 Agile Factor less often than Control Groups

The race for quality never ends its indecisive process where we can never be sure for coping with all users need. Improving quality and delivering less error prone software manages to win satisfied customer with long term relationship. In software development, often errors are regarded as unavoidable. The rapid delivery in agile results in lack of quality by distressing service it provides to the user and their experience with the product creating appalling subjective view about product. Due to this payoff the quality factors suffer. To overcome these factors

Cleanroom technology approach will be introduced that will embed in Agile development to attain quality products. Cleanroom analyzed the data to determine the common characteristics of the successful group. It is the set of engineering principles that supports the development of reliable software. It carries the phenomena of "hit right on first time". Failures always cost something leaving behind unsatisfied customers. In Cleanroom correctness is built in based upon the structured programming and implication of mathematical reasoning to software in a methodical way [16, 8].

In our research we investigate enhancement of QoE, QoS and QoP using Cleanroom software engineering techniques in Scrum methodology of agile development. Cleanroom entrenched in agile development methodology will help timely development and less error rate with clear customer needs, augment the users experience and service quality. After studying all these factors that lead the agile development towards success we propose an idea to divide these factors into three quality categories i.e. QoS, QoP and QoE. Each factor lies into one or more categories. A possible solution is to embed Cleanroom software engineering practices into scrum development that works by appliance of sound engineering discipline to prevent error rather than detection and removal. Paper further probes Section II covers the related work done in the field. Section III illustrated the cleanroom software engineering phases, and our proposed approach. Section IV comprises of outcome achieved after implication of proposed approach.

## II. BACKGROUN AND RELATED WORK

Complete requirement knowledge is utterly essential to the process, but getting complete information is challenge. Integrating user-centered approaches certifies to have "fulfilling required needs". Quality of experience targets what point of view user builds after using the product, how is the quality of service and what level of quality of perception we present to customer. Prior work has shown that adopting agile approaches is beneficial in management of SDLC and customer relationships, decreases the amount of overtime and increases customer satisfaction [13, 14]. QoS is critical but not sufficient, for determining user experience QoE and QoS are vital.

Agile development provides team with management structure, but it does not stipulate human factors *QoE* requirements the engineering practices a team should use. The factors coupled with users are of major significance

and of primary concern while the values associated with process come in the secondary level. If a product is to achieve its full potential, it is vital that its user interface should be designed to match the skills, experience and expectations of its anticipated users [11]. In past successful agile development depends on multiple factors such as Organizational, Compliment People to Improve Processes, Partnership, People, Project, Process, Technical, Environment, and Invest in Root Cause Analysis, Initiate Test Drives and Nature of Requirements [1],[2],[ 3], [4], [5].

Clean room uses waterfall model as its baseline and adds incremental strategy to the traditional model. It encompasses box structures to indorse the correctness of properties for each increment against the specification for that increment [14]. It's a common believe that Cleanroom techniques are too theoretical, mathematical and rely on correctness verification and statistical quality control. In same survey teams involved suggested that peer reviews conducted during Cleanroom technique helped in getting better outcome [10]. Even though applying CSE Doesn't promise to guarantee software product has zero defects, but it is possible to know that it has zero defects with high probability and with high productivity [9].

## III. THEORETICAL CONSIDERATION

### A. Quality parameters for Cleanroom Software Team (CSE)

Clean room software team approach constitutes of three quality parameters.

- i. QoS
- ii. QoE
- iii. QoP

Under the umbrella of these factors three Cleanroom Software engineering teams are defined [Fig 3]

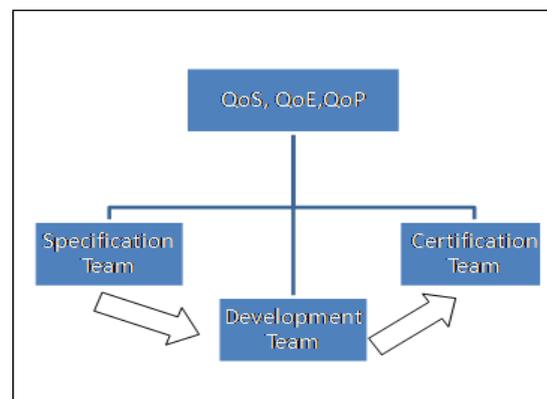


Fig 3 CSE Quality Parameters

#### 1) Specification team

Specification Team members carry out the Requirement Analysis, Function Specification, Usage

Specification, and Increment Planning activities [Fig 4]. Gathering and analyzing the requirements are finalized in cooperation with the customer. The requirements are typically documented in user terminology [15]. The Function Specification process postulates complete functional behavior of the software. An agreement is done with the customer on the specified function [15]. The Usage Specification process identifies and classifies software users, usage scenarios, and environments. The drive of Usage specification is to have user approval on the specified usages [15]. This specification identifies the scope for the testing and acts as a baseline for the incremental usage model. Usage specification also helps complete and validate the Function Specification [15]. In the Increment Planning phase team aim is allocation of user requirements into increments. Resource allocations and schedules are finalized involving the intended customer [15]. The specification team produces an Increment Construction Plan which is used by management team later to, track progress, assign tasks, and monitor product quality and process control.

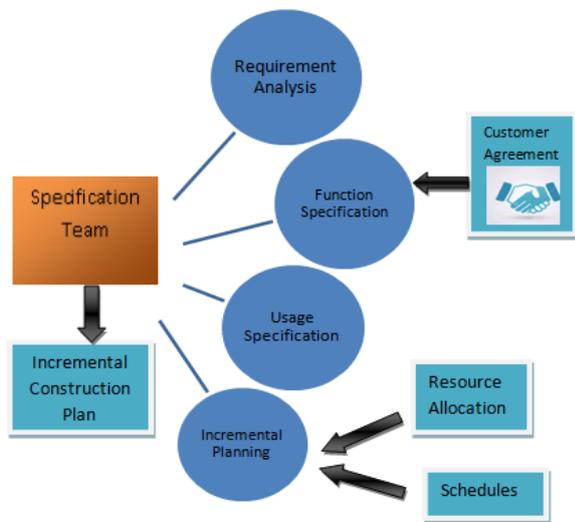


Fig 4 Specification Team Work Flow

2) *Development team*

The Development team deals in Software Reengineering, Incremental Design, and Correctness Verification. [Fig 5] The purpose for the Software Reengineering approach is to develop reused software for assimilation into the software product. The intent for the Incremental Design process is to prepare system design and implement a software increment. System Increments are designed and implemented by decomposing the box structure specifications into chunks. Correctness verification is conducted using strict approaches to verify either the correct specifications are met. Black box specifications are verified to be consistent, complete, and correct. The faulty specifications and designs are then re-reviewed and re-verified [15]. The objective of Correctness Verification is to transfer product into testing stage with aim of no fault in design. After this phase the

increment is turned over to the certification team. Code is executed for the first time by the certification team. [15].

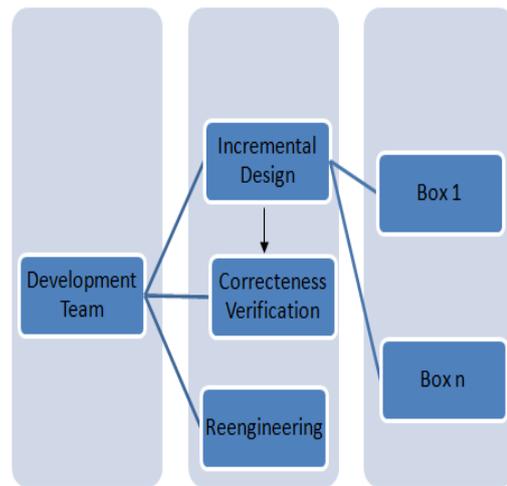


Fig 5 Development Team Work Flow

3) *Certification Team*

The Certification team handles phases of Usage Modelling, Test Planning, statistical testing and certification testing. The intention to have Usage Modelling and Test Planning process is to cultivate Usage Specification that can further help in creating models for testing, outline test plans. This also helps to have customer opinion about the models and test plans [Fig6]. Software is executed by the certification team for the very first time in statistical testing stage. Increments are compiled, working system is built, and test cases are executed and evaluated. A comparison is done between the software behavior and the one listed in function specification. Failures are identified and recorded in statistical testing report. [7] Values of certification measures are equated with the certification goals. Conclusion is adding final decision to the status of testing. Evaluations and decisions regarding product quality and process control are documented in the Increment Certification Report [7].

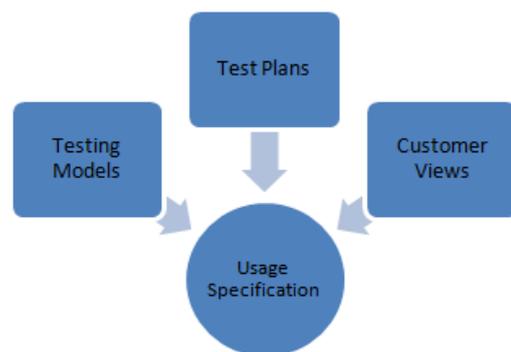


Fig 6 Usage Specification Modules

B. *Proposed Approach*

### 1) Key Quality Factors

We have proposed some factors like Organizational, Compliment People to Improve Processes, Partnership, People, Project, Process, Technical, Environment, Invest in Root Cause Analysis, Initiate Test Drives and Nature of Requirements these are essential to improve the overall quality in Agile development Software's. We proposed an idea to divide these factors into 3 quality categories i.e. Quality of Service (QoS), Quality of Perception (QoP) and Quality of Experience (QoE). In order to improve QoS we have to emphasize on following factors i.e. Organizational, Compliment People, Project, Environment, Technical, Process, Initiate Test Drives and Invest in Root Cause Analysis. QoP and QoE will be improved by following factors i.e. Partnership, People and Nature of Requirement [Fig 7]. Now we want a technique that collectively enhances all three quality categories into agile development.

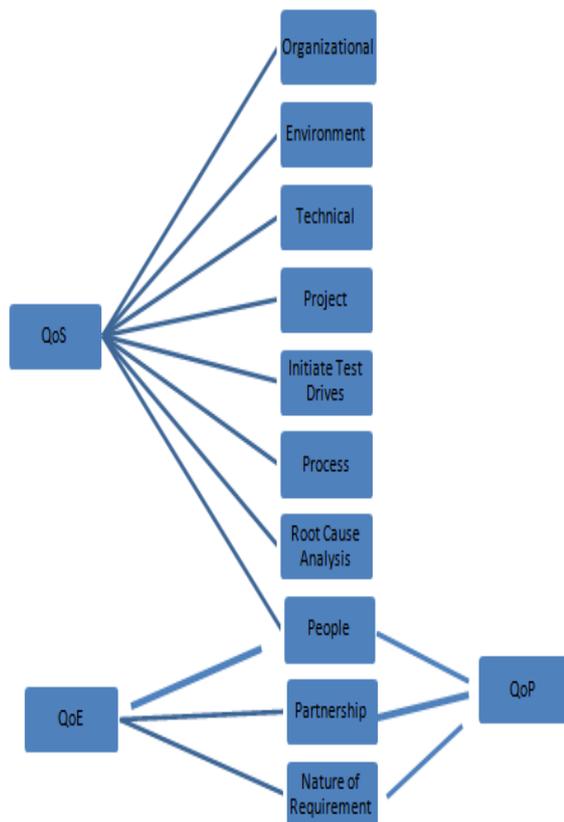


Fig 7 Proposed Quality Factors

In order to improve the above quality factors we have proposed a Cleanroom strategy. There were several appealing factors that persuaded us to embed Cleanroom software development into scrum practice of agile development.

### C. Proposed CSE Approaches

#### 1) Product Backlog

Product backlog is requirement pool, prioritized by the product owner and users. All the requirements elicited by the product owner are broken down into chunks in product backlog to calculate cumulative time taken and work to be done. This calculation technique is carried out using Burn down charts. As CSE product development is an iterative approach, increments in the Cleanroom software developments are replaced by the sprints in scrum approach [Fig 8].

#### 2) Box Principle for specification and Design

Requirements in the Cleanroom software approach are tagged statements. Figure 3 is depicting the introduction of CSE requirements approach into scrum in the very first phase of adding requirements to product backlog [Fig 8]. Specifications are refined and analyzed using strict and sound box structure process. These box structured specifications portray the behavior and usage of system and are further used in the design and development processes. Usage specification talks about the usage scenarios, environment and intended users. Functional specification depicts complete functional behavior of product to be implemented. This gives a better insight into understanding and achieving quality of user's perception and quality of experience. Design building in the figure3 is assimilation of Cleanroom and scrum strategies. Sprints will be designed and implemented using box structure decomposition. This helps in verification of having correct specifications for design.

#### 3) Code Execution in Development phase

In Cleanroom methodology developers are restricted from execution of implemented increment. Each increment for the first time is executed in statistical testing phase by the certification team. This phase comes when the development team completes the Correctness Verification process [15]. Development team doesn't get to know if system is providing correct outcome on execution. Our methodology relieves this constraint from developers as it compels the developer to implement a system without compiling it. This act will help developers to be more productive; at same time will also bug out the some errors related to implementation stage.

#### 4) Correctness Verification

In scrum practice each sprint should depict the intended functionality at end of each short iteration review section. The implication of sound engineering principles may lag behind and quality is compromised. Correctness verification is an addition to the scrum practice, which ensures each sprint meets its stated specification. Verification review approach is steered in Cleanroom to verify the software and provides correctness proofs. Black box specifications are substantiated to be consistent, complete, and correct. State box conformance is checked in correspondence with their black box specifications. Faults identified in the verification reviews are documented, amended by the specification and development teams.

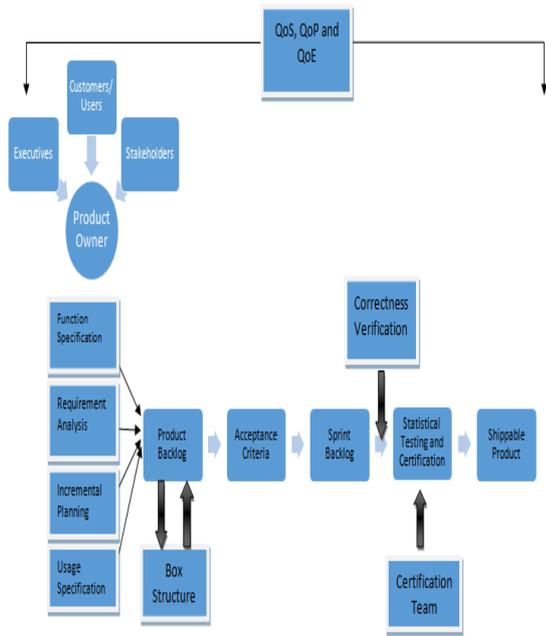


Fig 8 CSE Embedded into Scrum Approach

5) Scrum Certification Phase

Entrenchment of one separate certification team in scrum practice assures to achieve high quality product. Before demonstration of working product to users and stakeholders certification team performs statistical testing. This is final approach to bug out all the faults in the sprint and then demonstrate it to user for acceptance. Statistical testing replaces the traditional testing. Certification process is comprised of usage modelling and test cases planning along with statistical testing. Statically based independent testing generates probability weighted usage models [Fig 8]. Test cases are randomly generated based on the usage models. Usage scenarios are reviewed and generated by the customer that adds to user centered approaches. Customer provides approval for these usage models and test plans.

Usage modelling refines usage specifications; these help achieve QoP and QoE in parallel with user QoS. In the statistical testing and certification segment test cases are executed and evaluated. Success is finalized with the amount of conformance of software behavior with one mentioned in the Function specification phase. The failures found in testing are reported in statistical testing report. Measured Values are compared with certification goals. After successful demonstration of Sprint the product targeted to have zero error is presented to the user. Like all the agile methodologies, presented approach also encapsulate key agile manifesto that is “Customer Satisfaction”.

IV. PROCESS IMPLICATION RESULTS

We conducted a survey in multiple software development where scrum approach is used as development methodology. With the help of this survey we were able to produce the results of enhanced percentages of quality factors. Before and after results are shown in figures which we have achieved in agile methodology without using Cleanroom Software engineering technology and after using it. Our aim was to check how the Quality of Service, Quality of Experience and Quality of Perception improves after using CSE in agile. Our research is divided into 3 categories those are given below.

A. Improvement in Quality Factors

In our research we have define some factors that directly affect the quality of agile. Some of them lie in QoS and some of them in QoE and QoP as we have discussed above. It seems so interesting that after using CSE in agile all the quality factors improve with high percentage then before using CSE. [Fig9, Fig 10]

According to statistics, improvement in Organization factor is 21.66%, in Environment factor it is 3%, in Technical factor it is 27.92%, in Project factor it is 29%, in Initiate test drives factor it is 27.5%, in Process factor it is 38.75%, in Root cause analysis factor it is 22.5%, in People factor it is 18.34%, in Partnership factor it is 20% and in Nature of requirement it is 20.75% [Fig 11]. Improvement in quality factors has directly relation to the improvement in QoS, QoE and QoP that we will discuss in the next category.

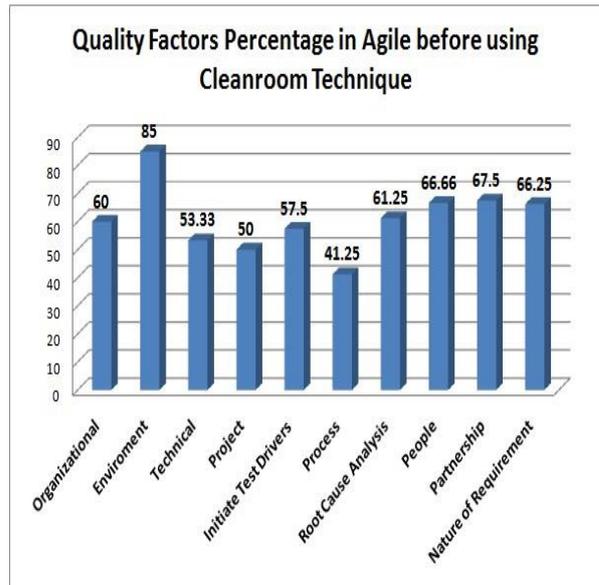


Fig 9 Quality Factors % in agile before using CSE

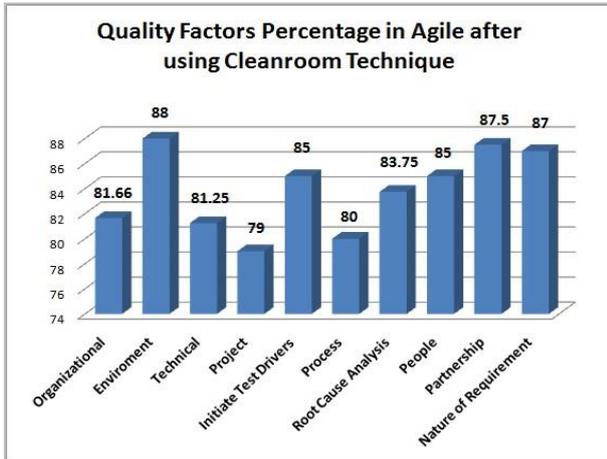


Fig 10 Quality Factors % in agile after using CSE



Fig 13 QoS in Agile Development after Using CSE

It is observed that about 23% improvements of QoS in agile has been taken out by using CSE. Similarly about 23% Quality suffering area has been decreases.

Now come towards the QoE and QoP that's directly relates to each other, drastically improvement has been recorded in both factors after using CSE. Here is the comparison.

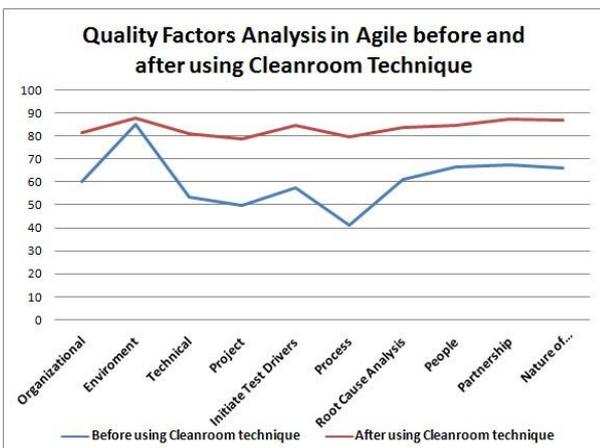


Fig 11 Quality Factors analysis before and after CSE in agile

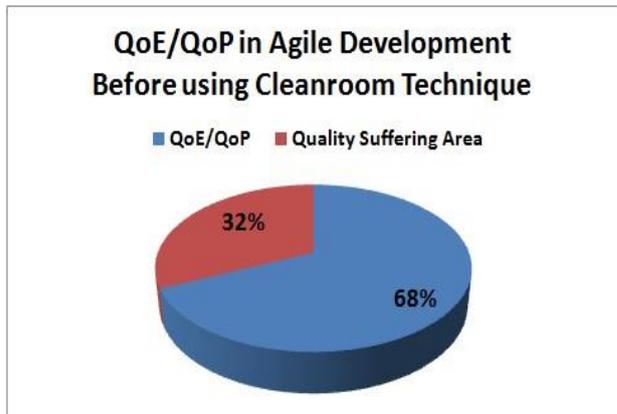


Fig 14 QoE/ QoP in Agile before CSE

a) Improvement in QoS, QoE and QoP

Main target of our research is to improve QoS, QoE and QoP in agile development and after introducing the concepts of Product backlog, box structure, correctness verification and certification team in agile leads us to come closer towards our goal. As the improvement in quality factors has done due to CSE it directly accelerate the improvement graph of QoS, QoE and QoP in agile. Here we are giving the comparison of how CSE increases the percentages of QoS, QoE and QoP in agile and how these Qualities percentage were in agile before CSE.

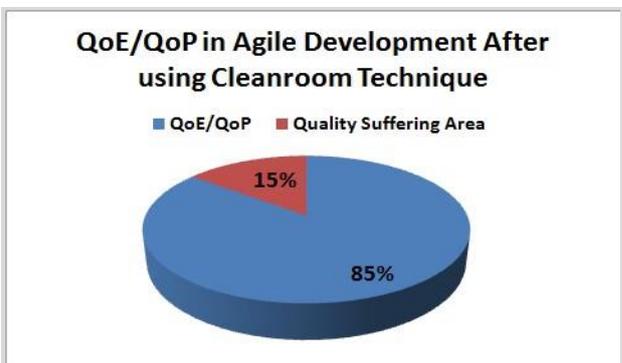


Fig.15. QoE/QoP Pie Chart of agile after CSE

In the above comparison it is observed that about 17% improvement of QoE/QoP in agile has been taken out by using CSE. Similarly about 17% Quality suffering area has been decreases.

b) Improvement in Zero Quality Control

It was one of our research targets to achieve Zero quality control in agile. As the quality of the developed software increases it helps to achieve the ZQC. Here is the analysis how much we achieve it after using CSE.

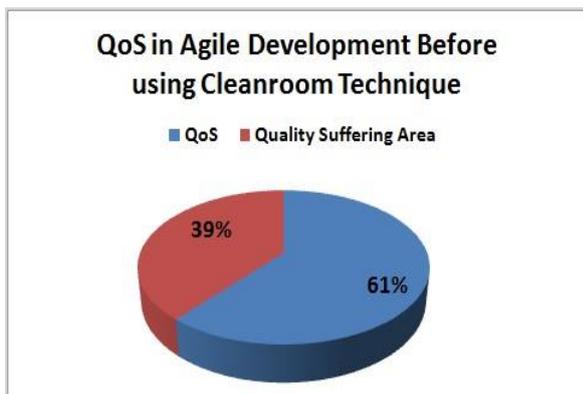


Fig 12 QoS in Agile Development before Using CSE

About 22.08% improvement in ZQC is observed after using CSE concepts in agile.

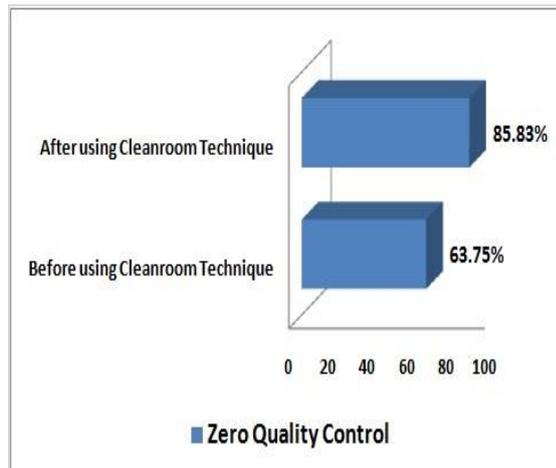


Fig 16 ZQC Before and after using CSE

## V. CONCLUSIONS

The recent focus of our work diverts simple agile methods to CSE in agile development. The objective of this study was to investigate CSE approach and embed it in scrum development. In order to explore this experience we conducted data analysis in 30 software companies capturing real-world scenarios and experiences of this proposed approach's implication in software development. The patterns of combining agile and CSE in these experience reports were identified and categorized in a more systemic way. The results were positive; all the quality factors were improved with high percentages, supporting the proposed technique's advantageous feature. By injecting the core concepts of CSE it enhanced the QoS, QoE and QoP in agile software development. Zero Quality control in software development, in order to achieve quality factor has been enhanced in significant figures as shown in above results. The agile methodology has still room for improvements, further re-research is needed to improve the quality suffering areas in agile.

## ACKNOWLEDGMENT

The authors heartily thank Eng. Ali Javed, Assistant Professor at department of Software Engineering, University of Engineering and Technology Taxila, Pakistan for his full time dedication and prestigious guidance.

## REFERENCES

- [1] Dyer and Michael, *The Cleanroom Approach to Quality software Development*, John Wiley and Sons, 1992.
- [2] Arezo Nasehi, "A Quantitative Study on Critical Success Factors in Agile Software Development Projects; Case Study IT Company", University of Boras.
- [3] Madu Ratnayake, "The Five Factors of Agile Suitability" a Blog Posted on June 29, 2010.

- [4] Subhas C. Misra, Vinod Kumar, and Uma Kumar, "Success Factors of Agile Software Development", Carleton University, Ottawa, Canada.
- [5] Tsun Chow and Dac-Buu Cao, "A survey study of critical success factors in agile software projects" School of Business and Technology, Capella University, Minneapolis, MN 55402, USA.
- [6] Zarrin Langari, Anne Banks Pidduck "Quality, Cleanroom and Formal Methods", School of Computer Science University of Waterloo Waterloo.
- [7] Mills, H.D., M.Dyer, and R.C. Linger, "Cleanroom Software Engineering", IEEE Software, pp19-25, September.1987.
- [8] Foreman, John. (1997). *Cleanroom Software Engineering* Retrieved March 27, 2006.
- [9] Richard C. Linger, "Cleanroom Software Engineering for Zero-Defect Software," 1993 IEEE.
- [10] Robert Oshana, "An Industriail Application of Cleanroom Software Engineering Benefits Through Tailoring," IEEE 31<sup>st</sup> Annual Hawaii International Conference on System Sciences, 1998.
- [11] Osama Shoaib and Khalid Khan, "Integrating Usability Engineering and Agile Software Development: A Literature Review," International Conference on Computer Design and Applications, Vol 2, No 34, 2010.
- [12] M. Ceschi, A. Silliti, G. Succi and S. S. De Panfilis, "Project Management in Plan-Based and Agile Companies," IEEE Software, 22(3), 21-27, 2005.
- [13] C. Mann and F. Maurer, "A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction," Proceedings of the Agile Development Conference, 70-79, 2005.
- [14] Zarrin Langari and Anne Banks Pidduck, "Quality, Cleanroom and Formal Methods," ACM 17 May. 2005.
- [15] Linger, Richard C., Trammell, Carmen J. (November 1996), *Cleanroom Software Engineering: Reference Model, Version 1.0*. Retrieved March 27, 2006.
- [16] Laurie Williams, A.Gabe Brown, Adam Meltzer and Nachiappan, "Scrum+ Engineering Practices: Experiences of Three Microsoft Teams," International Symposium on Empirical Software Engineering and Measurement, 2011.
- [17] "Agile Software Development and the Factors That Drive Success," September 2012, Survey Conducted by Forrester Research, Inc.

## Authors:



**Engr. Ali Javed** - is serving as an acting chairman at software engineering department at university of engineering and technology taxila, Pakistan. Engr. Ali javed remained visiting faculty at Fatima Jinnah University Rawalpindi, Pakistan from 2012- 2013. He has received a Gold Medal and MS degree in Computer Engineering from UET taxila, Pakistan in February, 2010. He has received B.Sc. degree in Software Engineering from UET taxila Pakistan in September 2007. His recent focused areas of research are Image Processing, Computer Vision, Video Summarization, Software Quality Assurance, Software Testing, Software Requirements Engineering, and Mobile Application Development.

**Munazza Jannisar Khan** is MS scholar in the Software Engineering department at UET Taxila, Pakistan. She graduated from university of engineering and technology peshawar, Pakistan in software engineering in August 2012. Her focused research areas are Software Testing and Quality Assurance, Computer Networks, Artificial Intelligence, Recommender systems and Data Mining.

**Sana e Zainab** is MS scholar in the software engineering department at university of engineering and technology taxila, Pakistan. She graduated from UET taxila, Pakistan in Software Engineering in July 2010. Her focused research areas are Model Driven Engineering, Web- Engineering, Computer Networks and Digital Signal Processing and databases.