

A Framework For Software Quality Assurance Using Agile Methodology

Maria Sagheer, Tehreem Zafar, Mehreen Sirshar

Abstract: Agile methodology that uses iterative and incremental approach is widely used in the industry projects as they can satisfy to the change of requirements. Efficient product delivery is achieved by using short iterations. But assuring the quality in any system is essential and imperative part which is very difficult task, which raises a question that: Is quality is assured in the software which is developed using agile methodology? The research paper proposed a solution and summarizes that how the quality is achieved or assure in agile software development using different factors. The major focus of this research paper is to analyse the quality parameter which assure quality in the agile methodology. The research paper includes the brief overview of the parameters which assure quality in agile.

I. INTRODUCTION.

For the successful software engineering, the delivery of high quality software is needed. Mainly the customer satisfaction level is considered as the quality attribute which defines high quality of any system. Traditionally waterfall approach is used for the software development in which system is developed by freezing the requirements. To achieve high quality Agile methods rather than waterfall approach for the software development are adapted by many organizations to compete in the volatile market where the customer requirements are changing at the internet speed, changing organizational infrastructure and need for the short time deliverables. Agile methods that are most popular in use are: Extreme Programming (XP), Scrum, Feature Driven Development and Crystal Method. Agile uses the iterative and incremental approach for the software development due to which the time needed to meet errors is limited and it also emphasize developers to solve a problem as soon as possible. Agile development focuses on the interaction of individuals, customer participation and coping with change. The agile development gives highest priority to the customer needs, regular delivery of software and the quality of the software. But it more focuses on development rather than design and also not applicable for large projects.

II.EFFECT OF AGILE METHODOLOGY ON QUALITY PARAMETERS IN DIFFERENT SCENERIO

Agile software processes, such as extreme programming (XP), Scrum, etc., rely on best practices that are considered to improve software development quality. It can be said that best practices aim to induce software quality assurance (SQA) into the project. The quality assurance activities, in software development are also the backbone of the project.

These activities are also responsible for process quality. The quality of the project is also affected by different parameters such as time constraint, efficiency, reliability etc. The effects of agile on different parameters have been figured out in analysis. The evaluation criteria for software quality assurance using agile methodologies is given in Table I, for comparing effects of different parameters that are discussed in analysis. The effects of agile; its strengths and weaknesses on quality assurance activities, software development process, maintainability, productivity and reusability are discussed below.

2.1) Evaluation of Quality Assurance factors in Agile Methodologies (S. Mehreen&A. Fahim ,2012)

For the improvement in the quality of software products, agile method for software development has been adapted by many organizations to sustain in the market. Evaluating agile methodologies against main quality factors, acts as a tool for studying quality in agile technique. By analysing the quality factors against each methodology it has been notice that very simple and commonly followed methodologies are used. This defined tool introduces a new research era in development of agile technology where as the limitation of this tool is that is not applied in the industry yet.

2.2) Software Quality Assurance in Agile and Waterfall – A Comparative Study (J. Sana & N. Jedaiah, 2008)

Formal software development methods or techniques are linear and static while this technique is not effective in the volatile environment where requirement changes at internet speed. Software quality is achieved in development process through two aspects; first is providing technique for software developed is of high quality while the other one is that in which quality factors is assured in the existing software. To analyse agile quality assurance, agile is compared with waterfall model. In this paper detailed waterfall model is presented which is helpful in checking the quality in agile methodology on the other hand waterfall model is not applicable perfectly in the present scenario therefore quality in agile should have to be compared with the present and mostly used technology.

2.3) Agile Software Assurance (Noura, M. Andrew & B. Gray, 2007).

Agile methodology is presented since 90's in many books, articles journals etc but a little research is conducted upon the quality assurance in agile. The main purpose of this paper is to search for the answer that how quality is assured in agile

- Maria Sagheer, Tehreem Zafar, Mehreen Sirshar
- Department of Software Engineering,
- Fatima Jinnah Women University, the Mall
- Rawalpindi, Pakistan

software development. CMMI gives many standards regarding the modification of agile methods but main reason to introduce CMMI was to provide a standard which will be suitable for all modern iterative methodologies. Using qualitative and quantitative research method data is collected from agile practicers about the common practices followed in agile development. The main advantage of this is that it provides a successful quality assurance model for the agile projects.

2.4) A Quality Management Model for Agile Manufacturing Environment (S. Nookabadi & E. Middle, 2010).

Agility is coping up with the huge change in the way to create a system which is more flexible, responsive and efficient enough to adapt the changing marketing trends. In the agile environment the errors, defects and losses should be least in the process of design creation process. Study on the quality management in the agile methodology must be based on the customer satisfaction and diverse marketing trends. Through the implementation of this model the high quality is ensured in the definition, design, development, sales and services and the chances of the defects will be reduces making it efficient.

2.5) Towards an Understanding of Quality Assurance in Agile Software Development (Juha, R. Kristian & L. Casper, 2008)

The focus of agile development is towards the individuals and the interaction, customer collaboration, development of deliverables and its delivery of the software product. From the view of quality assurance these factors are difficult to measure and the traditionally accepted methods are seems to be lacking in agile development. Many practices are proposed in the paper for testing the agile software development. Two major challenges are described in this paper first is, quality assurance practices over existing agile methods to show enhancements and second is, finding testing practices in agile. More research should have to be focused on session based agile development.

2.6) Handling of Software Quality Defects in Agile Software Development (R. Jorg, 2009)

In the high quality software systems, efficient and effective development of the system is concerned. Refactoring is very important in the development using agile but in one phase all the defects are not removed. Therefore, handling of defects in maintaining software is a major problem. The proposed solution promises to handle the systematic and semi-automatic support for refactoring activities. Quality activities become simpler by recording quality defects and code transformation and the quality of the software increases. This will helpful in maintaining records of the defects and minimizing them.

2.7) Improved Software Quality with Agile Processes (A. Tabinda, 2009)

Due to the changing nature of software development, quality professionals must change with it. Quality is the basic aspect of agile, which is tested by the developers and the customer will have a better quality of the system. As this technique will improve the quality but lessen the participation of quality assurance team. A true agile framework is best then the traditional one because the testing and error fixing is much

easier and quick. Using this approach all the testing is done on the developers end but acceptance and usability testing can be done on users end. Despite of all the advantages, achieving a true agile quality assurance flow is not easy and it requires coordination among stakeholders.

2.8) Agile Practices: An Assessment of Perception of Values of Professionals on Quality (S. Mariana and H. Paulo, 2011).

In the current scenario the agile technique is become very important. This paper presents an analysis of survey report which shows the relationship between use of agile practices and quality of software products. This study suggests that the practitioners should use combination of agile practices and it will improve the understanding of the software development. This research suggests the professional's perception and the working environment and doing the work in the defined time. Delivering high quality software in time or in the defined deadline becomes a challenge and this suggests that the organized set of agile practices should be adopted to achieve better quality product.

2.9) Enhancing Software Quality Using Agile Techniques (M. A. Kashem et al, 2013).

Agile technique produces software faster and enhances the quality. The paper focuses on the quality factors and depicts how these factors enhance the quality of software. An agile software life cycle is drawn which shows the software quality process. Response to the variable requirements, customer satisfaction level and the continuous delivery of the software is the major advantage of agile while its disadvantage is that it is difficult to access the effort (in terms of time and cost) required at the beginning of the life cycle. In short agile technique enhances the flexibility of the software system.

2.10) An Efficient Objective Quality Model for Agile Application Development (J. Ali et al, 2014).

With a large paradigm shift in the software industry, a number of software development methodologies have been proposed. Along with these methods, software quality methods and techniques have also been evolved. In this research paper a quality matrix for the agile development has been shown which will ensure the quality for the product being developed. In this matrix eight quality attributes along their attributes have been mentioned which depict the role of these attributes in all the phases of SDLC. Analyzing the matrix it has been noticed that the most important quality attribute is flexibility while portability and understandability comes afterwards.

2.11) Agile Software Development Quality Assurance (G. Loannis et al, 2007)

The four ways that are mentioned in research paper to increase quality of the software: providing better quality evaluation, more measuring technique, good tools and good series of processes. Waterfall model is an approach in which process of QA (quality assurance) is clearly described. But there are some drawbacks in waterfall therefore; to overcome the major issues in the Waterfall approach, agile methods were developed. To develop software of high quality, the incremental and iterative approach in software development process should be used. Agile strictly follows the iterative approach meaning that all the software parts should be build on regular basis therefore the time needed to handle the

errors are limited and the problem is detected by the developer in earlier stages. A lot of planning is required for this approach but is helpful in getting a high quality software product.

2.12) Modelling Software Maintainability and QA in Agile Environment (U. Priyanka et al, 2014)

Testing software through agile is a new approach that aims to get high quality software which is more maintainable and better result can be achieved. In agile, the problems of maintenance and quality assurance are: deliveries over quality, development over planning, prioritization of scheduling tasks of different programs, divide system in modules and degree of changing requirements. But along with these there are many advantages of agile testing assuring maintainability and quality which are: it improves customer interaction as customer involved at every step, defects can be seen and handled in earlier stages so risks are reduced and flexibility to add or change requirements. So customer satisfaction is achieved at every step and time or cost reduced and on later stages is helpful in maintaining and assures quality of software.

2.13) Do agile Methods Increase Productivity and Quality (R. Gabriela & G. Daniel, 2014)

To deal with software development projects, new methodologies appeared and because of their particular characteristics they fell into two broad categories: Traditional and Agile. In traditional method system design is fully specified and the organization is very large and communication within team is formal. In agile methods design gets improved continuously, regardless the size of organization work is divided into small groups and communication within team is informal. The empirical study is conducted in an academic environment to show effect of traditional and agile methods on productivity and quality. By applying a traditional method, we obtain a more reproducible result, but we could not obtain proof of an improvement in quality. On the other hand, there was evidence about obtaining higher productivity by using Agile methods.

2.14) Impact of Agile methodology on Software Development Process (K.Gaurav & K. Pradeep, 2012)

A group of methods to develop software focusing on iterative and incremental process are called as Agile Methodologies. It is helpful in achieving high software quality in less time, interaction of customers and less written work. The priority of agile methodology is to satisfy customer, welcome to change requirements and deliver working software frequently. The most commonly used agile methodologies are XP, SCRUM, FDD and Crystal method. Various studies depicts that developing software using agile has great effect on quality and the benefits of agile in software development include: handling change of requirements, fault detection, increased performance, iterative and incremental delivery and improvement in quality. But there are also some limitations of Agile which are main focus on development rather than design, do not scale well to large projects and management overhead is increased. Adopting agile technique has a wide effect on the quality and the productivity of the software. And then, both the customer and the team who is linked with development process becomes satisfied with the work.

2.15) Enabling Reusability in Agile Software Development (C. Inderveer et al, 2012)

Agile focuses on the accelerated and less costly software development. Achieving both this technique put somewhat compromise in the quality and will unable to provide the reusability of its software developed parts. In computer engineering as well as in software engineering reusability is the important factor of the source code, which is then used to add more functions to that system having no or some modification. With reusability the productivity of the developers are increased as well as with the increase in reliability and the maintainability of the software. Through the following three ways the reusability is added in the agile development that are: Component based development, Reusable designs and refactoring the design patterns. According to the proposed model, searching will become faster in agile enhancing reusability. Pattern based designs, UML designing and analysing is incorporated. Agile development includes quality factor but is unable to provide reusability of its modules.

2.16) Agile Software Development: It's about feedback and Change (W. Laurie, 2007)

The paper focuses upon the defined processes. A process that can be started and produced the same results every time it is run is called as the defined process. In the context of engineering empirical processes are consists of short parts or chunks. These short cycles are helpful for the better handling of the software having conflict in demands, developed using agile. The three categories of risks that can be used to turn the project towards agile are described by Boehm and Turner is: risk stemming using agile approach, using plan driven approach and general environmental risks. Their work focuses on the experience and characteristics of development techniques.

2.17) Generalized Framework for Agile Software Development Process (S. Bhalerao & M. Ingle, 2009)

AM's (agile methods) have the ability to incorporate variable requirements even though in tough timings. Due to reluctant management these methods can become the failure. Therefore generalized agile development framework process (GADFP) has the ability to cope and change the constraints. The defined framework described testing techniques for assuring quality for the software that are developed that are: test of smoke, mock clients test and exploratory test. It aims to build trust in the practitioners. The advantages of this framework is: provides clear understanding the process of agile development to the developers, these methods are also used in real time life critical systems and it defines practices that over sees the progress and resource utilizations.

2.18) Quality Assurance Activities in Agile (I. Malik et al, 2009)

As agile methodologies are not process comprehensive therefore it is known as light weight method. These methodologies also changes the way of quality assurance activities. Only those activities are documented that are required by the customers or the users. A very vast and tremendous change occurs in the quality assurance activities. Agile focuses on the individuals as it is people oriented development. Customer collaborations are very rapid and done before every iteration this will increase the quality of the

system. The challenges that the agile is facing is that it requires customer iteration is needed after every step but that may cause misunderstanding because of technically unaware. It reduces the need of long technical documentations while on the other side write-up is essential as that it is needed in redefining the quality assurance factors.

2.19) Effective implementation of Agile practices J.E. Veerapaneni et al, 2011)

The Quality Facilitator tasks in different aspects of the agile software development process are: project management, facilitation and deployment, change management, release management, software configuration and management and testing and quality assurance. This paper also shows that an important part in agile software development is traceability. In agile methods traceability helps to maintain the entire information gather, effortless to locate and structured. It is also important for teams to be competent of tracing the information and the decisions that were made during the whole process. The imaginative and speculative framework is given which shows Agile Manifesto based on twelve principles. Agile software development occurs with earlier release and within financial plan by following those principles.

2.20) Agile Methods and Quality Assurance (N. Bodge et al, 2013)

Quality defects are viewed by the techniques that are building upon several research fields such as: Software inquiry, code analysis and inquiry test the software and debugging, etc. There is a process that improves the internal quality of object-oriented software systems in agile software development called refactoring. Refactoring is required to remove quality defects that are introduced earlier and often by unsystematic development. To reduce the cost, effort, and time-to-market of software systems refactoring is used and methods along with tools to support refactoring are becoming more important particularly in agile software development. A company name Siemens had experience that customer connection, continuous re-estimation and continuous reprioritization of scope and features is an intrinsic tool of agile methods that steer to high software quality.

III. ANALYSIS

Implementing agile is a very different approach from the tradition one. But all the agile methods does not fulfils all the quality factors on the other hand one methodology may satisfy the multiple quality assurance factors. The non-functional attributes that is the quality factors can be stated as:

Maintainability:

In agile, locating and fixing of errors in the modular design becomes easy within a specific time. Integration within the modules helps in achieving the maintainability. Feature driven development (FDD) and the crystal methodology are the main techniques to achieve maintainability in agile.

Reliability:

Reliability measures that how consistence the performance of the software or the service for a specific time period. In agile the code transformation and system prototyping will become easy to achieve therefore, refactoring and the system

metaphors are used to achieve the reliability in the agile methodology.

Reusability:

When new functionalities can be incorporated in source code reuse of previous code again is called reusability. Object oriented designs focuses on the principle of reusability. In agile, design the increments of the system and keeping track of them in such a way so that these increments become available and reused in short time.

Testability:

Ease of testing the increments to ensure that the specified function is performed or not is testability. In agile, software is developed in increments which are easy to test. For testing the increments techniques like unit testing, acceptance testing, refactoring and pair programming are used.

Timing Constraint:

In an iterative process it is very important to fulfil all the tasks with the defined timeline. When every increment of the system is completed on time then it will become easy to achieve the time constraint set at the start.

Portability:

In an agile environment nothing is constant; requirements may change which lead to change in the platform and hardware changes therefore, moving software to another environment is portability. In agile, web services designs and distributed computing maximize portability.

Efficiency:

Efficiency is that using fewer resources more functionality is achieved. Pair programming, object oriented designs and common class designs are used to achieve efficiency in agile.

Generalized:

Take specific thing then apply it broadly is generalized. In agile, if the quality of the software increased by using a specific model then it can be applied in the development of every software that follows agile approach/methods.

Scalability:

The main functionalities are developed in the earlier increments and then new functions are incorporated in later stages so, the product needs to be scalable. COTS, Continuous integration of functionalities and on site customer feedback are the techniques which are used to achieve scalability in agile.

Ease of use:

Customer collaboration increases the perception of the design of the system and customer becomes familiar with the interface design. Interface creation is very important in web designs but sometimes problems are created with open source development. Scrum, XP, RUP and JAD (joint application development) methodologies are used to achieve the user familiar and understandable design.

Security:

Low coupling and high cohesion in increments of the system is helpful in achieving the security of the information as well as the rectitude of the software.

Tool support:

Tools are used for system analysis, analysis of code, testing the software and running. In agile after each iteration the software is tested using some specific tools to achieve high quality.

Case study:

The analysis related to projects or systems etc is case study. To analyse the effect of agile methods on quality different methodologies like comparative studies, empirical studies and surveys are conducted.

Productivity:

The productivity of the developer increases by using reusability during agile software development. Errors are detected at early stage so the cost to solve the problem is much more less than the cost that is used for the error detection after the development of whole software/system.

Cost effectiveness:

Delivering the prioritised requirements in the smaller and earlier increments and iterations is helpful to achieve the cost effectiveness.

Correctness:

Agile technology requires customer involvement therefore onsite customer feedback provides correctness in requirement and thus correct functionality of the system is achieved.

Flexibility:

Flexibility is the ability of undergoing the change without altering the overall system. Agile is a very flexible technique as it can handle change anytime by using the iterative and incremental approach.

Robustness:

By following the development standards robustness can be achieved but it cannot be explicitly defined.

Compatibility:

In open source agile methodologies the platform independence is achieved. By applying the object oriented design in local practices, compatibility is also achieved.

Performance:

The performance is related to the efficiency of the system. As agile is flexible so to remain efficient different performance testing is applied to check the performance of the system and then increase the performance by applying changes accordingly.

The maintainability, reusability, efficiency and feasibility etc. are the parameters that effect software quality and by analyzing them the quality assurance can be achieved. To explore and integrate risk factors in Agile Software Development scientifically and find the decisive accomplishment factors of the agile software development process can be done as the future work.

REFERENCES

- [1] H. Amran, M. A. Kashem and S. Sahelee (2013, March 01) Enhancing Software Quality Using Agile Techniques (Volume 10)
- [2] Juha, R. Kristian and L. Casper (2008, June 10) Toward an Understanding of Quality Assurance in Agile Software Development (volume 2)
- [3] M. Usman, M. Haseeb and J. Ali (2014, January) An Efficient Objective Quality Model for Agile Application Development (Volume 85)
- [4] S. Mehreen and A. Fahim (2012, February) A Quality Management Model for Agile Manufacturing Environment (volume 2)
- [5] Noura, G. M. Andrew and W. B. Gray (2007) Agile Software Assurance
- [6] K. J. Sana and M. N. Jedaiah (2008, December) Software Quality and Assurance in Waterfall model and Agile techniques - A Comparative Study (Volume 7)
- [7] A. Tabinda (2009, January) Improved Software Quality with Agile Processes
- [8] R. Jorg. (2009, March) Handling of Software Quality Defects in Agile Software Development
- [9] S. Mariana and B. H. Paulo (2011, December) Agile Practices: An Assessment of perception of values of Professionals on Quality Criteria
- [10] A.S. Nookabadi and J.E. Middle. (2010, June) A Quality Management Model for Agile Manufacturing Environment
- [11] I. Malik and Z.A.Waqar (2009, September) Quality Assurance Activities in Agile
- [12] N. Bodge, N. Kauh and G.M. Nasira (2013) Agile Methods and Quality assurance (Volume 3)
- [13] J.E. Veerapaneni, K.R. Nageswara (2011, March) Effective Implementation of Agile Practices (Volume 2)
- [14] S. Sukhpal and C. Inderveer (2012, July) Enabling Reusability in Agile Software Development (Volume 50)

IV. CONCLUSION

The paper discusses the effect of various factors and parameters on software quality and the assertion of software quality in agile environment. Agile methods provide an competent software product by releasing it before time to the running software surroundings, simplifying cooperation and getting higher the customer satisfactions. In Agile, there is a constant communication with the customer, so according to the customer requirement, the new features are added to satisfy the customer which reduces the time and cost which auxiliary help in modeling the quality assurance of software.

[15] W. Laurie (2007, June) Agile Software Development: It's about Feedback and Change

[16] S. Bhalerao and M. Ingle (2009, November) Generalized Framework for Agile Software Development Process (Volume2)

[17] K.Gaurav and B.K. Pradeep (2012, August 4) Impact of Agile methodology on Software Development Process (Volume 2)

[18] R. Gabriela and G. Daniel (2014, April 20) Do Agile Methods Increase Productivity and Quality (Volume 3)

[19] U. Priyanka, S. Abhishek and G. Naveen (2014) Modeling Software Maintainability and Quality Assurance in Agile Environment (Volume 7) [online]

[20] S.G. Loannis and S. Panagiotis (2007) Agile Software Development Quality Assurance

TABLE 1 EVALUATION CRITERIA FOR QUALITY ASSURANCE USING AGILE METHODOLOGIES

Evaluation Parameters	Meaning	Possible Values
Maintainability	System is maintainable or not.	Yes, No
Reliability	System is working or not till the time line is given.	Yes, No
Reusability	Proposed technique is reusable or not.	Yes, No
Testability	Proposed design tested or not.	Yes, No
Timing constraint	Quality can be specified through timing.	Yes, No
Portability	Software can run on different platform.	Yes, No
Efficiency	System is efficient in terms of hardware resources.	Yes, No
Generalized	Design is generalized enough to be customized or not.	Yes, No
Scalability	New functionality can be added to the system.	Yes, No
Ease of use	Software is easy to learn or use for the users.	Yes, No
Tool support	Tools are available for the proposed model.	Yes, No
Case study	Examples can use to support the methodology.	Yes, No
Security	The proposed technique is able to detect and correct errors.	Yes, No
Cost effectiveness	The system provides proper functionality within the budget.	Yes, No
Productivity	The proposed technique increases productivity or not.	Yes, No
Correctness	System is working according to the specification.	Yes, No
Flexibility	System is able to accept change.	Yes, No
Robustness	System is able to correct errors that are not specified.	Yes, No
Compatibility	System elements can combine with other elements or not.	Yes, No

TABLE 2 ANALYSIS OF PARAMETERS FOR QUALITY ANALYSIS USING AGILE METHODOLOGY

S #	Techniques	Maintainability	Reusability	Reliability	Testability	Timing Constraint	Portability	Efficiency	Generalized	Scalability	Ease of use
1	Malik and Waqar,2009	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2	Bodge et al, 2013	Yes	Yes	No	Yes	Yes	No	No	Yes	No	Yes
3	Veerapaneni and Nageswara, 2011	Yes	No	No	Yes	No	No	No	Yes	No	Yes
4	Sukhpal and Inderveer, 2012	No	Yes	No	Yes	No	Yes	No	No	Yes	Yes
5	Laurie Williams, 2007	No	Yes	No	No	No	No	Yes	No	Yes	No
6	Bhalerao and Ingle, 2009	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
7	Gaurav and Pradeep, 2012	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No

8	Gabreela and Daniel, 2014	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes
9	Priyanka et al, 2014	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
10	Al-Jidaiah and Khalaf, 2008	No	Yes	Yes	Yes	Yes	No	Yes	No	No	No
11	H. Amran et al, 2013	Yes	Yes	Rhapsody as a support tool	Yes, it is model driven approach	Yes, it is executable	Yes	Yes	No	Yes	Yes
12	Juha et al, 2008	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes
13	M. Usman et al, 2014	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes Printer paper path	Yes

TABLE 3 ANALYSIS OF PARAMETERS FOR QUALITY ANALYSIS USING AGILE METHODOLOGY

S#	Techniques	Tool support	Security	Case study	Cost Effectiveness	Productivity	Correctness	Flexibility	Robustness
1	Malik and Waqar, 2009	No	Yes	Yes	No	Yes	Yes	Yes	No
2	Bodge et al, 2013	No	No	Yes,	No	Yes	Yes	Yes	Yes
3	Veerapaneni and Nageswara, 2011	No	No	No	Yes	No	No	Yes	Yes
4	Sukhpal and Inderveer, 2012	No	Yes	No	Yes	No	Yes	Yes	Yes
5	LaurieWilliams, 2007	No	No	No	Yes	No	Yes	Yes	No
6	Bhalerao and Ingle, 2009	No	No	Yes	No	Yes	No	Yes	No
7	Gaurav and Pradeep, 2012	No	No	Yes	Yes	Yes	Yes	Yes	Yes
8	Gabreela and Daniel, 2014	Yes	No	Yes	Yes	Yes	Yes	Yes	No
9	Priyanka et al, 2014	No	No	Yes	Yes	Yes	Yes	Yes	Yes
10	Al-Jidaiah and Khalaf, 2008	No	No	Yes	No	No	Yes	Yes	No
11	H. Amran et al, 2013	No	No	No	No	Yes	Yes	Yes	No
12	Juha et al, 2008	No	No	Yes	Yes	Yes	No	No	No
13	M. Usman et al, 2014	No	No	No	No	No	Yes	Yes	Yes
14	Mehreen and Fahim, 2012	No	No	No	Yes	Yes	No	No	Yes
15	Noura et al, 2007	Yes	Yes	No	No	Yes	Yes	Yes	No
16	Sana and Jedaiah, 2008	No	No	No	No	Yes	Yes	No	No
17	Tabinda, 2009	Yes	No	Yes	No	No	Yes	No	No
18	Jorg, 2009	Yes	No	Yes	No	No	No	Yes	No
19	Mariana and Paulo, 2011	Yes	No	Yes	No	No	Yes	No	No
20	Nookabadi and Middle, 2010	No	No	Yes	Yes	Yes	No	Yes	No