International Journal of Information Technology and Web Engineering
Volume 14 • Issue 3 • July-September 2019

# A Customized Quality Model for Software Quality Assurance in Agile Environment

Parita Jain, Amity Institute of Information Technology, Amity University, Noida, India
Arun Sharma, Department of Information Technology, Indira Gandhi Delhi Technical University for Women, Delhi, India
Laxmi Ahuja, Amity Institute of Information Technology, Amity University, Noida, India

## ABSTRACT

The agile approach grew dramatically over traditional approaches. The methodology focuses more on rapid development, quick evaluation, quantifiable progress and continuous delivery satisfying the customer desire. In view of this, there is a need for measurement of the agile development process. In this respect, the present research work investigates the inter-relationships and inter-dependencies between the identified quality factors (QF), thereby outlining which of these QF have high driving power and dependence power, working indirectly towards the success of agile development process. This paper proposes a new agile quality model, utilizing an interpretive structural modeling (ISM) approach and the identified factors are classifies using Matriced' Impacts Croise's Multiplication Applique´e a UN Classement (MICMAC) approach. The research findings can significantly impact agile development process by understanding how these QF related to each other and how they can be adopted.

## KEYWORDS

Agile Process, ISM Approach, ISO/IEC 9126-1, MICMAC Analysis Approach, Quality Factors

## INTRODUCTION

From the past few years, the focus is more on to enhance software development practices by improving reusability, understandability of requirements, software delivery time and cost-effectiveness and many other characteristics. Quality being the most important aspect either in discrete production process or continuous production process, needs to be focused more to deliver a product that is acceptable by customers. According to International Standard Organization ISO 9000, quality is defined as the totality of characteristics of a product as a whole to satisfy the stated and implied needs in accordance with its capability. Here the stated needs mean the requirements that are given by the customer at the time of an agreement, and the implied needs are the needs that are identified by the developers as the necessary needs to be included while developing the product. Definition of quality has been perceived differently by various persons, but the one that has been given by the customer is the definition that counts the most. Quality corresponds to standards, cost of the product, conformance to requirements and value for performance (Juran & Gryna, 2010; Weinberg, 1992).

Overall quality is the much more complicated term than it appears. There are varieties of perspectives for consideration, for example, customer's perspective, developer's perspective, tester's perspective, specification based perspective, manufacturing based perspective, quality assurance based perspective and many more. In every single domain, quality is one of the most important factors for

DOI: 10.4018/IJITWE.2019070104

a product to survive in the market. Many more definitions are given by different researchers from the perspective of manufacturing the product and engineering the product respectively. Definition of quality has a contextual bias towards these two industries. A general definition is possible for all the areas; however, when applying to IT software products the context is slightly different. Some of the authors named as Meyer (2000), Pressman (2011), and Sommerville (2015) states' software quality differently but from the same perspective towards software product. The quality of the development process significantly affects the value received by customers and development teams. Hence, for high quality product it's important to concentrate more on development process as development of good quality software is usually an organizational effort, "something of higher quality has more value than of low quality".

### Need of Quality Model for Agile

Over the years, developers found that for development the traditional software development methodologies proving to be a, "well defined process for development" work poorly in practice. Moreover, from the literature, it appears that developers are not more interested to adopt traditional processes and are finding ways to reduce efforts to adopt them either intentionally or unintentionally. With the escalation in quality software's, specifying relevant development process is a necessity. Assimilating agile methodology process for development is a good option to counter ever escalating software complexity. Software organizations are ready to accept the quick approach for developing software with the availability of resources, unable to do so due to uncertainty of proving agile process quality. Initially, when introduced agile methodology was applicable on small-scale projects where the methodology proved as a best practice for software development but with the increase in utility the agile methodology also applicable on large-scale projects.

It has been observed that agile methodology is the best practice so far for developing quality software but quantification of quality parameters is still a major challenge. Besides this, in current scenario, expectations of customers, stakeholders and need of quality changed significantly. The agile methodology emerged because of two significant characteristics, as follows: firstly, it can handle customer changing needs throughout the software development life cycle (SDLC). Secondly, it can release software in shorter time with right delivery strategy within the defined cost.

The paper proposes a quality model, derived using a two stage approach. Firstly, critical quality factors (QF) are identified through a survey of literature and expert opinions from the industry, working within the field of agile development from last many years. Furthermore, the methodology interpretive structural modeling is used to develop a structural model to identify how each QF interacts with each other. Thereafter, an analysis based on driving power and dependence power is done and the QF are further categorized according to the MICMAC approach. Present study attempts to find inter-relationships and inter-dependencies between QF in agile development process which will help in identifying which QF have greater driving or dependence power than others and are thus critical quality factors towards agile development process success.

## LITERATURE REVIEW: THE FOUNDATION OF ANALYSIS

As software industry adopted agile software development methodology, it becomes inevitable to *distinctly* outline its characteristics, benefits and organizational ramifications. The agile methodology emerged as a lightweight methodology for development. It offers many benefits, such as customer involvement, greater reusability, iterative and incremental perspective of software. Various approaches are proposed and had some extent of success in improving software quality, usability, efficiency, maintainability and helping many organizations to develop large complex software's on different platforms. However, still organizations face enormous problems while developing software's.

Timperi (2004) focused on quality assurance practices of different agile methodologies providing study towards lack of balance in quality assurance activities for producing good-quality software. He found that an agile methodology proposes various quality assurance practices, excluding testing as an important factor for quality assurance. Furthermore, he concluded that none of the agile practices had any pragmatic evidence for quality assurance work in the process itself. Mnkandla and Dwolatzky (2006) in their paper stated that in existing literature, there has not been any evidence that provides an aspect of software quality improvement as the basis of any software quality characteristics. For this purpose, an innovative technique in the form of a tool has been introduced and tested on two agile methodologies' XP and lean development, determining which factors of software quality they improve. Hashmi and Baik (2007) provided an overview on quality assurance practices of the agile processes. The study reveals that still there are ongoing debates on the flexibility to change, productivity and quality of agile methodology processes. In depth, they compare XP practices with traditional spiral model from the perspective of ensuring product quality, built in QA practices in SDLC of XP and productiveness.

A literature reviewed up to 2009 given by Sfetsos and Stamelos (2010) identified 535 studies of which 46 were found to be empirical studies regarding agile practices as a quality process. They classified their study into three clusters to evaluate quality perspective in agile methodologies. The findings from the study show that if agile practices have implemented in a correct manner, they can improve product quality. In addition, they reported wide range of improvements in various phases of agile practices in order to provide good-quality software. Based on the existing literature, the authors Imreh and Raisinghani (2011) found that not much research has been done in the field of quality of agile software development. Some of the IT organizations found the new practices that are successful, but some of the experiences showed them as less successful. The paper provided a study that focuses more on the quality aspects rather than the technical aspects of agile development.

Besides the several recent studies claiming very less work for quality of agile approaches, sufficient studies are there that proves popularity of different practices within agile methodologies and comparison between them showing again the lack of quality aspects in all of them. Galvan et al. (2015) presented a study that provides an overview of at what ratio SCRUM and XP agile methodologies coping up with ISO/IEC 2911 standard. The study shows that software developers are more interested in acceptance with project management in SCRUM and XP practices and less care about software implementation, which will indirectly, affects the quality of the software product. The work done by Torgeir and Lassenius (2016) focused on emerging practices in agile software development. There is an increase in interest on continuous integration from 2006 to 2015 in usage of some of the practices. The Scrum being the most popular practices; more research has been done on the Scrum practices in comparison to Extreme programming and other agile practices. They also identified some of the key research areas that can be taken up in the future as a significant area of study such as focus on an organization processes, product value and many more.

A systematic review of large-scale agile transformations from traditional to agile practices conducted by Dikert et al. (2016) identifying 35 reported challenges grouped into nine categories and 29 success factors grouped into 11 categories. The study presents that still agile methodology lacks behind in some of the categories that reported as most important ones and research is seriously lagging behind with the growing adoption of agile practices for software development. Another paper presented by Papadopoulos (2015) focuses on how well light weight agile methodologies perform better than heavy weight software development methodologies. With the analysis of a case study author found evidence for the same. With the effectiveness of agile over traditional, he also observed that still there is a scope of improvement in the quality aspect and customer perception aspect of the end product in agile practices. Indicating on growing need of agile practices within the software industry for the development of software rather than adopting heavy weight traditional practices, review of an ample number of research papers presented by Jain et al. (2016) discussed that the traditional development incorporates a QA (quality assurance) factors in the process while in agile methodologies, there is no separate QA factor that must be taken care of, they just rely on routine activities of the development team for QA of the process and product. Hence, still there is a need to deal with backlogs of quality assurance for agile practices.

Focusing on lack of quality aspects, various studies have been carried out to overcome with this problem. For this Opelt and Beeson (2008) provided steps to integrate QA into the agile process. As the study shows that agile processes did not include QA into practices because of the reason that QA might slow down the software development process. They further discussed why the role of QA is important for balancing quality of product with the needs of the business. Some of the researchers in addition presented a quality evaluation scheme that focuses on the reliability aspect of software quality in iterative agile development based on traditional metrics. Jinzenji et al. (2013) presented this scheme due to the lack of management of software quality in agile methodologies. At the end with the experimental results, they showed that traditional metrics quality evaluation scheme can be applicable on iterative development from the aspect of conventional reliability metrics only.

Olszewska et al. (2016) presented a metrics model to measure quantitatively the influence of agile methodologies' adoption for software development in organizations. The study done, also showed that lot of research has been done on the adoption of agile practices but in the form of success stories and case studies, mostly of qualitative in nature. Hence, there is a need to prove the impact of growing agile software development empirically. They used goal question metric approach providing a quantitative metrics model declaring significant improvements in 6 of the metrics and decline in one of the metrics taken in study. Finding the limitations of agile practices in industry subjectively Agarwal et al. (2016) applied ANNOVA test based on the online survey provided by agile experienced professionals to satisfy the literature quantitatively. According to the literature, they found out one of the major flaws in adopting agile methods, lacked of upfront planning which overall effects the process quality and product quality both.

Among the literature that exists, very few agile quality models are presented and evaluated in a universal manner for agile practices. Sanghoon Jeon et al. (2011) focused on the roadblocks of SCRUM methodology. With the literature, they found fall in the use of SCRUM practices in the aspect that it only concentrates on functional features rather it should also target on software quality aspects. Due to this reason they presented a new quality attribute driven agile development method named ACRUM that is derived from SCRUM practices including quality aspects as well. The model proposed proves to be efficient, but it can only be used on behalf of SCRUM practice. One cannot take the proposed quality model on behalf of all the agile practices that are used for software development in industries.

Bajnaid et al. (2012) addressed the shortcomings of agile practices of not including quality assurance as a part of their process to ensure whether the quality assurance process has been followed and quality assurance standards have been met or not. To overcome the disadvantages, they proposed a process driven e-learning system that senses developer's activities and guides them through required software quality assurance practices while development of the software. Another framework proposed by Sagheer et al. (2015) as a solution to an issue of analyzing whether quality has been assured in the software which is being developed using agile software development methodology and recapitulate that how quality can be assured in agile practices using different quality factors. Discussing the limitations of various quality models proposed previously in the paper, Jain et al. (2016) presented a new framework for measuring agile process quality using an interpretive structural modeling approach so that developers can focus their efforts on the identified attributes for deploying higher-quality products. The model identified the most critical factors important for agile quality development but did not provide any empirical solution for evaluating the quality. That is, there is a need to measure the proposed model. Additionally, the model did not include factors on which agile practices actually depends upon.

Various quality models have been proposed taking quality perspective into account as a whole for software development but still organizations are in a way for finding out a solution for agile quality assurance. Hence, there is a need for a quality model that includes all the important factors of the agile development process on which distinct practices of agile development depend, verifying the model to be holistically substantiated for all the defined agile practices.

### Approaches for Decision Making

Interpretive Structural Modeling is a well establish methodology used for depicting interrelationship among varied variables that define the problem to be solved. It is one of the popular techniques that have been used for multi-criteria decision making in different fields of research. The literature, in the contrary presents different other approaches like analytic network process (ANP), analytic hierarchy process (AHP), decision making trial and evaluation laboratory (DEMATEL) and many more techniques that are useful in almost all problems related to decision making. Several review papers are present in the literature which elaborate distinctive methods of multi-criteria decision making. However, these papers mainly include the comparison between different multi-criteria decision making methods, as well as their applications in various fields of research.

Velasquez and Hester (2013) provide an analysis of commonly used multi-criteria decision making (MCDM) methods, identifying their advantages and disadvantages and explaining how these methods can be used, in particular, applications. Majumder (2015) focuses on basic working principle of different MCDM methods and specifically identifies limitations of AHP method. Gayatri and Chetan (2013) also provide the comparison of varying MCDM methods, which are gaining importance for solving a specific problem. Thakkar et al. (2008) have compared three MCDM techniques, ISM, ANP and AHP and extract various advantages of ISM in unusual areas for modeling and solving decision making problems over other two techniques. Many researchers reviewed and compared different MCDM methods and found ISM as a method of solving problems, which are subjective in nature, has higher ability for capturing dynamic complexities, proving ISM approach better than other decision making approaches.

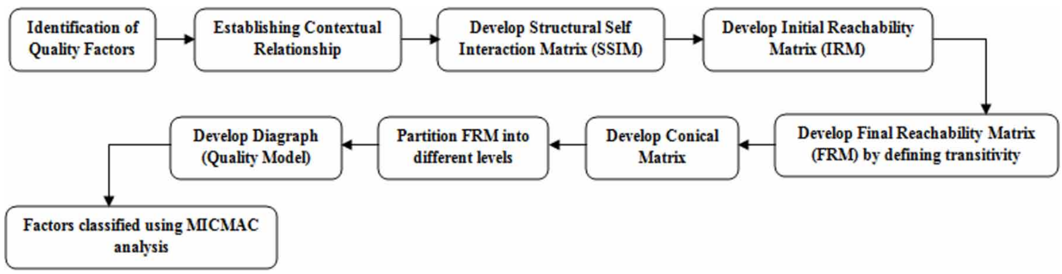
## PROPOSED AGILE QUALITY MODEL

The quality model proposed for agile development refers to all the agile quality attributes or factors that must be included within the model to assure quality of the agile practices, and the product deployed after processing. With the identified quality factors based on literature review and experts' opinions from the software industry, working within the field of agile development from last many years, a new quality model based on ISO 9126-1 is proposed in the present paper. The model proposed after specifying the quality factors and applying ISM approach to establish interrelationships among these recognized factors and extracting an overall tree-like structure, which further portrays a hierarchical structure model. After that, a MICMAC analysis approach is applied that provides insight on the impact of quality factors in a holistic way. The step by step procedure for developing a model is shown in Figure 1.

For the present work, ISO/IEC 9126-1 standard quality model introduced by ISO and IEC in 2001 is taken as a basis. The model includes six quality factors, identifying both internal and external characteristics of the software product. Factors that are taken as important factors for quality assurance are functionality, reliability, usability, efficiency, maintainability and portability. ISO/IEC 25010:2011 is a quality model that replaces ISO/IEC 9126-1:2001 quality model, which has been technically revised after ISO/IEC 9126:1991. This model is also a part of SQuaRE series of international standards. In comparison with ISO/IEC 9126, the model ISO/IEC 25010:2011 incorporates security and compatibility with the existing characteristics as amendments in the model (Al-Qutaish & Al-Sarayreh, 2008; Idri et al., 2013; Samadhiya et al., 2010).

The proposed agile quality model specifically based upon ISO/IEC 9126-1 quality model, as when analyzed, it has been identified that only four of the quality characteristics can be taken up, for solving the defined problem or issue, validating the process quality of agile methodologies. For this, ISM is applied on the identified 19 quality factors and at the end a customized agile quality model is proposed in the form of a hierarchal structural model. ISM is an interactive learning process used to identify the structural model that illustrates inter-relationships among identified 19 quality factors (Jain et al., 2016).

**Figure 1. Step by step procedure**




Step 1: It starts with an identification of QF, which are applicable to the defined problem or issue (Sohani & Sohani, 2012). Then the contextual relationship between the QF is determined with respect to which inter-relationships and inter-dependencies are chosen. Initially, literature survey and brainstorming with industry expert's opinion was used to develop the initial matrix Structural Self Interaction Matrix (SSIM) as depicted in Figure 2, where each symbol has some meaning to it according to the symbol interpretation as shown in Table 1.
Step 2: Replacing each cell in SSIM by binary numbers' 0s or 1s, an Initial Reachability Matrix (IRM) is developed based on the rules defined in Table 2.
Step 3: Then Final Reachability Matrix (FRM) is obtained, defining the transitivity which states that, if any variable A is related to variable B and variable B is related to variable C then A is related to C. The FRM for the given QF is depicted in Figure 3 where the value 1* depict transitivity.
Step 4: Thereafter, Driving Power and Dependence of QF are calculated. The Driving Power of the factors' is derived by summing the entries of one in the rows and the Dependence of the factors' is derived by summing the entries of 1's in the columns. The matrix obtained called as Conical Matrix as shown in Figure 4.
Step 5: To achieve Level Partitioning of the QF in agile development process, the Final Reachability Matrix undergoes recurrent iterations. In Agile Development Process, it was achieved in five iterations thus; partitioning of the QF in five levels is depicted in Table 3.
Step 6: Then according to the levels identified, a diagraph is developed showing the hierarchal dependence of quality factors. The QF having minimum influence gets isolated first and is placed at the top of the diagraph. While on the other hand, the QF having the maximum influence hold till the last and placed at the bottom of the diagraph. The diagraph illustrates how the QF influence each other and how QF are influenced by others. The diagraph portrays an "Agile Quality Model" describing the hierarchal inter-dependence between the quality factors in agile development process. This is represented in Figure 5.
Step 7: After structuring QF in levels, MICMAC analysis is performed based on Driving Power and Dependence of QF in agile development process. The analysis is based on cross-impact matrix multiplication and classifies the critical agile quality factors into four clusters, named as Autonomous, Dependent, Linkage and Driving.

## VALIDATION OF THE PROPOSED MODEL

After identifying the factors and structuring them at different levels, MICMAC analysis is used for analysis purpose. It is a validation method that provides insight on the impact of factors on the total system. That is, the analysis is mainly to identify key variables or factors that are important to overall system changes. It is basically a cross-impact matrix multiplication applied to classification. MICMAC method was given by Godet (1986), which is based on cross-impact matrix multiplication

International Journal of Information Technology and Web Engineering
Volume 14 • Issue 3 • July-September 2019

Table 1. SSIM symbol interpretation

| Symbol | Meaning | Interpretation |
|---|---|---|
| **V** | $i \longrightarrow j$ | variable i influence variable j |
| **A** | $j \longrightarrow i$ | variable j influence variable i |
| **X** | $i \longleftrightarrow j$ | variable i and j influence each other |
| **O** | $i \overset{/}{\longleftrightarrow} j$ | variable i and j will not influence each other |

Figure 2. Structural self interaction matrix for QF in agile development process

| | | QF1 | QF2 | QF3 | QF4 | QF5 | QF6 | QF7 | QF8 | QF9 | QF10 | QF11 | QF12 | QF13 | QF14 | QF15 | QF16 | QF17 | QF18 | QF19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Functionality | QF1 | 1 | O | A | A | A | X | A | V | A | A | V | A | A | V | A | O | O | A | A |
| Committed Story Points | QF2 | | 1 | X | O | V | O | X | O | V | O | O | V | O | V | V | V | O | O | O |
| Total Numbers of User Stories | QF3 | | | 1 | V | V | O | V | O | V | V | O | V | O | O | O | V | O | V | O |
| Functional Completeness | QF4 | | | | 1 | A | O | A | A | A | A | X | V | O | O | O | X | O | A | O |
| Sprint Stretch Factor | QF5 | | | | | 1 | V | A | V | X | V | V | V | V | V | A | O | A | V | O |
| Efficiency | QF6 | | | | | | 1 | O | O | A | X | O | O | O | O | A | X | A | X | X |
| User Stories Planned | QF7 | | | | | | | 1 | O | O | X | O | V | O | O | V | V | V | X | O |
| Post Delivery Defects | QF8 | | | | | | | | 1 | X | A | X | O | A | X | O | O | O | O | X |
| Defect Removal Efficiency | QF9 | | | | | | | | | 1 | V | V | X | A | O | O | O | O | O | X |
| User Stories Delivered | QF10 | | | | | | | | | | 1 | O | X | A | O | A | X | A | X | A |
| Maintainability | QF11 | | | | | | | | | | | 1 | X | A | X | O | O | O | O | O |
| Functional Correctness | QF12 | | | | | | | | | | | | 1 | X | X | O | A | O | A | A |
| Pre Delivery Defects | QF13 | | | | | | | | | | | | | 1 | O | O | O | A | O | A |
| Reliability | QF14 | | | | | | | | | | | | | | 1 | O | A | O | O | O |
| Planned Effort | QF15 | | | | | | | | | | | | | | | 1 | O | V | X | O |
| Earned Story Points | QF16 | | | | | | | | | | | | | | | | 1 | A | O | O |
| Actual Effort | QF17 | | | | | | | | | | | | | | | | | 1 | V | O |
| User Stories Delivered per Sprint | QF18 | | | | | | | | | | | | | | | | | | 1 | A |
| Defects Removed per Sprint | QF19 | | | | | | | | | | | | | | | | | | | 1 |

Table 2. Rules for converting SSIM to IRM

| Symbol | Rules for replacement |
|---|---|
| V | (i, j)th entry becomes 1 and (j, i)th entry becomes 0 |
| A | (j, i)th entry becomes 1 and (i, j)th entry becomes 0 |
| X | (i, j)th entry becomes 1 and (j, i)th entry also becomes 1 |
| O | (i, j)th entry becomes 0 and (j, i)th entry also becomes 0 |

Figure 3. FRM for QF in agile development process

| | QF1 | QF2 | QF3 | QF4 | QF5 | QF6 | QF7 | QF8 | QF9 | QF10 | QF11 | QF12 | QF13 | QF14 | QF15 | QF16 | QF17 | QF18 | QF19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QF1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| QF2 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1* | 1* | 1 | 1* | 1 | 1 | 1 | 0 | 1* | 0 |
| QF3 | 1 | 1 | 1 | 1 | 1 | 1* | 1 | 1 | 1 | 1 | 1* | 1 | 1* | 1* | 0 | 1 | 1* | 1 | 0 |
| QF4 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1* | 0 | 1 | 0 | 0 | 0 |
| QF5 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1* | 0 | 1 | 1* |
| QF6 | 1 | 0 | 0 | 1* | 0 | 1 | 0 | 1* | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| QF7 | 1 | 1 | 0 | 1 | 1* | 1* | 1 | 0 | 0 | 1 | 1* | 1 | 0 | 1* | 1 | 1 | 1 | 1 | 0 |
| QF8 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| QF9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1* | 0 | 0 | 0 | 1* | 1 |
| QF10 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1* | 1 | 0 | 1* | 0 | 1 | 0 | 1 | 0 |
| QF11 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| QF12 | 1 | 0 | 0 | 0 | 0 | 1* | 0 | 1* | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| QF13 | 1 | 0 | 0 | 1* | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1* | 0 | 1* | 0 | 1* | 0 |
| QF14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| QF15 | 1 | 0 | 0 | 1* | 1 | 1 | 0 | 1* | 1* | 1 | 1* | 1* | 1* | 1 | 1 | 1* | 1 | 1 | 1* |
| QF16 | 1* | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1* | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| QF17 | 1* | 0 | 0 | 1 | 1 | 1 | 0 | 1* | 1* | 1 | 1* | 1* | 1 | 1 | 0 | 1 | 1 | 1 | 1* |
| QF18 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1* | 1 | 0 | 1* | 1 | 1* | 0 | 1 | 0 |
| QF19 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1* | 1 | 1 | 1* | 0 | 1* | 0 | 1 | 1 |

used in the analysis of indirect and hidden relationships among the factors of the agile quality model obtained using ISM technique.

All the quality factors have been classified into four clusters based on their Driving Power and Dependence Power as depicted in Figure 6 (Sohani & Sohani, 2012) named as:

- Autonomous or Excluded variables: The cluster includes those variables having weak driving power and weak dependence power.
- Dependent variables: The cluster includes those variables having weak driving power but strong dependence power.
- Linkage or Relay variables: The cluster includes those variables having strong driving power as well as strong dependence power.
- Independent variables: The cluster includes those variables that influence the most. That is, the variables having strong driving power but weak dependence power.

In this research work, an MICMAC software tool named as MICMAC (Identification of Keys Variables) has been used for MICMAC analysis on the proposed new agile quality model, which was developed by French Computer Innovation Institute 3IE (Institute d'Innovation Informatique pour l'Entreprise), given by LIPSOR Prospective (foresight) Strategic and Organizational Research Laboratory.

**Figure 4. Conical matrix for QF in agile development process**

| | QF1 | QF2 | QF3 | QF4 | QF5 | QF6 | QF7 | QF8 | QF9 | QF10 | QF11 | QF12 | QF13 | QF14 | QF15 | QF16 | QF17 | QF18 | QF19 | Driving Power |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QF1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 |
| QF2 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1* | 1* | 1 | 1* | 1 | 1 | 1 | 0 | 1* | 0 | 14 |
| QF3 | 1 | 1 | 1 | 1 | 1 | 1* | 1 | 1 | 1 | 1 | 1* | 1 | 1* | 1* | 0 | 1 | 1* | 1 | 0 | 17 |
| QF4 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1* | 0 | 1 | 0 | 0 | 0 | 6 |
| QF5 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1* | 0 | 1 | 1* | 14 |
| QF6 | 1 | 0 | 0 | 1* | 0 | 1 | 0 | 1* | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 8 |
| QF7 | 1 | 1 | 0 | 1 | 1* | 1* | 1 | 0 | 0 | 1 | 1* | 1 | 0 | 1* | 1 | 1 | 1 | 1 | 0 | 14 |
| QF8 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 6 |
| QF9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1* | 0 | 0 | 0 | 1* | 1 | 12 |
| QF10 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1* | 1 | 0 | 1* | 0 | 1 | 0 | 1 | 0 | 11 |
| QF11 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 6 |
| QF12 | 1 | 0 | 0 | 0 | 0 | 1* | 0 | 1* | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 9 |
| QF13 | 1 | 0 | 0 | 1* | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1* | 0 | 1* | 0 | 1* | 0 | 11 |
| QF14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| QF15 | 1 | 0 | 0 | 1* | 1 | 1 | 0 | 1* | 1* | 1 | 1* | 1* | 1* | 1 | 1 | 1* | 1 | 1 | 1* | 16 |
| QF16 | 1* | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1* | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 8 |
| QF17 | 1* | 0 | 0 | 1 | 1 | 1 | 0 | 1* | 1* | 1 | 1* | 1* | 1 | 1 | 0 | 1 | 1 | 1 | 1* | 15 |
| QF18 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1* | 1 | 0 | 1* | 1 | 1* | 0 | 1 | 0 | 12 |
| QF19 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1* | 1 | 1 | 1* | 0 | 1* | 0 | 1 | 1 | 12 |
| Dependence | 17 | 3 | 2 | 14 | 7 | 13 | 5 | 15 | 10 | 14 | 18 | 16 | 8 | 18 | 4 | 13 | 4 | 12 | 7 | |

## RESULTS AND DISCUSSIONS

In this research work, on performing MICMAC analysis in Figure 6, based on Driving Power and Dependence of QF in agile development process. It has been found that no quality factor is emerging in 3rd cluster as an Autonomous variable, thus depicting that none of the QF are autonomous and hence all the selected 19 QF are contributing towards the success of agile development process. In the 4th cluster, QF 1,4,6,8,11,12,14 and 16 are emerging as dependent variables having high Dependence Power and low Driving Power and hence are dependent upon other QF having high Driving Power. Analyzing the first cluster, three QF emerged as linkage variables and are considered as powerful factors having both high Driving Power as well as high Dependence Power. Focusing more on these variables will lead to increase success rates for agile development process.

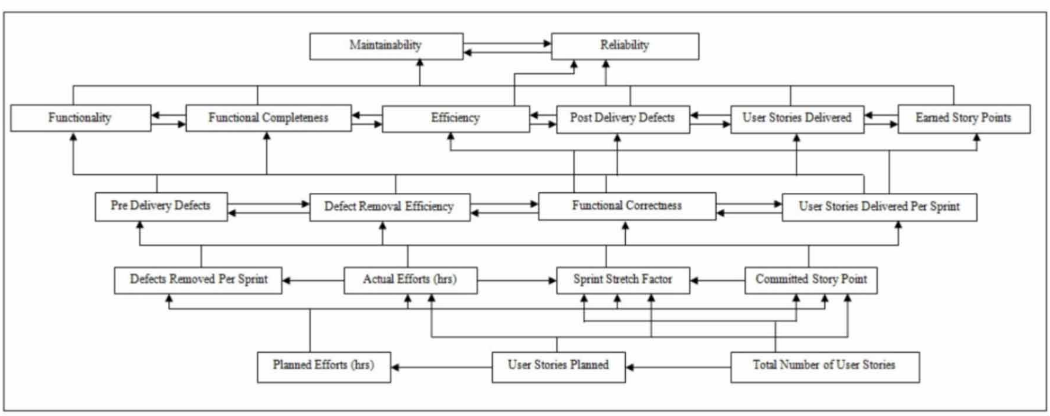
On the analysis of the 2nd cluster, it has been found that eight QF emerged as independent variables, i.e QF 2, 3, 5, 7, 13, 15, 17 and 19. They have high Driving Power and low Dependence Power. These factors should be handled carefully as they highly affect the agile development process as a whole. Quality factors having high driving and dependent power are the strongest QF as they overall define the quality of the whole development process. Both the powers provide valuable insight about the importance and inter-dependence of quality factors on each other.

Thus, with this MICMAC analysis, all the 19 QF have been segregated into different levels based on their Driving Power and Dependence Power. Also, QF has been categorized into four clusters as Linkage, Independent, Autonomous and Dependent. Focusing more on Independent QF by development team members during agile process, together with Linkage QF leads to success of agile development process. Also, the model proposed has been validated as both the hierarchal

**Table 3. Level segregation of QF in agile development process**

| | Quality Factors | Level |
|---|---|---|
| 1 | Functionality | II |
| 2 | Committed Story Points | IV |
| 3 | Total Numbers of User Stories | V |
| 4 | Functional Completeness | II |
| 5 | Sprint Stretch Factor | IV |
| 6 | Efficiency | II |
| 7 | User Stories Planned | V |
| 8 | Post Delivery Defects | II |
| 9 | Defect Removal Efficiency | III |
| 10 | User Stories Delivered | II |
| 11 | Maintainability | I |
| 12 | Functional Correctness | III |
| 13 | Pre Delivery Defects | III |
| 14 | Reliability | I |
| 15 | Planned Effort | V |
| 16 | Earned Story Points | II |
| 17 | Actual Effort | IV |
| 18 | User Stories Delivered per Sprint | III |
| 19 | Defects Removed per Sprint | IV |

**Figure 5. Diagraph depicting QF levels in agile development process/ agile quality model**




structure obtained with ISM approach and the analysis done through MICMAC giving almost the same results as discussed in Table 4. Hence, the new proposed customized agile quality model can be used for empirically evaluating the agile process quality in a holistic manner.

**Figure 6. MICMAC analysis for QF in agile development process**

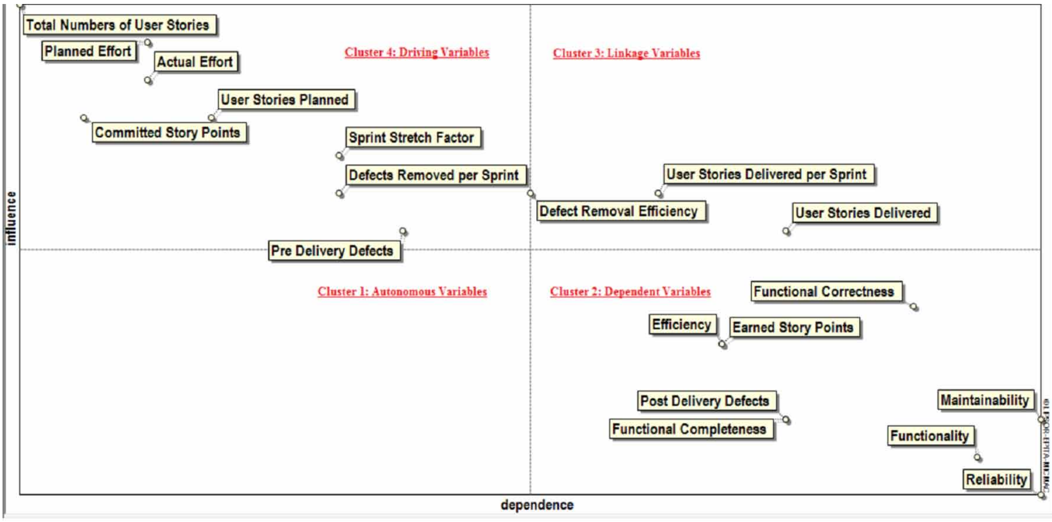



**Table 4. Results of validation**

| Classification of Quality Factors | MICMAC Validation Approach | ISM Modeling Approach |
|---|---|---|
| **Autonomous Variables** | No quality factor (QF) is emerging in 1st cluster as an autonomous variable | Using ISM, in the model no quality factor (QF) emerged as a variable that can be excluded or not important. |
| **Dependent Variables** | In this cluster, QF- 1,4,6,8,11,12,14 and 16 are emerging as dependent variables having high dependence power and low driving power. | Within the quality model, QF- 1,4,6,8 and 16 emerged at level II depending on other factors for evaluation. Within this, QF-11, 14 emerged at level I having highest dependence power. Also, QF-12 emerged at level III defining less dependency with some driving power. |
| **Linkage Variables** | QF- 9, 10 and 18 are emerging as linkage variables in 3rd cluster and are considered as powerful factors. | QF- 9 and 18 emerged at level III. The variables overall link the dependent and driving variables by providing moderate behavior. The QF- 10 emerged at level II. |
| **Driving Variables** | In 4th cluster, large number of quality factors emerged as independent variables. QF- 2, 3, 5, 7, 13, 15, 17 and 19. They have high driving power and low dependence power. | QF- 3, 7 and 15 emerged at level V, providing the highest influence or driving power on other factors. The QF- 2,5,17 and 19 emerged at level IV and QF-13 emerged at level III. |

## CONCLUSION

To access the quality of agile development process is still a major challenge for software practitioners though very few studies have been carried out on accepting agile process quality and deploying a quality product however, still the process quality for agile methodologies has not been proven quantitatively. The research work aims to find out the impact of different quality factors in agile development process of software development. A customized new agile quality model based on ISO/

IEC 9126-1 is proposed using interpretive structural modeling approach for modeling of identified quality factors and analyzing the model with MICMAC analysis. With this, inter-relationships and inter-dependencies between the quality factors in agile development process were identified thereby improving the agile process quality and proving it quantitatively in a holistic manner. In the present research work, total 19 quality factors have been identified and analyzed, proving to be important at different levels through their hierarchical structure. Prior knowledge of quality factors, having high influence on the success of agile development process would help agile project development team members to provide their focus more towards these quality factors in order to get a quality product as an outcome. Thus, focusing more on limited quality factors, success of agile development process can be ensured. The scope of the present research work is limited with a theoretical validation and proving it with the real life projects. Further, we are working on the limitation by implementing proposed quality model on some real life application developed with agile methodology.

## REFERENCES

Agrawala, A., Atiq, M. A., & Maurya, L. S. (2016). A Current Study on the Limitations of Agile Methods in Industry Using Secure Google Forms. *Procedia Computer Science*, *78*, 291–297. doi:10.1016/j.procs.2016.02.056

Al-Qutaish, R. E., & Al-Sarayreh, K. (2008). Software Process and Product ISO Standards: A Comprehensive Survey. *European Journal of Scientific Research*, *19*(2), 289–303.

Bajnaid, N., Benlamri, R., & Cogan, B. (2012). An SQA e-Learning System for Agile Software Development. In R. Benlamri (Ed.), *Networked Digital Technologies. NDT 2012. Communications in Computer and Information Science* (Vol. 294, pp. 69–83). Springer; . doi:10.1007/978-3-642-30567-2_7

Dikert, K., Passivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, *119*, 87–108. doi:10.1016/j.jss.2016.06.013

Dingsøyr, T., & Lassenius, C. (2016). Emerging themes in agile software development: Introduction to the special section on continuous value delivery. *Information and Software Technology*, *77*, 56–60. doi:10.1016/j.infsof.2016.04.018

Galvan, S., Mora, M., O'Connor, R. V., Acosta, F., & Alvarez, F. (2015). A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process. *Procedia Computer Science*, *64*, 188–195. doi:10.1016/j.procs.2015.08.480

Gayatri, S., & Chetan, S. (2013). Comparative Study of Different Multi-criteria Decision-making Methods. *International Journal on Advanced Computer Theory and Engineering*, *2*(4), 9–12.

Godet, M. (1986). Introduction to La Prospective: Seven Key Ideas and One Scenario Method. *Futures*, *18*(2), 134–157. doi:10.1016/0016-3287(86)90094-7

Hashmi, S. I., & Baik, J. (2007). Software Quality Assurance in XP and Spiral - A Comparative Study. In *Proceedings of the International Conference on Computational Science and Applications* (pp. 367-374).

Idri, A., Moumane, K., & Abran, A. (2013). On the Use of Software Quality Standard ISO / IEC 9126 in Mobile Environments. In *Proceedings of the 20th Asia-Pacific Software Engineering Conference (APSEC),* (pp. 1-8). Bangkok, Thailand

Imreh, R., & Raisinghani, M. S. (2011). Impact of Agile Software Development on Quality within Information Technology Organizations. *Journal of Emerging Trends in Computing and Information Sciences*, *2*(10), 460–475.

Jain, P., Ahuja, L., & Sharma, A. (2016). Current State of the Research in Agile Quality Development. *Proceedings of the International Conference on Computing for Sustainable Global Development* (pp. 1877-1879). Delhi, India.

Jain, P., Sharma, A., & Ahuja, L. (2016). ISM Based Identification of Quality Attributes for Agile Development. In *Proceedings of the 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)* (pp. 615-619). Noida, India. 10.1109/ICRITO.2016.7785028

Jeon, S., Han, M., Lee, E., & Lee, K. (2011). Quality Attribute driven Agile Development. *Proceedings of the Ninth International Conference on Software Engineering Research, Management and Applications (SERA)* (pp. 203-210). Baltimore, Maryland, USA.

Jinzenji, K., Hoshino, T., Williams, L., & Takahashi, K. (2013). An Experience Report for Software Quality Evaluation in Highly Iterative Development Methodology Using Traditional Metrics. *IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 310-319). Southern California, United States.

Juran, J. M., & Gryna, F. M. (2010). *Juran's Quality Control Handbook*. New York: McGraw-Hill.

Majumdar, M. (2015). Multi Criteria Decision Making In: *Impact of Urbanization on Water Shortage in Face of Climatic Aberrations*, Part of the series Springer Briefs in Water Science and Technology (pp. 35-47). Singapore. 10.1007/978-981-4560-73-3_2

Meyer, B. (2000). *Object-Oriented Software Construction*. New Jersey: Prentice Hall PTR.

Mnkandla, E., & Dwolatzky, B. (2006). Defining Agile Software Quality Assurance. *Proceedings of the International Conference on Software Engineering Advances (ICSEA'06)* (pp. 36-42). Papeete, Tahiti.

Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., & Porres, I. (2016). Quantitatively measuring a large-scale agile transformation. *Journal of Systems and Software*, *117*, 258–273. doi:10.1016/j.jss.2016.03.029

Opelt, K., & Beeson, T. (2008). Agile Teams Require Agile QA: How to make it work, an experience report. Paper presented at the International Conference on Practical Software Quality and Testing (PSQT 2008 West), Las Vegas, Nevada.

Papadopoulos, G. (2015). Moving from traditional to agile software development methodologies also on large, distributed projects. *Procedia: Social and Behavioral Sciences*, *175*, 455–463. doi:10.1016/j.sbspro.2015.01.1223

Pressman, R. S. (2011). *Software Engineering a Practitioner's Approach*. New York: McGraw-Hill.

Sagheer, M., Zafar, T., & Sirshar, M. (2015). A Framework For Software Quality Assurance Using Agile Methodology. *International Journal of Scientific & Technology Research*, *4*(2), 44–50.

Samadhiya, D., Wang, S. H., & Chen, D. (2010). Quality models: Role and value in software engineering. In *Proceedings of the 2nd International Conference on Software Technology and Engineering* (pp. 320–324). San Juan, Puerto Rico: ICSTE; . doi:10.1109/ICSTE.2010.5608852

Sfetsos, P., & Stamelos, I. (2010). Empirical Studies on Quality in Agile Practices: A Systematic Literature Review. In *Proceedings of the Seventh International Conference on the Quality of Information and Communications Technology,* Porto, Portugal (pp. 44-53). 10.1109/QUATIC.2010.17

Sohani, N., & Sohani, N. (2012). Developing Interpretive Structural Model for Quality Framework in Higher Education: Indian Context. *Journal of Engineering, Science &. Management in Education*, *5*(2), 495–501.

Sommerville, I. (2015). *Software Engineering*. California: Addison-Wesley.

Thakkar, J., Kanda, A., & Deshmukh, S. G. (2008). Interpretive Structural Modeling (ISM) of IT enablers for Indian Manufacturing SMEs. *Information Management & Computer Security*, *16*(2), 113–136. doi:10.1108/09685220810879609

Timperi, O.P. (2004). An Overview of Quality Assurance Practices in Agile Methodologies. T76.650 Seminar in Software Engineering.

Velasquez1, M. & Hester, P.T. (2013). An Analysis of Multi-Criteria Decision Making Methods. *International Journal of Operations Research, 10(2),* 56-66.

Weinberg, G. M. (1992). *Quality Software Management* (Vol. 1). New York: Dorset House Publishing.

*Parita Jain Completed her M.tech (IT) from CDAC Noida. Currently, she is working as Assistant Professor, Department of Computer Science and Engineering at KIET Group of Institutions, UP. She is pursuing Ph.D. from AIIT, Amity University.*

*Arun Sharma completed his Ph.D. Degree from Thapar University, Patiala and M. Tech. (CSE) from Punjabi University, Patiala. Having vast teaching experience of approx. 20 years, currently, he is working as Associate Professor, the Department of IT at Indira Gandhi Delhi Technical University for Women (IGDTUW), Delhi. His research interests include Big Data, Data Mining, Software Engineering, and Soft Computing. He has published more than 60 papers in international journals and conferences.*

*Laxmi Ahuja M.Phil (CS) and Ph.D (CSE). Having vast experience of approx.18 years in academics; currently she is working as Dy. Director(Student Affairs) in Amity Institute of Information Technology. Her areas of interest include Search Engine, Data Mining and Cloud computing; published more than 40 research papers in International and National Journals and Conferences including Inderscience, Springer, IEEE, Elsevier and several others. Session chair in various Conferences. She has organized various Guest Lectures, Workshops and Faculty Development Programme She is actively involved in research activities. She has currently Eight research scholar under her guidance. Have supervised one student for PhD. (Degree awarded by Amity University). She has also published number of Books related to operating system and Information Technology. She is Life Member of Computer Society of India (CSI), Vice chairman of IETE -The Institute of Electronics and Telecommunications Engineers, Senior member of IACSIT and is actively involved in various activities of these societies,she is also member of IEEE .She is a Member of Technical Programme Committee (TPC) in various conferences across globe.*