ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Agility Assessment Model to Measure Agility Degree of Agile Software Companies

Taghi Javdani Gandomani* and Mina Ziaei Nafchi

Department of Computer Engineering, Boroujen Branch, Islamic Azad University, Boroujen, Iran;
tjavdani@yahoo.com

Abstract

This paper presents an Agility model to measure Agility degree of software companies. The model presented in this paper is easy to use and compatible to Agile principles and values. Focusing on Agile practices, this study identified the importance of Agile practices in being Agile. The underpinnings of the proposed model are Agile practices and their importance in achieving Agile values.

Keywords: Agility Assessment, Assessment Model, Agile Practices, Agile Software Development

1. Introduction

Software companies are replacing traditional methods in software development with Agile methods, primarily due to prevent the inherent challenges of traditional methods. However, transitioning to Agile is not an easy process and needs enough time and effort[1,2].

Several different values have been offered to software companies including quick and frequent delivery, embracing changes in requirements, increasing customer satisfaction, empowering development team, increasing team collaboration, achieving higher quality, and so on[3,4]. Different Agile methods focus on different Agile values. While some of them focus on achieving values in project management, some others focus on software development process. However, combination of them may help software companies to achieve the maximum possible values.

The ultimate goal of Agile transformation -leaving traditional methods and adopting to Agile methods- is being Agile. Achieving more Agility degree means that company can achieve more Agile values. However, measuring Agility degree is somewhat subjective. This is mainly because of the nature of Agility which is not a quantitative value. Furthermore, measurement approach in Agile software development is different from traditional approach[5]. Nonetheless, there are some studies focused on Agility assessment. These studies mainly have focused on comparing companies in term of Agility, considering particular Agility level, and goodness of Agile methods adopted by software companies. However, the primary disadvantage of the proposed methods, tools and assessment techniques is their limited scope and application[6].

We got motivated by lack of a well-known and comprehensive Agility assessment approach to conduct a research study in order to develop a simple and easy to use Agility assessment model. We particularly wanted to develop a quantitative based model to indicate Agility degree of software companies by a ratio scale. In this way, such a model, can simply a ratio value to each company and so, comparing software companies in term of Agility degree would be easy and perceptible.

The rest of this paper is organized as follows: Section 2 presents the previous studies and background of the study. Section 3 provides the proposed model, and finally, Section 4 concludes the paper.

*Author for correspondence

2. Related work

Several studies have been conducted regarding to Agile assessment, Agility measurement, and goodness of Agile methods. This section presents the most significant related work and provides a concise discussion on drawbacks of their proposed models or tools.

Sidky et al.[7] proposed a multi-level Agility model. Considering a set of objectives for each level, a set of Agile practices are mapped to each level. Indeed, the practices identified for each level reflect some of the values identified by Agile manifest to[8]. Like CMMI approach, they considered five levels of Agility. Companies adopted all practices in each level are considered in that level of Agility. Following CMMI approach, maintaining Agility level in each level of Agility requires adopting to all practices in lower Agility levels. The primary drawback of this Agility assessment model is forcing companies to accept a pre-defined set of practices for each level of Agility[6]. This seriously compromise the flexibility promised by Agile approach[6].

Qumer and Henderson-Sellers[9] proposed Agile Adoption and Improvement Model (AAIM) to measure degree of Agility of an Agile method. Like Sidky et al.[7], they also created a multi-level Agility model comprising six levels of Agility within three Agile blocks. They also used five parameters including "flexibility, speed, leanness, responsiveness, and learning" to define Agility[10]. The degree of Agility at each block is measured using the 4-DAT, "4 Dimensional Analytical Tool"[10]. The 4-DAT can examine an Agile method from four dimensions. Like sidky et al.'s model, AAIM follows CMMI approach and potentially reduces the flexibility afforded by Agile methods[11].

Two other approaches focused on Agility assessment based on the adopted Agile practices. These third party Agility assessment tools are Comparative Agility (CA) and Thoughtworks Agile Assessment survey[12,13]. The rationale behind CA is that usually software companies only intend to being more Agile than their competitors[13]. The assessment tools, developed based on the CA approach, help software companies to assess their Agility degree relative to other companies that responded to the tool[13]. This survey-based tool is freely accessible through a particular website[14]. A serious drawback of CA is that when comparing companies in term of Agility, it is unclear whether or not the adopted Agile practices are suitable for a company[11].

Toughtworks, a leading consulting company in Agile software development, developed an online Agility assessment tool[12]. This tool shows the Agility degree of an organization based on the twenty questions about existence of Agile practices in an organization. However, this tool does not identify the effectiveness of Agile methods in the survey. Also, it seems that considering only twenty practices is not enough, while there are many other Agile practices.

Soundararajan et al.[11] proposed a framework for assessing goodness of an Agile method. The main focus of this framework is assessing Agile methods from adequacy, capability, and effectiveness perspectives. Although this framework is helpful for assessing goodness of an Agile method compared to other Agile methods, it is not applicable to assess Agility degree of a software company.

Considering the advantages and disadvantages of the previous works, we intended to propose a model to assess Agility degree of an organization and overcome the drawbacks of the previous models.

3. Agility Assessment Model

The most important thing in proposing an assessment model is identifying measurement criteria.

3.1 Assessment Criteria

Agility assessment criteria need to have particular characteristics as follow.

3.1.1 Method-independent

The selected criteria should be method-independent; otherwise, they will lead to the superiority of one or some methods over others. For instance, "team size" may be used for identifying Agility, but since team size may vary from one Agile method to another Agile method, it could not be a good metric for assessing Agility. The best team size has been defined 7–9 for scrum[15], but the best team size for Crystal Family totally depends on the project type and criticality[16].

3.1.2 Agility Indicator

Good Agility assessment criteria should reflect an organizational or behavioral change[17]. In other words, assessment criteria should be different from those which are applicable in traditional methods. This causes that organization does not inherit those criteria from traditional methods. Indeed, any improvement in achieving these criteria directly is the result of trying to achieve more Agility.

3.1.3 Simplicity

Agility assessment criteria should be simple as much as possible. Obviously, those criteria which are simple and understandable are more effective and are more acceptable in industrial environments[18].

Considering the above features, Agile practices were considered as the assessment criteria to measure Agility of an organization. Although reviewing the literature may show more than 50 Agile practices, 44 Agile practices are the most important one[19]. These practices almost cover the concept of Agility in Software Company. The list of these practices will be presented in next section.

3.2 Criteria's Weight

The weight of each practices extracted from a previous survey performed previously by Prof. Laurie Williams. A decade after creating Agile manifesto, she intended to assess to what extent Agile principles and their associated practices are still important from the perspective of communities[19]. The importance of each Agile practice was asked using a five point likert scale (1= not very important; 5= essential).

Regarding to the Agile practices, 326 responses were received. However, some of the participants did not answer some of the questions. More detailed information about the survey and respondents is available in[19]. Table 1 shows the mean score of Agile practices and weight of them according to the responses received.

3.3 Agility Assessment Model in Action

With respect to the assessment criteria and their weights, the application of the model is quite easy. Agility degree of a company is calculated by the following formula.

$$\textbf{Agility} = \sum_{1}^{44} \frac{(P_i * W_i)}{range}$$

In this formula, P_i represents the extent of an Agile practice adopted in company. The accepted range for this value can be set by companies. However, in our model, we set it to equal 10. This number reflects the range in the above formula too. W_i shows the weight of P_i in our Agility model and can be obtained from the Table 1.

This model can be easy developed as a web service and software companies can use it simply to indicate their Agility degree based on the existence of Agile practices in their companies.

3.4 Comparison with Previous Models

Considering the previous models, we tried to prevent their drawbacks when developing this model. Our model does not consider any Agility level, like Sidky et al.[7] and Qumer et al.[9] models. It also does not force companies to adopt any particular Agile practices for achieving specific Agility degree or level. Comparing to the thoughtworks and CA, our model covers more Agile practices and highlights the importance of each Agile practice in achieving Agility. Finally, unlike the Soundararajan et al.[11] model, the main focus of our model is on all practices rather than any particular Agile methods. All of these advantages herald usefulness of the model comparing to the previous models.

4. Conclusion

Assessing Agility degree of software companies has been a center of focus for many of software engineering researchers. Although a few Agility assessment model have been proposed, they suffer from some serious disadvantages including incompatibility with Agile approach, less flexibility, limited scope and application, and so on. In this paper, we proposed a model which has none of the drawbacks of the previous models. We used 44 Agile practices and their importance (values) in achieving Agility in software companies to create an assessment model. The proposed model can simply calculate the Agility of a company based on the adopted practices in that company. It will be later implemented as a web service to be easily accessible by software companies.

5. Acknowledgement

This study was performed as a research plan and was financially supported by Islamic Azad University, Boroujen Branch, Iran. The authors also intend to express their gratitude to Prof. Laurie Williams who contributed to this study by having provided the data.

Table 1. Agile practices and their weights in Agility assessment model

Agile Practice	Weight %	Mean	Agile Practice	Weight %	Mean
"Done" criteria	2.61	4.4	Daily customer/product manager involvement	2.32	3.9
Short iterations (30 days or less)	2.61	4.4	Small teams (12 people or less)	2.32	3.9
Continuous integration	2.61	4.4	Emergent design	2.26	3.8
Automated tests are run with each build	2.61	4.4	Release planning	2.26	3.8
Embracing changing requirements	2.55	4.3	Configuration management	2.26	3.8
Features in iteration are customer-visible/customer valued	2.55	4.3	Informal design (no big design up front)	2.2	3.7
Iteration reviews/demos	2.55	4.3	Test-driven development acceptance testing	2.2	3.7
Prioritized product backlog	2.55	4.3	Team documentation focuses on decisions rather than planning	2.14	3.6
'Potentially shippable' features at the end of each iteration	2.55	4.3	Team velocity	2.14	3.6
Automated unit testing	2.55	4.3	Co-located team	2.14	3.6
Whole' multidisciplinary team with one goal	2.55	4.3	Requirements written as informal stories	2.08	3.5
Synchronous communication	2.55	4.3	Coding standard	2.02	3.4
Retrospective	2.49	4.2	Task planning	2.02	3.4
Collective code ownership	2.49	4.2	Ten minute build	2.02	3.4
Sustainable pace	2.49	4.2	Acceptance tests written by product manager	1.96	3.3
Complete feature testing done during iteration	2.43	4.1	Pair programming	1.9	3.2
Negotiated scope	2.43	4.1	Burndown charts	1.9	3.2
Refactoring	2.43	4.1	Design inspections	1.84	3.1
Timeboxing	2.38	4.0	Code inspections	1.84	3.1
Test-driven development unit testing	2.38	4.0	Stabilization iterations	1.72	2.9
Scrum meeting /Stand up meeting	2.38	4.0	Planning poker	1.72	2.9
Just-in-time' requirements elaboration	2.32	3.9	Kanban	1.66	2.8

6. References

1. Fraser S, Boehm B, Jarkvik J, Lundh E, Vilkki K. How do Agile/XP development methods affect companies? Paper presented at the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering, Oulu, Finland; 2006.
2. Gadnomani TJ, Zulzalil H, Abdul Ghani AA, Sultan ABM. Important considerations for agile software development methods governance. J Theor Appl Inform Tech. 2013; 55(3):345–51.
3. Cohen D, Lindvall M, Costa P. An introduction to Agile methods. Adv Comput. 2004; 62, 1–66. doi: 10.1016/S0065-2458(03)62001-2
4. Greer D, Hamon Y. Agile software development. Software - Practice and Experience. 2011; 41(9):943–44. doi: 10.1002/spe.1100
5. Javdani T, Zulzalil H, Abd Ghani AA, Md. Sultan AB, Parizi RM. On the current measurement practices in agile sofware development. International Journal of Computer Science Issues. 2012; 9(4):127–32. doi: http://www.ijcsi.org/papers/IJCSI-9-4-3-127-133.pdf
6. 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.
7. Sidky A, Arthur J, Bohner S. A disciplined approach to adopting agile practices: the agile adoption framework. Innovat Syst Software Eng. 2007; 3(3):203–16. doi: I 10.1007/s11334-007-0026-z
8. Beck K, Cockburn A, Jeffries R, Highsmith J. Agile manifesto; 2001. Available from: http://www.agilemanifesto.org
9. Qumer A, Henderson-Sellers B, McBride T. Agile adoption and improvement model. Paper presented at the 4th European and Mediterranean Conference on Information Systems, EMCIS; 2007; Valencia.
10. Qumer A, Henderson-Sellers B. A framework to support the evaluation, adoption and improvement of agile methods in practice. Journal of Systems and Software. 2008; 81(11):1899–19. doi: 10.1016/j.jss.2007.12.806
11. Soundararajan S, Arthur JD, Balci O. A methodology for assessing agile software development methods. Paper presented at the Agile Conference; 2012; Agile, Dallas, TX.
12. Thoughtworks. Agile Assessments; 2010. Available from: http://www.agileassessments.com/
13. Williams L, Rubin K, Cohn M. Driving process improvement via comparative agility assessment. Paper presented at the Proceedings of the 2010 Agile Conference; 2010.
14. Rubin K, Cohn M, Williams L. Comparative Agility (CA); 2010. Available from: http://comparativeagility.com/
15. Cohn M. Succeeding with Agile: Software Development using Scrum. Boston, MA: Addison-Wesley Professional; 2009.
16. Cockburn A. Agile software development. Boston, MA: Pearson Education, Inc; 2002.
17. Nierstrasz O, Lungu M. Agile software assessment (Invited paper). Paper presented at the 20th IEEE International Conference on Program Comprehension, ICPC 2012, Passau; 2012.
18. Spinellis D. Agility drivers. IEEE Software. 28(4); 2011. doi: 10.1109/ms.2011.72
19. Williams L. What agile teams think of agile principles. Communications of the ACM. 2012; 55(4):71–6. doi: 10.1145/2133806.2133823