

A Reference Model Based on Agile Values, Principles, and Aspects of Scrum, XP and Kanban to Foster the Adoption of Agility in Industry

César Pardo^{a,*}, Wilson Ortega^a, Omar S. Gómez^b

^a University of Cauca, GTI Research Group, Calle 5 # 4, 70, 19001, Popayán, Colombia

^b Escuela Superior Politécnica de Chimborazo, GrIIsoft Research Group, 060155, Riobamba, Ecuador

Corresponding author: *cpardo@unicauca.edu.co

Abstract—Currently, there is a broad range of software development agile approaches mainly based on the values and principles defined in the agile manifesto (AM). However, in many cases, their implementation is carried out informally and without being aligned with the values and principles stated. In practice, practitioners and consultants may lose sight of the AM recommendations, which could jeopardize the companies' agility, therefore, applying an agile approach does not make a company agile. In this article, we present a reference model called AgilityRef, which allows practitioners to support the understanding and implementation of the values and principles of the agile manifesto in the software development processes of a company, this, through twenty-two aspects defined from relations established between principles and values described in the AM and processes elements described in Scrum, XP, and Kanban. The evaluation of AgilityRef was carried out through a focus group where its completeness, understandability, and suitability were evaluated. Our findings suggest good enough acceptance by professionals and consultants who evaluated the proposal. The proposed reference model seems to allow professionals and companies to improve the understanding and implementation in practice of the concept of agility in their companies' software development processes, thus, minimizing the subjectivity and error of their process adoption, implementation, evaluation, and articulation with the principles and values of the agile manifesto.

Keywords— Agile approaches; assessment of agility; eXtreme programming; Kanban; Scrum.

Manuscript received 7 Nov. 2020; revised 20 Feb. 2021; accepted 27 Jun. 2021. Date of publication 31 Dec. 2021.
IJASEIT is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

Currently, it is possible to find a wide range of agile approaches used in the software industry. To take advantage of them, companies usually face a social and technical transformation process that generates changes in practices at the level of management, people, processes, techniques, technologies, and tools used with the main objective of being agile [1], [2]. Many of these challenges are addressed in the guiding concepts of values and principles that are declared in the agile manifesto (AM). The challenges involve removing authoritarianism elements, controlling and integrating elements of collaboration with teams [2].

However, as discussed in Ambler [3], an increasing number of challenges (36%) or failed (3%) projects associated with agile approaches have begun to appear. Reported failures indicate that projects, and thus companies, have not fully benefited from the advantages of agile

software development goals [4]. Companies tend to adopt agile practices, adapt them to their context, and convince themselves that they are agile. However, they fail to see positive results from their agile adoption. It seems that projects fail more often than with their traditional practices; some companies even misinterpret the agile concept and become undisciplined companies [4]. Therefore, adopting agile approaches not necessarily makes a company agile, team members should internalize agile principles and values, these are the litmus test to determine whether or not the processes of a software company are agile.

Therefore, it is necessary to support organizations in adopting agile approaches or practices and be guided to improve their agility [5]. From the analysis of the results of a systematic mapping performed [6], it has been possible to find eighteen solutions to support the adoption or assessment of agility in software organizations. However, we note that there is no consensus about criteria to assess the agility, a clear terminology to relate agile approaches and the

assessment of agility, and a relationship between agile values, principles, and elements defined in agile approaches. In general, most of the proposals do not completely cover the principles and values expressed in the agile manifesto; even they are not considered as a complement or extension to facilitate their understanding and adoption.

In order to address this need, we propose AgilityRef, a reference model to facilitate the understanding and implementation of the values and principles of the agile manifesto in the software development processes of a company. Our model differs from the proposed reference models, which are based on proposing a list of agility elements to consider, but which have the disadvantage of understanding what to do with each of the elements they suggest. Thus, AgilityRef provides a detailed and disaggregated description of each of the proposed aspects. The processing element is explicit when a company wants to design the metrics to support the implementation level of the agility processes. In addition, the proposed aspects show their relationship between agile values and principles. On the other hand, the aspects proposed in AgilityRef arise from identifying the relationship between elements of agile approaches, and the software industry widely uses Scrum, XP, and Kanban according to the state of agile report [7] and agile principles.

The rest of the article is organized as follows: Section II analyzes the related work, whereas Section III discusses the research method used to define our proposal. Section IV sets out in detail AgilityRef, a reference model that can support users to understand what to consider achieving agility in their development processes; this section also presents the evaluation of the model's completeness, suitability, and comprehensibility a focus group. Finally, Section V presents conclusions and future work.

A. Related Work

As discussed in the systematic mapping review reported in Ortega *et al.* [6], there is a growing research interest regarding the assessment of agility in software companies. According to Ortega *et al.* [6], 22.2% of the studies [8], [9], [10], [11] propose to assess the software process used by the company. The same percentage of the studies [12], [13], [14], [15] propose to assess a particular development approach, e.g., Scrum, XP, or an adaptation. Still, 22.2% of the studies [16], [17], [18], [19] focus on assessing the agility of companies. 16.7% [4], [20], [21] focus on assessing the agility of software development projects. 11.1% [22] propose to assess the agility of companies and their projects. Finally, a 5.5% study proposes to assess the agility of development teams [23].

Taking as reference the work in Ortega *et al.* [6], we found 18 studies, which were analyzed and compared taking into account: (i) the type of study; (ii) the inclusion of the principles, agile values, and other evaluation criteria used in the proposed solutions; (iii) the evaluation and validation methods used; and (iv) the proposed tools. In addition, the factors that influence the assessment of agility in companies were determined.

Based on Ortega *et al.* [6], we noted that there is no consensus in the definition of agility in software companies, and therefore, there is no agreement regarding the criteria

that should be considered a reference to assess it. We note that none of the studies present in a clear and detailed way the terminology related to agile approaches and the assessment of agility, which leads to difficulties applying assessment approaches by the companies themselves. It can also be observed that none of the found proposals presents clearly and succinctly the relationship between the agile principles and values and the elements of agile approaches. It is hard to establish a degree of agility according to the agile manifesto; this makes it difficult to interpret the evaluation results, identify opportunities for improvement and formulate solutions that impact the process elements of the companies. Regarding the proposed software tools, they are focused on assessing the agility of the company, the team, or the project, but there is no evidence of the existence of a model to assess the agility of software processes in companies.

II. MATERIALS AND METHOD

In order to build the proposed agility reference model, we used as reference the concepts of values and principles that are declared in the agile manifesto [24]. We provided a detailed description of the aspects that foster agility. We designed the assessment instruments that allow knowing the state of agility reached in a company and thus establish improvement actions more objectively.

In this sense, the agile values and principles were adopted; in the case of values, a more detailed description was added. Then, the relationships between values and agile principles were identified. Finally, aspects of the most used agile approaches were identified. These were obtained by analyzing the activities, roles, products, and tools proposed in agile approaches such as Scrum [5], XP [26], and Kanban [27], which are directly related to one or more agile principles. The selection of these approaches was made considering that they are the most used in the current software industry, according to the results of the annual agility survey State of Agile Report [7], where it is possible to observe that Scrum is the most used approach by the consulted companies, with 58%, followed by the combination of Scrum with Kanban [27] with 10% and the combination of Scrum and XP with 8%. The components and relationships of AgilityRef arose from the creation of the OntoAgile ontology presented in Ortega *et al.* [28], which provides concepts, definitions, and relationships around agile approaches and software development processes.

Our model, called AgilityRef, is divided into three main components: (i) agility aspects, (ii) agile principles, and (iii) agile values.

- *An aspect of agility* describes a processing element (activity, role, product, tool) that evidences implementing one or more agile principles. AgilityRef considers that the agility of a software development process consists of implementing process elements that contribute to the fulfillment of agile principles.
- *Agile principles* describe the characteristics that an agile software development process must have and support the fulfillment of one or more agile values in the development process.

- *Agile values* represent the main attributes that a software development process must have to be considered agile, as stated in the agile manifesto [24].

III. RESULTS AND DISCUSSION

In this section, AgilityRef is described in agile values, agile principles, and agility aspects. Also, the evaluation through a focus group is presented.

B. The Agile Values and Principles

Table I shows each of the agile values; an identifier is set to facilitate their reference in this document. The interpretation of these values in AgilityRef results is from analyzing the values described in the agile manifesto. The agile manifesto [24] defined the twelve agile principles and provided general guidelines so that a software development process is aligned with agile values. AgilityRef is adopted without being modified. Only an identifier was assigned to facilitate their use in the document, and it goes in the order in which they appear in *Beck et al.* [24]. Due to the space limit, these are not presented in the document.

The relationship between principles and agile values proposed in AgilityRef is shown in Table I, in which it can be seen that the majority of agile principles are related to the agile value 01, with a total of 7 principles, confirming that an agile development process is people-centered. The second value with more related principles is the agile value 02, with a total of 3 principles, which reflects the importance of an agile process to obtain as output a product that provides value to the customer. Finally, the agile values 03 and 04 have a relationship with each one of them, related to the agile principles 04 and 02, respectively.

C. Agile Aspects

Agile aspects correspond to artifacts such as activities, roles, products, and tools that demonstrate compliance with agile principles in a software development process. It is important to highlight that the aspects describe the “what” and not the “how,” because companies that implement agile approaches should define their processes according to their context and pillars of the agile values and principles. Table II presents the agile aspects proposed, their respective identifier, name, description, source, type of artifact that it represents, and agile principles with which it is related. It also shows the agile aspects and their relationship with agile principles.

D. Evaluation Through a Focus Group

The proposed model was evaluated through a specific qualitative research method, the focus group method [25]. According to Barbour [25], the focus group method is faster and effective for gathering information from practitioners and users. Therefore, it is possible to obtain a rich and qualitative report of information with experiences revealing ideas and feedback difficult or expensive to obtain through other methods. The activities carried out together with the experiences reported by professionals and users are summarized below.

1) *Defining the research problem:* In this activity, a focus group objective was defined, which was aimed at knowing the opinion and perception of professionals with experience in agile software development regarding the completeness, suitability, and comprehensibility of AgilityRef. The evaluation objectives were focused on: (i) evaluating the proposal; (ii) obtaining recommendations for lessons learned; and (iii) update the proposal based on the recommendations suggested by the participants.

TABLE I
INTERPRETING AGILE VALUES IN AGILITYREF

AV	Agile Value	Description	AP
01	Individuals and interactions over processes and tools.	Agile processes should be people-centered, not process or tools-oriented. There is no denying the importance of having processes that guide work and tools that improve their efficiency. However, it is much more important to have a team of motivated people with excellent technical, communication, and self-organization skills to react quickly to change, interact with the customer and their teammates, and generate products that offer value to the customer. The processes and tools should be adapted to the team and not vice versa.	05, 06 08, 09 10, 11 12
		Although the documentation generated during a software development process is important, the main objective of the process should be to deliver iteratively to the customer; the product increases that work and adds value to the business. Each of the deliveries of product increments must be accompanied by the minimum necessary documentation and must also exceed a set of minimum acceptance criteria by the customer.	01, 03 07
02	Working software over comprehensive documentation.	An agile development process requires a customer to collaborate frequently with the team during product development. This collaboration will allow prioritizing features according to the value they will bring to the business and detect risks early. If possible, the customer should be available at the product development site to make communication more effective, but if this is not possible, a customer representative with decision-making power should be available.	04
03	Customer collaboration over contract negotiation.	Unlike traditional development approaches where the goal is to have a detailed plan from the start, an agile development process plans only what is necessary to start product development. As each iteration progresses, the changes detected from communication with the customer are an opportunity to deliver a product that will add more value.	02
04	Responding to change over following a plan.		

Acronyms used: AV = Agile Value, AP = Agile Principles

TABLE II
AGILITY ASPECTS IN AGILITYREF

AA	Name	Description	Source	Artefact	AP
01	Iterative and incremental	The product is created in an iterative and incremental way. The intervals have a maximum duration of one month. Also, each new iteration begins immediately after the completion of the previous iteration.	Scrum, XP, Kanban	Activity	01, 03
02	Increments with value	Each iteration generates a potentially useful and functional version of the product.	Scrum	Product	01, 03, 07
03	Negotiable scope	The scope of each iteration can be clarified and renegotiated between the team and the customer.	Scrum, XP	Activity	02, 04
04	Daily meeting	The team meets daily to assess progress toward the iteration goal. The length of the daily meeting is limited to a time that allows each participant to concisely share what they did the day before, what they plan to do that day, and impediments to achieving the iteration's goal.	Scrum, Kanban	Activity	06, 12
05	Refinement of requirements	The product requirements list is constantly updated to identify what the product needs to be suitable, competitive, and useful. Additionally, the requirements can be updated, including more detail, estimates, and priorities.	Scrum	Activity	01, 02, 10
06	Team self-organization	Each member of the team can choose the best way to carry out the tasks under their responsibility. The team can give higher priority to the product requirements that present the highest risk.	Scrum, XP	Role	05, 11
07	Shared responsibility	Team members may have specialized skills in areas they are most focused on, but product responsibility rests with the entire team. Any member of the team could make changes to any of the components of the product.	Scrum, XP	Role	05
08	Steady pace	The duration of the iterations is consistent throughout the development of the product.	Scrum	Activity	08
09	Negotiable iterations	An iteration can be canceled when its goal is determined to be obsolete. For example, if market or technology conditions change.	Scrum	Activity	02
10	Retrospectives	At the end of each iteration, the team and stakeholders review the results regarding people, processes, and tools. The most important elements that went well and possible improvements are identified and ordered, a plan is created to implement the improvements, and the list of product requirements is adapted if necessary.	Scrum	Activity	04, 06, 09, 12
11	Elimination of impediments	It is practiced assigning a responsible role to help the team and stakeholders carry out activities in the best way and eliminate any impediments.	Scrum, XP	Role	05
12	Customer engagement	Customers can decide which parts of the problem should be resolved with higher priority so that the team can focus on what delivers value. Customers participate in creating functional tests of the product and in the choice of technology to be used for its development. A real customer is available at the same team location to answer questions, resolve discussions, and set priorities on a small scale. If it is not possible to have a real customer, we must have a representative of the customer and those interested in the product to make decisions.	Scrum, XP	Role	01, 04, 07
13	Team estimation	The team oversees estimating the time it will take to implement a product feature.	XP	Role	05, 11
14	Customer feedback	When the team makes technical decisions that may have an impact at the business level, it must inform the interested parties.	XP	Role	04
15	Shared vision	A vocabulary and a vision of the product to be developed are defined and shared by the team and customers.	XP	Activity	04
16	Attention to good design	During the design of the product, it is considered that components are not duplicated, all the aspects considered important by the team are included, and only what is required at the moment are included. Every time a new feature is added to the product, it is checked if its internal structure can be modified to make it easier. If possible, changes are made, and all tests are verified to work.	XP	Activity	09, 10
17	Collaborative work	When a team member implements a new feature or makes a product change, they can ask another team member for collaboration to work together.	XP	Activity	11
18	Continuous integration	All changes to the product are tested and integrated after a few hours.	XP	Activity	01, 03, 07
19	Technical rules	A set of rules is agreed upon that the entire team must comply with when adding functionalities to the product or making modifications.	XP	Activity	09
20	Use of metrics	Metrics that provide relevant information on team performance are used.	XP, Kanban	Activity	08, 12
21	Continuous integration tools	Tools are used to support the continuous integration of new functionalities and changes made to the product.	XP	Tool	01, 03, 07, 10
22	Automatic testing	Tools are used to support the execution of automatic tests.	XP	Tool	01, 03, 07, 10

Acronyms used: AA = Agile Aspect, AP = Agile Principles

2) *Selecting the participants*: For the selection of experts, the following criteria were defined: (i) to be active in the software industry or the academic environment, either as a teacher or as a student; (ii) to have knowledge of agile approaches and their application in the software industry; and (iii) to be a professional with experience in the software industry at least 7 years, what was verified with proven and certified experience. There was a list of 10 potential participants during recruitment, of which 4 were discarded, and only 6 were selected. Once the participants were selected, an e-mail invitation to coordinate the discussion session was sent, suggesting a date and time; when we had an affirmative response from the participants, a second e-mail with the proposal documentation was sent 3 weeks before the suggested date for the discussion.

3) *Conducting the focus group session*: During this activity, we held a focus group session. The discussion session was conducted and coordinated by one member of the research group (acting as moderator) and another person as rapporteur. The order and sequence of the session were previously sent to the participants. During the session, the rapporteur was the person in charge of taking note of each observation and comment made by the participants; this allowed the moderator to concentrate full time and not have to interrupt the group session. Besides, in support of the comments received, participants were asked to complete a questionnaire at the end of the discussion. The subjective influence of the moderator in the discussion was minimized since the discussion was structured, and a questionnaire was

applied in order to each expert had to answer. The main task of the moderator was to listen and deepen when necessary, requiring that the moderator should be able to understand issues and addressing them quickly. It is often necessary to paraphrase the points of the participants to confirm that the contribution has been understood correctly. The main task of the moderator was to listen; he was prepared to answer any question, deep and technical though it was. On the other hand, there were no last-minute cancellations of the participants.

4) *Data analysis and reporting*: Once the results were obtained, the research group carried out an analysis of the questionnaires by counting the responses of the participants. To carry out the questionnaire, it was taken into account that the questions were aimed at determining the degree of relevance, completeness, and clarity of the research proposal; for this purpose, Table III presents the questions asked, these questions allowed to measure the completeness (2 questions), comprehensibility (4 questions) and suitability (2 questions) of the proposed model (questions 1-8). In addition, 2 open-ended questions were included that allowed participants to propose adjustments to the model and make additional comments (questions 9-10). Questions 1-8 used a level of conformity through a Likert scale as follows: Strongly Disagree (SD): value (1), Disagree (DI): value (2), Neither Agree nor Disagree (NAD): value (3), Agree (AG): value (4) and Strongly Agree (SA): value (5), see Table III. Fig. 1 graphically indicates the distribution of the results obtained from questions 1-8.

TABLE III
EVALUATION QUESTIONNAIRE USED IN THE FOCUS GROUP

Attribute	Id	Questions
Completeness	Q1	Do you consider that the proposed agility indicators cover the process elements (activities, roles, work products and tools) that must be present in a software process that implements agile principles?
	Q2	Do you consider that AgilityRef contains all possible relationships between principles and agile values?
	Q3	Do you consider the structure of AgilityRef to be easy to understand?
Understandability	Q4	Do you consider that the proposed agility indicators are easy to understand?
	Q5	Do you consider that the proposed relationships between values and agile principles are easy to understand?
Suitability	Q6	Do you consider that the proposed relationships between indicators of agility and agile principles are easy to understand?
	Q7	Do you consider that the proposed agility indicators serve as the basis for defining questions and metrics that allow knowing the degree of implementation of agile principles in a software process?
Open-ended questions	Q8	Do you consider that AgilityRef can be used to define an evaluation method that allows knowing the degree of agility of software processes?
	Q9	Do you think that the elements proposed in AgilityRef should be added, removed or modified? Please explain the proposed changes.
	Q10	Do you have any additional comments about AgilityRef?

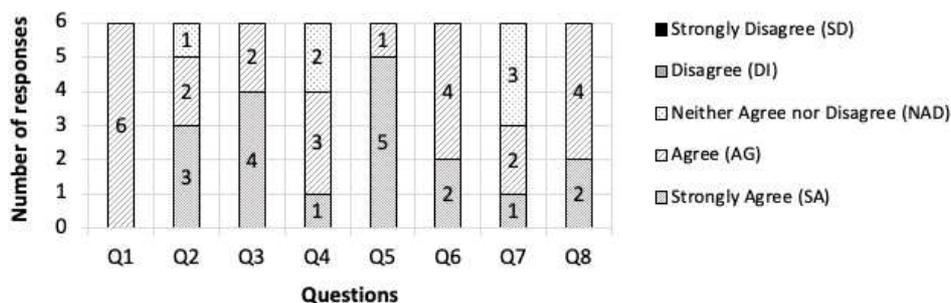


Fig. 1 Consolidation of questions 1-8 answered by the focus group

As can be seen, in general, there was a consensus with the proposal. This means that the elements presented and evaluated during the focus group session were mostly relevant for assessing agility in software companies. However, questions Q2, Q4, and Q7 were unfavorable (i.e., neither agree nor disagree according to the defined Likert scale). Hence, these were considered on the proposal as improvement actions. Likewise, the comments and opinions of the participants were selected and taken into account to carry out improvement actions on the proposal. Thus, obtaining a second version, which is presented in this document, the comments and opinions are as follows:

- It is clarified that the definition of assessment of agility of software development process used consists of the implementation of process elements that contribute to the fulfillment of agile principles.
- The label of *indicator of agility* is replaced by the label *aspect of agility* to refer the elements to be taken into account in an assessment.

5) *Research construct*: To ensure that the research construct in this study was valid and in line with our research objectives, we used three techniques as follows:

- Maintaining the content and format established for the focus group session.
- The instrumentation errors were reduced by making an audio recording.
- The potential bias in the interpretation of the results was reduced because a person external to the research reviewed all the interpretations made during the analysis.

E. Limitations

During the focus group, some limitations and solutions emerged, they are as follows:

- Although there were a predefined format and agenda, due to lack of experience, it was not so easy for the moderator to have control over the style of discussion on the least active participants. This was corrected by the most experienced researchers as soon as it was detected.
- Some embarrassing situations such as the incorrect responses of the participants were mitigated with the active and quick participation of the moderator
- To reduce the risk of participants' limited knowledge and understanding, they were selected taking into account the same level of experience.

In addition, reading material was provided early, and those complex topics were divided into easier-to-understand pieces.

IV. CONCLUSION

In this article, a reference model has been presented to support the understanding and implementation of agility in a company's software development processes. Our proposed model adopts agile values and principles and establishes your relationship. In addition, it establishes in an explicit, formal, and reusable way a set of fundamental agile aspects to consider to facilitate the agility assessment in software processes. These aspects have been obtained from the establishment of the relationships of process elements proposed in agile approaches and Scrum, XP, and Kanban,

approaches widely used in the industry according to the latest State of Agile Report [7].

The model differs from existing solutions that generally list the elements to be taken into account but do not detail, clarify and document what to do with those elements, making it difficult to understand and apply them. Our approach proposal supports the software companies and practitioners in understanding what to do in each aspect established. Therefore, the details of the aspects list were broken down and made more explicit and comprehensible by incorporating both the kind of process element and agile principle related.

The completeness, suitability, and comprehensibility model were evaluated by a group of experts and professionals in agile approaches. This allowed observing the good acceptance by the participants, who agree that the use in industry of the proposed model could facilitate the adoption, implementation, and assessment of the level of implementation of agility in the software processes, and add the need for this kind of solutions to guide the work carried out with agile approaches. In addition, with the evaluation results, the participants made suggestions that were considered opportunities for improvement to the proposed model and were considered to generate the version presented in this document.

One limitation of AgilityRef is that currently, it just proposes a reference about what aspects to consider but not how to assess them. Accordingly, as future work, we expect to incorporate a self-assessment instrument based on AgilityRef, which provides a set of elements to support this activity.

ACKNOWLEDGMENT

The authors are grateful for the contribution of the University of Cauca. Also, we acknowledge the contribution and participation of practitioners who were part of the focus group and who made it possible to accomplish this work.

REFERENCES

- [1] P. Patanakul and R. Rufo-McCarron, "Transitioning to agile software development: Lessons learned from a government-contracted program," *The Journal of High Technology Management Research*, vol. 29, no. 2, pp. 181-192, Nov. 2018.
- [2] R. Vallon, B.J. da Silva Estácio, R. Prikladnicki and T. Grechenig, "Systematic literature review on agile practices in global software development," *Information and Software Technology*, vol. 96, pp. 161-180, Apr. 2018.
- [3] S. Ambler. (2018) IT Project Success Rates Survey Results. [Online]. Available: <https://bit.ly/3fwXcz>.
- [4] Ö. Özcan-Top and O. Demirors, "Application of a software agility assessment model – AgilityMod in the field," *Computer Standards & Interfaces*, vol. 62, pp. 1-16, Feb. 2019.
- [5] C. Rupp. *Scaling Scrum Across Modern Enterprises: Implement Scrum and Lean-Agile techniques across complex products, portfolios, and programs in large organizations*. 1st ed., Packt Publishing, Aug. 2020.
- [6] W. A. Ortega, C. J. Pardo, and F. J. Pino, "Systematic mapping on the agility evaluation in software development organizations," *Journal ITECKNE*, vol. 16, no. 1, pp. 64–76, Jun. 2019.
- [7] VersionOne Inc. & Collabnet, (2020) 14th Annual State Of Agile Report. [Online]. Available: <https://bit.ly/3hw5ek3>.
- [8] F. McCaffery, P. S. Taylor, and G. Coleman, "Adept: A unified assessment method for small software companies," *Journal IEEE Software*, vol. 24, no. 1, pp. 24–31, Feb. 2007.
- [9] B. Shen and D. Ju, "On the measurement of agility in software process," in *Proc. ICSP'07*, 2007, p. 25.

- [10] D. M. Shawky and A. F. Ali, "A practical measure for the agility of software development processes," in *Proc. ICCTD'10*, 2010, p. 230.
- [11] TongShi, JianbinChen, and J. Zhu, "Study on assessment framework of software process in agile," in *Proc. ICIS'10*, 2010, p. 498.
- [12] A. Qumer and B. Henderson-Sellers, "An evaluation of the degree of agility in six agile methods and its applicability for method engineering," *Journal of Information and Software Technology*, vol. 50, no. 4, pp. 280–295, Mar. 2008.
- [13] M. Taromirad and R. Ramsin, "CEFAM: Comprehensive evaluation framework for agile methodologies," in *Proc. IEEE Software Engineering Workshop*, 2008, p. 195.
- [14] S. Soundararajan, J. D. Arthur, and O. Balci, "A Methodology for Assessing Agile Software Development Methods," in *Agile Conference*, 2012, p. 51.
- [15] M. Florêncio, F. Sabinelli, and M. A. Francisco Borges, "ASA: Agile Software Development Self-assessment Method," in *Proc. gile Methods*, 2018, p. 21.
- [16] V. Escobar-Sarmiento and M. Linares-Vásquez, "A model for measuring agility in small and medium software development enterprises," in *Proc. LACI'12*, 2012, p. 1.
- [17] K. M. Poonacha and S. Bhattacharya, "Towards a framework for assessing agility," in *Proc. of the HICSS'12*, 2012, p. 5329.
- [18] R. Wandler, "Development of the organizational agility maturity model," in *Proc. FedCSIS'14*, 2014, p. 1197.
- [19] Scrum Manager. (2018) Scrum Level Model. [Online]. Available: <https://bit.ly/3cUYs3Z>.
- [20] F. Mafakheri, F. Nasiri, and M. Mousavi, "Project agility assessment: an integrated decision analysis approach," *Journal of Production Planning & Control*, vol. 19, no. 6, pp. 567–576, Dec. 2008.
- [21] O. E. Adali, O. O. Top, and O. Demirors, "Assessment of agility in software organizations with a web-based agility assessment tool," in *Proc. SEAA'17*, 2017, p. 88.
- [22] M. Ikoma, M. Ooshima, T. Tanida, M. Oba, and S. Sakai, "Using a validation model to measure the agility of software development in a large software development organization," in *Proc. ICSE'09*, 2009, p. 91.
- [23] Spotify Labs. (2014) Squad Health Check model - visualizing what to improve. [Online]. Available: <https://bit.ly/3fj0lqm>.
- [24] K. Beck *et al.*. (2001) Manifesto for Agile Software Development. [Online]. Available: <https://agilemanifesto.org>.
- [25] R. Barbour. *Doing Focus Groups*. 2nd ed. Sage Publications, 2018.
- [26] D.A. Putri, R. Razali, and Z. Mansor, "Team Formation for Agile Software Development: A Review," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 10, no. 2, pp. 555-561, 2020.
- [27] M. Alqudah, and R. Razali, "An Empirical Study of Scrumban Formation based on the Selection of Scrum and Kanban Practices," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 6, pp. 2315-2322, 2018.
- [28] W.A. Ortega, C. Pardo, and F. Pino, "OntoAgile: an ontology for agile software development processes," *Dyna*, vol. 86, no. 209, pp.79-90, 2019.