International Journal of Computers and Informatics (IJCI)

Vol. (2), No. (2)

المجلة الدولية للحاسبات والمعلوماتية

الإصدار (2)، العدد (2)

June 2023

"Approaches and Models for Evaluating the Agility Degree of Agile Software Development Methods"

Hesham Ali Saad
M.Sc. of Information Systems, Baghdad University, Iraq
halialisaad@yahoo.com

Mohamed Abedulrashid Ghali
Ph.D. of Computer Science, Baghdad University, Iraq
mohamed.ghali@gmail.com

Abstract:

Agile Software Development (ASD) is the umbrella for several other popular methods such as Scrum, XP, Feature Driven Development, DSDM, Crystal, and others. ASD methods are based on developing software products using short iterations, each iteration is like a short project. Also, ASD methods use "inspect and adapt" practices to adjust the project plan. This paper presents a review of approaches, models, and frameworks for evaluating the agility degree of ASD methods. Each approach, model, or framework is based on a set of practices, features, or events that support agility. This review can help managers of software projects to choose the appropriate ASD method for their projects.

Keywords:

Agility Evaluation, Agile Methods, Agile Approaches, Software Projects.

1- Introduction

Agile software development is an approach based on early delivery, continuous adaptation, adaptive planning, and evolutionary development, with flexible and rapid response to variables [1]. Therefore, requirements and solutions are developed according to these features that are characterized by the collaborative effort of self-organized and cross-functional teams and their customers [2]. Agile is a method for developing software products using short iterations, each iteration is like a short project. Uses “inspect and adapt” practices to adjust the project plan. Agile is the umbrella for several other popular methods such as Scrum, XP, Feature Driven Development, DSDM, Crystal, and others.

An evaluation of different agile methods helps managers of software projects who need to introduce agility into their organization by providing a thorough knowledge of different features and aspects of the agile methods. By evaluating different agile methods, the manager can easily choose the best one that suits his project requirements and organizational characteristics. Different comparison techniques and approaches to compare agile methods were introduced in recent years by researchers. Agile approaches are often evaluated by comparing two agile methods: XP and Scrum. This is due to several reasons, the most important of which are the presence and citation of these methods in literature and the increasing number of developmental teams using one of these methods.

This manuscript has the following structure. In section 2, the main characteristics of agile methods and Agile Manifesto Principles are presented. A brief description of different agile methods is also included. The multiple techniques and approaches in this work to evaluate and compare agile methods are explained in section 3. Discussion and results are presented in section 4. Conclusions and future work are discussed in section 5.

2- Literature Review:

Agile Manifesto in 2001 defines the following values, which are the cornerstones of the agile methods [3]:

- Individuals and interactions over processes and tools.
- Working on software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

The principles behind the Agile Manifesto [4]: the highest priority is to satisfy the customer through early and continuous delivery of valuable software. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. Business people and developers must work together daily throughout the project. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. Working software is the primary measure of progress. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. Continuous attention to technical excellence and good design enhances agility. Simplicity-the art of maximizing the amount of work not done is essential. The best architecture, requirements, and design emerge from self-organizing teams. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

These principles have two main objectives: to promote a better understanding of agile methods and to guide the project teams to determine if they are in fact using an agile method.

Extreme programming (XP): it is a lightweight, more flexible, and low-risk disciplined approach for software development with the ability to manage vague or rapidly changing requirements [12]. It is considered more suitable for small and medium-sized teams [13]. XP believes in five values that are simplicity, courage, communication, feedback, and respect. XP is a set of five values, principles, and twelve practices that are applied in a disciplined way [14]. The whole development process consists of six phases: Exploration phase, Planning phase, Iteration to release phase, Productionizing phase, Maintenance phase, and Death phase [16].

Scrum is an agile software development method that allows for the delivery of the highest business value in the shortest time, scrum is considered a managerial framework for incremental product development using cross-functional, self-organized teams with less than 10 members. The iteration of scrum is called sprint and had a time-bounded from 2 weeks to 6 weeks, and this sprint is suitable for distributed teams of project initialization. Functionality is defined before a sprint begins. Scrum framework consists of roles, ceremonies, and artifacts. Scrum is the best if the requirements frequently change.

Approaches, frameworks, and measurement methods have been proposed for assessing agility degree in companies that use agile methods like Software Engineering Body of Knowledge Approach (SWEBOK), Four Dimension Analytical Tool (4-DAT), Objectives, Principles, Practices Approach (OPP) and Objectives, Principles, Strategies Approach (OPS).

M. N. Huda et al. [7] in 2011, referring that Joao M. Fernandes and Mauro Almeida followed the classification and comparison technique for agile methods proposed in 2010. This approach adopts qualitative analyses which are complemented with tool support that uses a quantitative parametric scale. They conducted a comparison between two agile software development methods (XP and Scrum) based on the attributes related to four IEEE's Software Engineering Body of Knowledge (SWEBOK) Knowledge Areas (KAs), the set of attributes that considered are software requirements, software construction, software testing, and software engineering management. Each attribute has sub-attributes. They identified the important practices of these two agile methods and compared them against each of these sub-attributes. The classification criteria adopted in the comparative analysis are Not Satisfied (NS), Partially Satisfied (PS), and Adequately Satisfied (AS).

Authors in [8] presented that there are some studies that compare two specific agile methods, such as Scrum vs. XP (Fernandes and Almeida, 2010). In general, the main aim of these studies is not to find the pros and cons of agile methods; they have tried to increase the knowledge of the applicability of the methods.

Authors in [9] proposed a technique for comparing and evaluating agile methods based on a set of relevant features and attributes. These features fall under the four Knowledge Areas (KAs) of SWEBOK. They compare XP and Scrum based on this technique.

Kumar and Henderson [5] developed a four-dimensional analytical tool to measure the degree of agility, tested and published it in A. Qumer, B. Henderson-Sellers, Comparative Evaluation of XP and Scrum using the 4D Analytical Tool (4-DAT), Proceedings of the European and Mediterranean Conference on Information Systems 2006 (EMCIS2006). It is based on the following four dimensions: method scope, agility characterization, agile values characterization and software process. This tool

has been included in the Agile Software Solution Framework (ASSF) as a tool for evaluation and analysis. This tool is characterized as an extension for expansion where the dimensions or elements can be added or removed from 4-DAT.

Based on Joao M. Fernandes and Mauro Almeida developed the classification and comparison technique for agile methods proposed in 2010. The Agile Software Solution Framework (ASSF), a complete framework to assist in the assessment of an enterprise necessary agility degree and to identify the appropriate way of introducing agility into the organization is described in [5]. The major element of this framework is the Agile Toolkit, which provides an analytic tool that allows the comparison of agile methods (4-DAT). The 4-DAT approach [6] examines software methods from four perspectives: method scope, agility characterization, characterization of agile values, and software process characterization. These perspectives were defined based on different studies and distilled from key aspects of agile methods. Additionally, the perspectives presented in [6] are partly qualitative and partly quantitative [20]

Mz Nafchi et al. [21] "On the Current Agile Assessment Methods and Approaches," In 2014, mention that Qumer et al. [6] proposed a four-dimensional framework for evaluating agile methods. They provide a specific definition of agile methods based on five factors: flexibility, speed, leanness, learning, and responsiveness. These four dimensions are as follows: method scope (such as project size, team size, development and coding style, technology environment, physical environment, business culture, and abstraction mechanism), agility characterization is based on the defined definition of agile, and features, characterization of agile values is used to check the existence of agile values in agile methods, and the fourth dimension is software process characterization. This assessment model has been introduced as the core of an agile adoption framework, called “Agile Adoption and Improvement

Model" (AAIM) [18]. This framework, like SAMI (Sidky agile measurement index), follows the CMMI (Capability Maturity Model Integration) approach.

Mz Nafchi et al. [21] said that Soundararajan et al. proposed a framework to assess the "goodness" of agile methods under the name of OPP (objectives principles and practices). This framework assesses an agile method based on its adequacy, the capability of the organization to apply this method, and the effectiveness of the method in terms of meeting the expected outcomes. Based on the agile manifesto and agile values they came up with five objectives and they mapped the objectives with nine agile principles, and finally, binding 27 agile practices to the principles was done.

Authors in [21] said Soundararajan et al. proposed a framework: Objectives-Principles-Strategies framework (OPS) [22][25]. The OPS is inspired from CMMI but the authors state it is „a primary disadvantage of these frameworks that a set of practices is "forced" on an organization at defined levels, which compromises the flexibility offered by agile methods." Therefore they „advocate the need for a more comprehensive agile assessment process that assesses the 4P's (people, process, project and product characteristics) of organizations adopting agile methods." and develop an approach that helps the organization identify the supportive environment for the implementation of agile methods as well as the extent Effective implementation of the technique in achieving its objectives[21].

3- Evaluation Approaches:

Several studies have been conducted regarding agile assessment, agility measurement, and the goodness of agile methods. The most important ones are as follows.

3-1 SWEBOK. Approach:

M. Fernandes, Mauro Almeida [10] "Classification and Comparison of Agile Methods"2010. Present a technique to compare and classify agile methods, using as criteria a set of selected attributes. This set includes attributes related to four Software Engineering Body of Knowledge (SWEBOK) Knowledge Areas (KAs) and to the agile principles defined in the Agile Manifesto. The attributes and sub-attributes chosen for their study were selected to assess, with respect to each agile method, the coverage degree to four Software Engineering Body of Knowledge (SWEBOK) Knowledge Areas (KAs) and the agility degree. They selected four of the eleven knowledge areas (KAs) defined in the SWEBOK: Software Requirements, Software Construction, Software Testing, and Software Engineering Management.

The published version of SWEBOK V3 has the following fifteen knowledge areas (KAs) within the field of software engineering [11]: Software Requirements, Software Design, Software Construction, Software Testing, Software Maintenance, Software Configuration Management, Software Engineering Management, Software Engineering Process, Software Engineering Models And Methods, Software Quality, Software Engineering Professional Practice, Software Engineering Economics, Computing Foundations, Mathematical Foundations, and Engineering Foundations.

Based on twelve XP practices (planning game, metaphor, pair programming, sustainable pace, on-site customer, testing, coding standards, refactoring, continuous integration, small releases, simple design, collective ownership) and nine proposed scrum practices (Product Backlog, Effort Estimation, Sprint, Daily Meeting, Sprint Planning Meeting, Sprint Backlog, Sprint Review Meeting, Sprint Retrospective, Sprint Burn Down Chart) the four Software Engineering Body of Knowledge (SWEBOK) Knowledge Areas (KAs) attributes chosen for study were selected to assess, with respect to each agile method practices, the coverage degree to four

(SWEBOK) KAs and the agility degree. The four selected (KAs) defined in the SWEBOK are:

1. Software Requirements: Identifies how the method addresses the requirements analysis in a software project, particularly regarding the following sub-attributes: Software requirements fundamentals, requirements analysis, requirements process, requirements validation, and requirements elicitation.

2. Software Construction: Identifies how the method deals with the software implementation, particularly regarding the following sub-attributes: Minimizing Complexity, Construction for Verification, Response Capability to Unexpected Changes, and Standards in Construction.

3. Software Testing: Identifies how the method validates the implemented features and the approach adopted for testing, particularly regarding the following sub-attributes: Software Testing Fundamentals, Test Related Measures, Test Levels, and Test Process.

4. Software Engineering Management: Identifies how the method addresses the project management, particularly regarding the following sub-attributes: Initiation and Scope Definition, Closure, Software Project Planning, Review and Evaluation, and Software Project Enactment.

All the above have sets of attributes and sub-attributes used for comparing agile methods. The challenge in quantifying the coverage of a given sub-attributes or principle, by a set of practices proposed by each analyzing method, has led to the choice of a qualitative classification system.

Therefore, they use three criteria for adopting in the comparative analysis which are Not Satisfied (NS), Partially Satisfied (PS), and Adequately Satisfied (AS). The aggregated table (1) shows the results of analysis performed under the attributes and sub-attributes for XP and Scrum.

Table (1): Results of analysis performed under the attributes and sub-attributes for XP and Scrum.

Attributes	**XP**			**Scrum**		
	NS	PS	AS	NS	PS	AS
Software Requirements Sub-attribute	**0%**	**40%**	**60%**	**20%**	**20%**	**60%**
Software Requirements Fundamentals		x				x
Requirements Process			x			x
Requirements Elicitation			x			x
Requirements Analysis		x			x	
Requirements Validation			x	x		
Construction of Software Sub-attribute	0%	25%	75%	100%	0%	0%
Minimizing Complexity			x	x		
Response to Changes			x	x		
Constructing for Verification			x	x		
Standards in Construction		x		x		

3-2 (4-D) Framework

Based on Qumer and Henderson- Sellers [17, 18] offer the following definition for the agility of any entity: ‘‘Agility is a persistent behavior or ability of a sensitive entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, uses economical, simple and quality instruments in a dynamic environment and applies updated prior knowledge and experience to learn from the internal and external environment.’’

Based on the agile definition, Qumar & Henderson developed a four-dimensional 4-DAT frame to evaluate the degree of agility for the agile method which is: method scope characterization, Agility characterization, agile value characterization, and software process characterization as shown in Table (2). The second dimension is the only quantitative dimension and is used to check the existence of agility in the agile method at both a process level and method practice level. This dimension contains a set of features (Flexibility, Speed, Leanness, Learning, and Responsiveness) derived from the agility definition proposed by Qumar & Henderson [18]. Based on this they derive a new agile method definition:

‘‘A software development method is said to be an agile software development method when a method is people-focused, communications-oriented, flexible[FY] (ready to adapt to expected or unexpected change at any time), speedy[SD] (encourages rapid and iterative development of the product in small releases), Lean[LS] (focuses on shortening timeframe and cost and on improved quality), responsive[RS] (reacts appropriately to expected and unexpected changes), and learning[LG] (focuses on improvement during and after product development)’’[6].

Table (2): 4-DAT dimensions (derived from [17])

Dimension1: Scope	1. Project Size 2. Team Size 3. Development Style 4. Code Style 5. Technology Environment Responsiveness 6. Physical Environment 7. Business Culture 8. Abstraction Mechanism
Dimension2: Features	1. Flexibility 2. Speed 3. Leanness 4. Learning 5. Responsiveness
Dimension 3: Agile values	1. Individuals & and interactions over processes and tools 2. Working software over comprehensive documentation. 3. Customer collaboration over contract negotiation 4. Responding to change over following a plan 5. Keeping the process agile 6. Keeping the process cost-effective
Dimension4: Process	1. Development Process 2. Project Management Process 3. Software Configuration Control Process / Support Process 4. Process Management Process

Table (3): Degree of agility in XP, Scrum

Process & Practices	**XP**	**Scrum**
Phases	21/30 = 0.70	9/15 = 0.60
Rank	**1**	**2**
Practices	44/60 = 0.73	28/35= 0.80
Rank	**2**	**1**

Table (3) shows the overall degree of agility for both phases and practices of XP and Scrum. These assessments of the degree of agility permit a separate ranking and visual comparison for both a process-based viewpoint and a practice-based viewpoint (Figure 1)

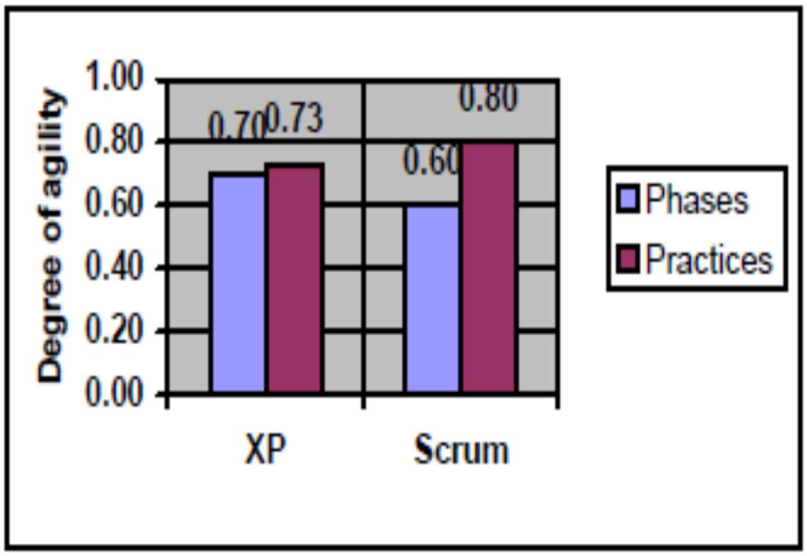

Figure 1: Degree of agility – a comparison of Phases and Practices for XP and Scrum (after [18])

The evaluation can be applied at various levels of granularity. Most agile methods favor discrimination between a high-level process or "phase" level and a lower-level "best practices" level. Dimension 2 is the only one of the dimensions that can be assessed quantitatively. Details of the algorithms Proposed are found in [17] and their application to two exemplary agile methods (XP and Scrum) in [18]. They found that, while XP was evaluated as being more agile at the phase level, Scrum showed more agility at the practice level as shown in (Figure 2) [19].

International Journal of Computers and Informatics (IJCI)

Vol. (2), No. (2)

المجلة الدولية للحاسبات والمعلوماتية

الإصدار (2)، العدد (2)

June 2023

3-3 OPP Framework

Soundararajan et al. developed the Objectives, Principles, and Practices (OPP) framework in Figure (2) to evaluate the "goodness" of a given agile method from three perspectives: adequacy (can the method meet the declared objective), the capability of the organization to apply this method which is dependent on its people, process, and project characteristics, and its effectiveness in terms of meeting the expected outcomes that dependent on process and product characteristics. Depending on agile manifesto and agile values they came up with objectives and mapped them with agile principles, and finally, linked agile practices to the principles.

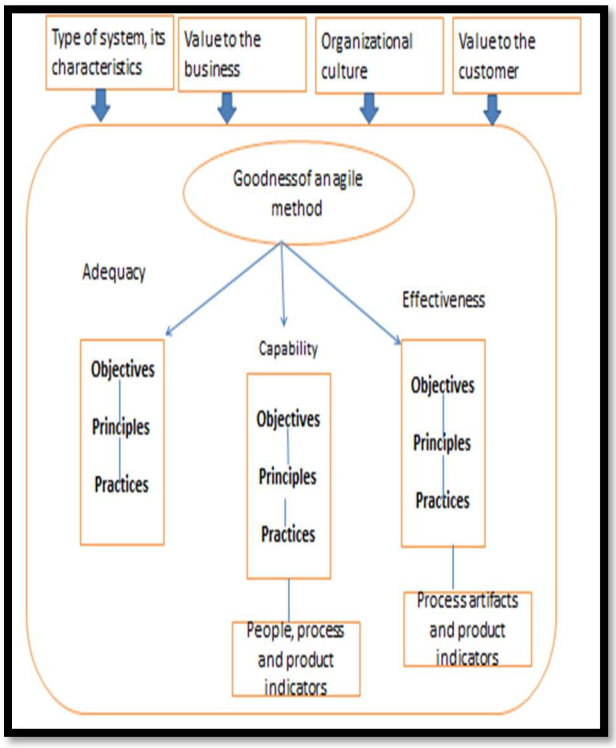

Figure 2. The Objectives, Principles, and Practices of the OPP Framework.

The OPP framework identifies:

1. Objectives of agile philosophy.
2. Principles that support the objectives.
3. Practices that reflect principles.
4. Linkages between the objectives, principles, and practices to assess the adequacy, capability, and effectiveness.
5. Indicators to assess the characteristics of the people, process, project, and product. These indicators are required for the assessment of capability and effectiveness. [22]

3-4 OPS Framework

OPS framework stands for Objectives, Principles, and Strategies. OPS framework is a hierarchical framework proposed by Soundararajan et al to assess the 'goodness' (adequacy, capability, and effectiveness) of agile methods, OPS identifies the following: five objectives at first level that reflect agile philosophy (Human-centric, Value–driven, Minimal Waste, Maximal Adaptability, Continuous innovation and learning);at the second level nine principles that supporting the achievement of previous objectives which are: (Frequent delivery of working software, Technical Excellence, Simplicity, Empowering teams of Motivated Individuals, Constant development Pace, Accommodating Change, Continual stakeholder communication, and collaboration, Frequent Reflection and Improvement, Striving for Customer Satisfaction) and at the third level 17 strategies that help to implement those principles which are:(Iterative Progression, Incremental Development, Short Delivery Cycles, Evolutionary Requirements, Continuous Feedback, Refactoring, Test First Development, Self-Managing Teams, Continuous Integration, Constant Velocity, Minimal Documentation, High-bandwidth communication, Retrospection,

Client-driven iterations, Appropriate distribution of expertise, Configuration Management, Adherence to Standards) [24]. The structure of the OPS Framework is illustrated in Figure 3. The Framework includes linkages that signify definitive relationships between identified objectives, principles, and strategies.

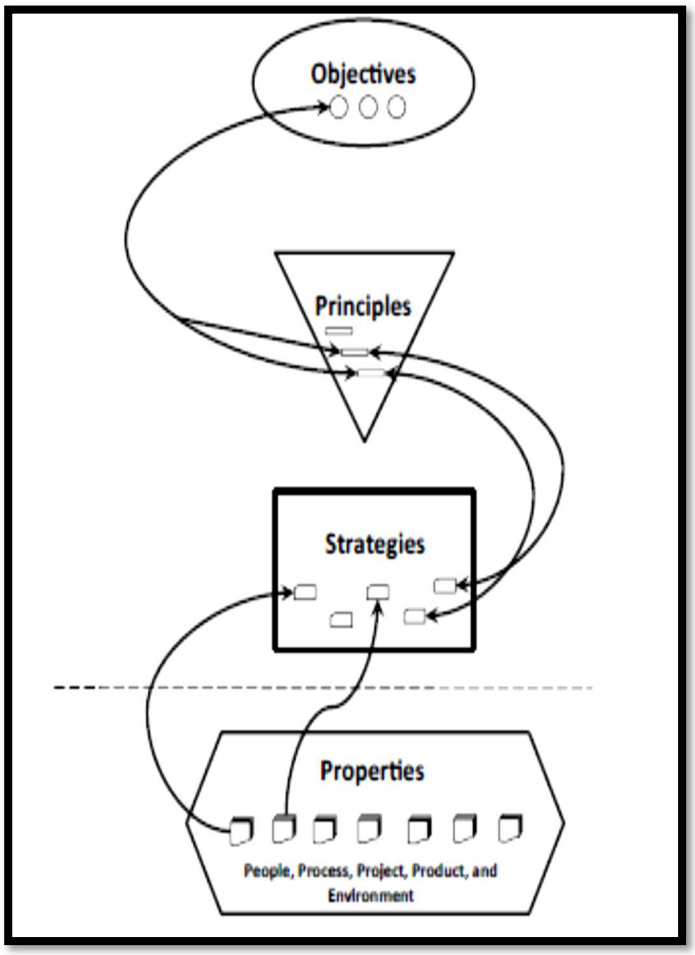

Figure 3. Structure of the OPS Framework [15]

Most assessment approaches focused firstly on product metrics alone which are insufficient for assessing agile methods. Software Engineering involves people, processes, projects, and products (the 4 P's) [23]. Hence, metrics used in the assessment of agile methods should incorporate characteristics of the 4Ps. The OPS Framework presented is designed to incorporate the 4Ps.

4- Discussion and Results

Regarding the 4-D framework, although the definition of agile and its key attributes are compatible with the reality of agile, this framework reduces the flexibility that is needed to be agile [26]. This is primarily because this model measures the agility of a company by analyzing the adoption of a set of practices, forcing companies to accept pre-defined sets of agile practices reduces the flexibility promised by agile. Furthermore, the defined agility level may not be "in sync" with organizational objectives in a company [22]. The overall degree of agility for both phases and practices of XP and Scrum shows that XP is better than Scrum at phases while Scrum is better than XP at the practice level. These assessments of the degree of agility permit a separate ranking and visual comparison for both a process-based viewpoint and a practice-based viewpoint. We have analyzed and compared two agile software development methods (XP and Scrum) by using 4-DAT, a framework-based assessment tool, described in detail by Qumer and Henderson-Sellers (2006). This analysis used both a qualitative and a quantitative approach to evaluate the agile methods at both the phase level and the practice level. Based on this analysis, we may rank the methods – here, XP has more agile phases but less agile practices than Scrum.

Kumar and Henderson developed a four-dimensional analytical tool to measure the degree of agility, tested and published it in A. Qumer, B. Henderson-Sellers, Comparative Evaluation of XP and Scrum using the 4D Analytical Tool (4-DAT), Proceedings of the European and Mediterranean Conference on Information Systems 2006 (EMCIS2006). It is based on the following four dimensions: method scope, agility characterization, agile values characterization, and software process. This tool has been included in the Agile Software Solution Framework (ASSF) as a tool for evaluation and analysis. This tool is characterized as an extension for expansion where the dimensions or elements can be added or removed from 4-DAT [5]. We can add some other features like a cycle, planning, artifacts, stakeholder involvement, communication style, requirement elicitation, documentation, design flexibility, development order defined by, code ownership, changes during iteration, feedback, and validation techniques [27]

The OPP framework defines objectives, principles, practices, indicators, and linkages between them to help the evaluation procedure. OOP framework is useful for indicating the goodness of each agile methodology compared to other methods. However, it cannot be helpful to measure the progress of agile transformation because most often transitioning to agile does not mean the adoption of a specific agile method. Indeed, most often software companies try to adapt to some agile practices rather than a whole particular agile method [21]. Evaluation of "GOODNESS":

A. Evaluating Adequacy

It is independent of an organization so adequacy can be assessed by standalone agile methods such as XP and Scrum with respect to the agile values and principles each espouses.

B. Evaluating Capability

People, process, and project indicators that denote the characteristics of the environment are used to assess the capability of the organization.

C Evaluating Effectiveness

The effectiveness of a method lies in the ability of an organization to produce the expected results. Assessing the effectiveness also involves a bottom-up traversal from process artifacts and product quality indicators to the objectives.

Regarding the SWEBOK Approach, under the attribute Software Requirements, figure 1(a) demonstrates that 3 out of 5 of the selected sub-attributes are adequately satisfied by XP and that 2 of those attributes are partially satisfied. Figure 1(b) represents the fact that 3 of those attributes are adequately satisfied, 1 is partially satisfied, and the last one is not satisfied by Scrum. The results of XP derive mainly from the close contact between the technical team and the client. The importance given to this relationship is reflected in the fact that XP addresses all sub-attributes of the Software Requirements KA. Regarding the attribute Software Construction, the analysis concludes that XP adequately satisfies 3 out of the 4 selected sub-attributes and partially satisfies 1 of those same sub-attributes, while Scrum does not satisfy any of the selected sub-attributes. The fact that Scrum does not explicitly suggest any practice-oriented implementation, delegating the choice of which practices to apply to the cross-functional teams, shows a clear difference between XP and Scrum under the attribute Software Construction.

XP puts a big value on testing. Based on test-driven development, XP places tests as the foundations of software development. The results of the analysis under the attribute Software Testing reflect this aspect of XP and the importance it places on

creating and running tests in a software project. XP analysis under the attribute Software Testing reveals that XP adequately satisfies all sub-attributes of this KA. Scrum does not fulfill any of the sub-characteristics of Software Testing.

Under the attribute Software Engineering Management, XP adequately satisfies 4 of the 5 selected sub-attributes and only the sub-attribute Review and Evaluation was not satisfied. When compared to the same set of sub-attributes Scrum reveals itself more complete than XP, adequately satisfying 4 of the selected sub-attributes and partially satisfying one of those same sub-attributes. Furthermore, the analysis shows that Scrum tackles this part of a software project in a much more detailed and precise way when compared to XP.

From the study, a few investigations concentrated on agility assessment. These investigations for the most part have concentrated on looking at organizations in terms of agility, considering specific agility levels, and the goodness of Agile methodology adopted by software companies. Be that as it may, the essential impediment of the proposed strategies, assessment, and appraisal procedures is their constrained scope and application [22].

5- Conclusion:

Some of the assessment methods have relied on agile practices and some others on agility levels they have defined. Generally, it seems that agility assessment is not a straightforward process.

Reviewing the above methods and tools reveals that there is a gap in agility assessment models. The existing approaches suffer from some serious drawbacks. Obviously, none of the above methods is recommended for agility assessment. However, each of them can be used only for the real purpose which has been

considered when proposing that method. For instance, OOP is very good for assessing the goodness of any specific agile method [21].

While the prevalence of agile methods is increasing in software companies, there is still a gap in assessing the agility degree of software companies. This review paper showed that there are a few agility assessment methods to assess the agility degree of software companies. However, they are subject to some serious challenges. In this paper, the four most important assessment methods have been described and their positions in agility assessment have been explained. In general, there is no perfect assessment model that is both compatible with agile and comprehensive for assessing the agility degree of software companies or teams who are adopting agile methods or practices. Considering the strengths and weaknesses of the current assessment model, potential future work is providing a better assessment model that does not have the drawbacks of the existing ones. Clearly, such a model is better to focus on the agile practices and their values in achieving agility in companies or teams. For future work, the OPPS (Objectives, Principles, Practices, and Strategies) approach can be developed and examined by companies to evaluate it, also the other eleven KAs of SWEBOK that did not do in the selected study can be completed later.

References

[1] "What is Agile Software Development?". *Agile Alliance. 8 June 2013*. Retrieved 4 April 2015.

[2] Collier, Ken W. (2011). Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing. Pearson Education. pp. 121 ff. ISBN 9780321669544. What is a self-organizing team?I. A. ElSayed, Z. Ezz, and E. Nasr, “Goal modeling techniques in requirements engineering: A systematic literature review,” J. Comput. Sci., vol. 13, no. 9, pp. 430–439, 2017.

[3] Manifesto for Agile Software Development (2001), http://www.agilemanifesto.org.

[4] Beck, K., Beedle, M., et al., Principles Behind the Agile Manifesto. [Online] [Cited: June 6, 2007] HTTP:// agilemanifesto.org/principles.html.

[5] Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption, and improvement of agile methods in practice. Journal of Systems and Software, 81(11), 1899–1919.2008.

[6] A. Qumer, B. Henderson-Sellers. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Information and Software Technology 50 (2008) 280–295, 2008.

[7] M. N. Huda, S. Sciences, S. Sciences, and A. Software, “Comparison between Adaptive Software Development and Feature Driven Development,” pp. 363–367, 2011.

[8] T. J. Gandomani and H. Zulzalil, “Towards Comprehensive and disciplined Change Management Strategy in Agile Transformation Process,” no. June 2014, 2012.

[9] W. Roongkaew, “Software Engineering Tools Classification based on SWEBOK Taxonomy and Software Profile,” pp. 122–128, 2013.

[10] A. Abran, P. Bourque, R. Dupuis, J. W. Moore (eds). Guide to the Software Engineering Body of Knowledge - SWEBOK. IEEE Press, 2004.

[11] https://en.wikipedia.org/wiki/Software_Engineering_Body_of_Knowledge

[12] J. Newkirk, “Introduction to agile processes and extreme programming,” in Proc. 24th Int. conf. Software engineering, May 2002, pp. 695-696.

[13] E. Mnkandla, and B. Dwolatzky, "A survey of agile methodologies," The transactions of the SA Institute of electrical engineers, vol. 3, pp.236-247, Dec. 2004.

[14] E. R. Mahajan and E. P. Kaur, "Extreme Programming: Newly Acclaimed Agile System Development Process," International Journal of Information Technology, vol. 3, no. 2, pp.699-705, 2010.

[15] A. Objectives, S. Approach, J. D. Arthur, S. D. Sheetz, and K. T. Stevens, "Assessing Agile Methods: Investigating Adequacy, Capability, and Effectiveness Computer Science and Applications Assessing Agile Methods: Investigating Adequacy, Capability, and Effectiveness," 2013

[16] F. Anwer, S. Aftab, and S. S. Muhammad, "Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum," no. May 2017.

[17] A. Qumer, B. Henderson-Sellers, measuring agility and adaptability of agile methods: a 4-dimensional analytical tool, Procs. IADIS International Conference Applied Computing 2006 (eds. N. Guimara˜es, P.Isaias, and A. Goikoetxea), IADIS Press, 2006, pp. 503–507.

[18] A. Qumer, B. Henderson-Sellers, Comparative evaluation of XP and Scrum using the 4D Analytical Tool (4-DAT), in Z. Irani, O.D.Sarikas, J. Llopis, R. Gonzalez, J. Gasco (Eds.), Proceedings of the European and Mediterranean Conference on Information Systems 2006 (EMCIS2006) CD, Brunel University, West London, 2006.

[19] B. Henderson-Sellers, M. K. K. Serour, C. Gonzalez-Perez, and a. Qumer, "Improving Agile Software Development by the Application of Method Engineering Practices," Proc. IASTED Int. Conf. Softw. Eng. SE 2007, pp. 55–60, 2007.

[20] M. Fernandes, “Classification and Comparison of Agile Methods,” 2010.

[21] M. Z. Nafchi and H. Zulzalil, “On the Current Agile Assessment Methods and Approaches,” pp. 251–254, 2014.

[22] Soundararajan S, Arthur JD. A structured framework for assessing the “goodness” of agile methods. Paper presented at the 18th IEEE International Conference and Workshops on Engineering of Computer-Based Systems, ECBS 2011; 2011; Las Vegas, NV.

[23] E. J. Braude (2001), Software Engineering: An Object-Oriented Perspective, John Wiley & Sons, New York, NY.

[24] S. Soundararajan, O. Balci, J. D. Arthur, and V. Tech, “Assessing an Organization’s Capability to Effectively Implement Its Selected Agile Method (s): An Objectives, Principles, Strategies Approach,”

[25] Soundararajan, S. (2013). Assessing Agile Methods: Investigating Adequacy, Capability, and Effectiveness.

[26] S. Soundararajan, J. D. Arthur, and O. Balci, "A methodology for assessing agile software development methods," in Agile Conference, Agile 2012, Dallas, TX, 2012, pp. 51-54.

[27] F. Anwer, S. Aftab, and S. S. Muhammad, “Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum,” no. May 2017.