Journal of Theoretical and Applied Information Technology
31st May 2021. Vol.99. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

METHODOLOGY ASSESSMENT FRAMEWORK (MAF)

EBRU DALBUDAK[1], KARIM EL GUEMHIOUI[2], AHMED LAKHSSASSI[3], REJEAN GRAVEL[4]

[1]Department of Computer Science and Engineering, Université du Québec en Outaouais, Canada
[2]Department of Computer Science and Engineering, Université du Québec en Outaouais, Canada
[3]Department of Computer Science and Engineering, Université du Québec en Outaouais, Canada
[4]The Dome Partners, Gatineau, Canada
E-mail: dale02@uqo.ca, karim.elguemhioui@uqo.ca, ahmed.lakhssassi@uqo.ca, gravel.rejean@videotron.ca

ABSTRACT

Selecting the appropriate SDM for a project is one of the most important steps to ensure the project success [1, 2]. There are so many SDMs in the market but how do we choose the right one? How do we determine and measure the concept of "fit" between the chosen SDM and the project?

This research introduces a new framework called Methodology Assessment Framework (MAF) which helps decision makers assess a given project against the seven factors and determine the type of SDM that would be best suited whether it be an agile, plan-driven methodology or a hybrid of the two. This tool is based on seven decision factors, which are outcomes, scope, CYNEFIN (complexity), constituents, agile principles, team knowledge & experience, and organization capability & maturity. The paper explains each of the seven factors that MAF uses along with their assessment metrics to appraise a given project and based on the evaluation results, suggests whether the project should be run using an agile or plan-driven methodology. Next, it presents a cases study which demonstrates the application of this new framework into an Electronic Design Automation (EDA) tool development project, it identifies the agile/iterative methodology as the most suited SDM for the project.

Keywords: *Systems Development Methodology, SDM, Methodology, Agile, Framework*

1. INTRODUCTION

One of the most important steps in achieving project success is to select the appropriate methodology. Making an informed System Development Methodology (SDM) decision must be the first step of every system project. While there are so many SDMs such as the agile methodologies and plan-driven methodologies to choose from, finding the most appropriate SDM for a given project seems to be a great challenge. How do we choose the right one?

Micic [3] argues that the process of selecting a methodology is more "subjective and less precise than technical" and he states "the choice of methodology, among other things, greatly depends on the size of the organization, the type of technology used, the style of management, and the structure of employees, locations, companies, clients/users and a number of other factors that need to be taken into account" [3]. According to Jones "selecting a software development methodology has more in common with joining a cult than it does with making a decision" and "many companies do not even attempt to evaluate methods, but merely adopt the most popular, which today constitute the many faces of agile" [4]. According to Mullaly & Thomas, "attaining fit suggests that there is an alignment between what is being implemented and the environment and situation of an organization"[5].

Our literature review has proven that there are several frameworks that have been used to define, compare and evaluate SDMs [3, 6-8]. However, there is yet no comprehensive framework that uses the crucial factors to determine whether the traditional/plan-driven/waterfall, agile or hybrid type of system development methodology would be most suited for a project.

To address the need of a comprehensive evaluation framework, this research proposes a new Multi Criteria Decision Aid (MCDA) framework called "Methodology Assessment Framework (MAF)". The MAF framework is based on seven decision factors to come up with an overall assessment that will support the decision-making process for

choosing the system methodology approach that is best suited for the project.

The article begins with the background and literature review relevant to the study as an overview. Next, it introduces the new MAF framework along with a detailed description of each of its seven decision factors. The article proceeds with the case study to demonstrate the use of the MAF framework in an Electronic Design Automation (EDA) tool development project. And it ends with concluding remarks.

2. BACKGROUND AND LITERATURE REVIEW

Alqudah & Razali performed a systematic literature review of 53 articles that were published between 2001 and 2015 to identify the key factors to be considered when selecting an appropriate agile method and they identified "the nature of the project, development team skills, project constraints, organizational culture and customer involvement" as the crucial factors that can be used in selecting Agile methods [8].

Griffiths introduced various agile suitability filters, such as the Gartner bi-modal IT, Alistair Cockburn's Crystal family of methods, DSDM Suitability filter, Boehm and Turner's radar chart and the organizational suitability filter, to help assess if an agile approach is suitable to an organization and project [7]. Gartner bi-modal IT uses three attributes (the perceived degree of governance, likelihood of change and the type of solution) to suggest if a plan-driven or an agile would be the right approach. Alistair Cockburn's work [9] argues that the team size and system criticality are two factors that should be considered for assessing the project's agile suitability. DSDM suitability filter uses a list of Yes/No questions to determine if a project would benefit from an agile development approach. Boehm and Turner [10] define five crucial attributes (personnel, project criticality, dynamism, project size and culture) to assess a project for either an agile approach or more of a traditional/plan-driven approach. "This approach is an exceptional contribution to the notion of tailoring the software process to match the project context" [11].

Datta [12] introduced an "Agility Measurement Index" as an indicator for determining whether a Waterfall, Unified Software Development Process (UP), or eXtreme Programming (XP) methodology should be used in a project.

Abrahamsson et al. [13] used a comparative framework to compare ten agile methods based on six analytical criteria: project management support, software development lifecycle coverage, availability of concrete guidance for application, adaptability in actual use, research objective and empirical evidence.

Qumer et al. [14] introduced the 4 Dimensional Analytical Tool called 4-DAT which uses four attributes (method scope, agility, agile values, software process) for the analysis and comparison of XP and SCRUM agile methods.

Taromirad & Ramsin [15] researched and assessed the evaluation frameworks that were used to facilitate the selection of an appropriate agile development methodology and concluded that "although several evaluation frameworks or methods have been introduced for comparing, analyzing or evaluating agile methodologies, they lack in addressing method engineering and project management requirements". In another study, Taromirad & Ramsin introduced a new evaluation framework called CEFAM, which uses five attributes (the nature of the project, development team skills, project constraints, customer involvement and organizational culture) to guide decision makers in the selection of an appropriate agile method [6].

There are several studies in the literature that compare traditional/waterfall methodologies with agile and discuss project characteristics that will make a difference when it comes to choosing the right development methodology [16]. Jones introduced several standard metrics, such as "function points, defect removal efficiency (DRE), Cost of Quality (COQ), and Total Cost of Ownership (TCO) to compare a sample of contemporary software development methods" [4].

However, even though several studies have investigated the factors on the selection of agile methods, according to our knowledge, there is yet no comprehensive framework that uses the crucial factors to determine whether a traditional/plan-driven/waterfall, agile or hybrid type of SDM would be most suited for a project. This research aims to address this need by introducing a new comparative framework called MAF that provides a full coverage of seven factors that are regarded as important in the literature when it comes to making such a decision.

3. MAF AND ITS SEVEN DECISION FACTORS

"A framework is a well-used method for clarifying the properties of, and comparing methodologies" [17]. Through a review of the literature, this study found out that existing evaluation frameworks lack several aspects. Most of them have not considered evaluating agility. The nature and the complexity of the project as well as the extent and enrollment of the project's constituent base have been neglected or only partially addressed. To address the need of a comprehensive evaluation framework, this research has identified seven factors that are critical to providing the project's executive sponsor and governance body with a dashboard view over the project landscape. They are as follows:

1. Outcomes being addressed by the project (OUTCOMES)
2. Scope /features of the project (SCOPE)
3. Nature and complexity of the project - CYNEFIN framework (CYNEFIN)
4. Extent and enrollment of the project's constituent base (CONSTITUENTS)
5. Applicability of Agile principles to the project (AGILE PRINCIPLES)
6. Team expertise and experience in system development methodologies (TEAM)
7. Maturity of the organizations involved on the project (ORGANIZATION)

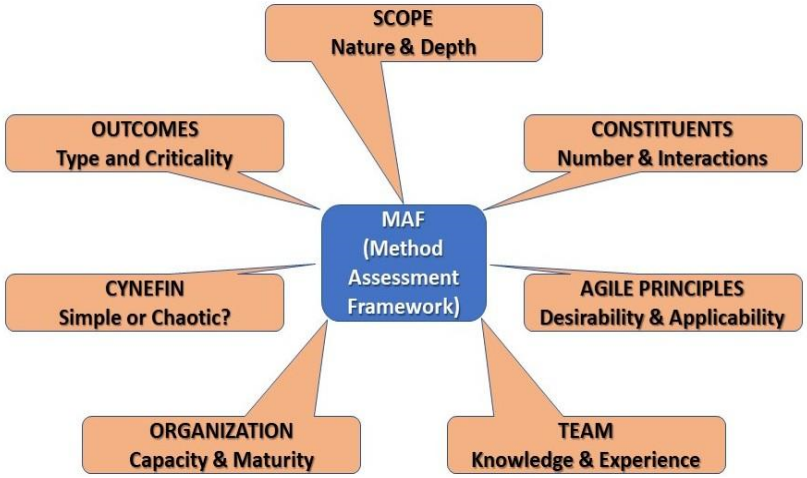

Figure 1 : MAF framework and its 7 decision factors

Table 1: Description and Ratings for 7 key factors that MAF uses presents, for these seven factors, the postulate for the selection of the appropriate system development methodology and the approach to assessing and rating each factor. Each factor and their assessment metrics will be discussed in more detail later in the document.

First, the project is evaluated across each of the seven factors by using the self-evaluation tools that are presented in the ANNEX: SELF-EVALUATION TOOLS. The radar chart in Figure 2: 7 key MAF dimensions affecting SDM selection can then be used to plot the assessment ratings to find out where your project currently is with respect to the 7 key axes of the MAF framework.

While some projects will be a good fit agile for others, using agile would be more problematic and they might instead favor a more traditional approach.

According to MAF, if all of the ratings are near the center, an agile SDM should be chosen. If they are at the periphery, a plan-driven SDM would be a better choice. If they are mostly in one or the other, a hybrid approach should be considered as plan-driven and agile approaches can be successfully combined to effectively address hybrid or hard to classify projects. In other words, "You can use all agile some of the time and some agile all of the time." [18].

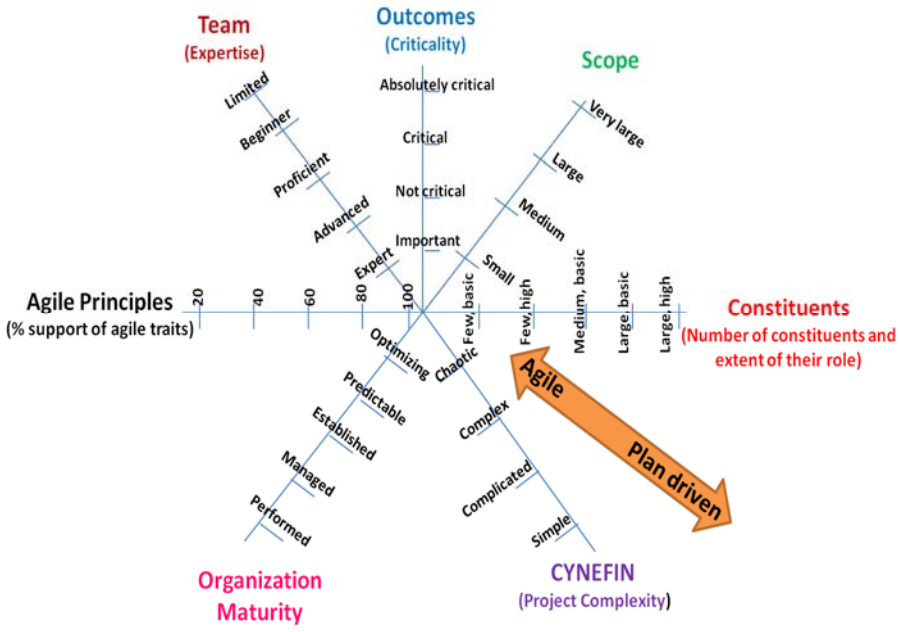

Figure 2: 7 key MAF dimensions affecting SDM selection

This framework is a tool that factors in these seven factors to come up with an overall assessment that will help determine the system methodology approach that is best suited for the project. Each of these factors are assessed and rated using assessment metrics on a well-knows Likert scale from 1 to 5.

Table 1: Description and Ratings for 7 key factors that MAF uses

Factors	Agile	Waterfall	Hybrid
OUTCOMES[1]	Low rating (1: Somewhat Important – 2: Important)	High rating (4: Critical - 5: Absolutely critical),	A middle-of-the-road rating
SCOPE[2]	A low rating (1: Very small – 2: Small)	A high rating (4: Large – 5: Very Large),	A middle-of-the-road rating
CYNEFIN[3]	Complex or Chaotic project rating of 4 or 5)	Simple and Complicated projects (rating 2 and 3)	Complex project
CONSTITUENTS[4]	Projects with few constituents and only a few basic roles (e.g.: Inform, Consult or Educate) (rating of 1 or 2)	Projects with a very large number of constituents and an extended number of roles (e.g., Develop, Engage, Involve or Approve) (ratings of 4 or 5)	A middle-of-the-road rating
PRINCIPLES[5]	A high rating (4: 80% support of agile traits – 5: 100% support of agile traits),	A low rating (1: 20 % support of agile traits – 2: 40% support of agile traits)	A middle-of-the-road rating (3: 60% support of agile traits)
TEAM[6]	Team with self assessment values 3: Proficient, 4: Advanced, or 5: Expert	Team with self assessment values 1: Limited or 2: Beginner	
ORGANIZATION[7]	Organizational capacity and maturity values 3: Established, 4: Predictable, or 5: Optimizing	When the assessment is 1: Performed or 2: Managed	

[1]**Outcomes being addressed by the project (OUTCOMES):** The number and criticality of outcomes have bearing on the development method selected. Projects with many important, substantial or critical outcomes will generally be better served by a formal method like Waterfall as the end product, A low rating (1: Somewhat Important – 2: Important) will tend to indicate that Agile would work fine. A high rating (4: Critical - 5: Absolutely critical), on the other hand, will tend to indicate that a formal methodology like Waterfall should be used. A middle-of-the-road rating would likely lead to a method that includes elements of both Agile and Waterfall. See Section 3.1 Outcomes – How Critical? for more information.

[2]**Scope /features of the project (SCOPE):** The "larger" a project is (with lots of features), the more we need coordinated action by many teams. Waterfall methodology will fit better for large projects as we would need a more structured, plan-driven way that provides a certain level of control. Agile is more applicable in case of small or medium sized projects involving small group of members. When you look at scale, the distribution of teams and the organization politics that would be involved in large projects, brings into question the four bases of agile manifesto. So, using agile would not be a great idea for large projects. A low rating (1: Very small – 2: Small) will tend to indicate that agile would work better for small to medium size projects. A high rating (4: Large – 5: Very Large), on the other hand, will tend to indicate that a formal methodology like Waterfall should be used as traditional methods work better for large size projects. in A middle-of-the-road rating would likely lead to a method that includes elements of both Agile and Waterfall. See Section 3.2 Scope of the project for more information.

[3]**Nature and complexity of the project (CYNEFIN):** The nature of the project (simple, complicated, complex, chaotic) has a bearing upon the method to be used. For Simple and Complicated projects with the requirements, users, processes are known (ratings of 0 and 1 respectively), Waterfall will work best When a project is characterized as Complex or Chaotic (rating of 4 or 5), where there is a lot of uncertainty, complexity, disorder and even chaos, an iterative rapid application development method like Agile will have the highest probability of delivering, results / product. Waterfall projects in such an environment will be unlikely to succeed! With regards to Complex project (rating of 3) it is also likely that a mix of Iterative and formal method would also work as these types of projects usually involved many sub or associated projects governed and delivered outside of the core project team. See Section 3.3 CYNEFIN of project for more information.

[4]**Extent /enrollment of the project's constituent base (CONSTITUENTS):** The best system will be a failure if stakeholders, partners, and service providers do not or cannot support it. Clearly identifying and managing all constituents at each phase of a project will make or break a system project. Projects with few constituents (e.g.: executive sponsor, project director and a small programmer team) and only a few basic roles (e.g.: Inform, Consult or Educate) (rating of 1 or 2) will work best with Agile type methods, providing that the team is experienced. Projects with a very large number of constituents and an extended number of roles (e.g., Develop, Engage, Involve or Approve) (ratings of 4 or 5) will require more planning and a capacity to present what the result will be and, consequently, would require a more formal approach like Waterfall. As for other factors, a middle-of-the-road result could work best with a hybrid approach (a bit of this and a bit of that). See 3.4 Constituents of the project for more information.

[5]**Applicability of Agile principles to the project (AGILE PRINCIPLES):** Time has proven that the twelve Agile principles (Early and continuous delivery of s/w, embrace change, frequent delivery, business and developers together, motivated individuals, face to face conversation, working software, sustainable development with constant pace, technical excellence, simplicity, self-organizing teams, regular reflection and adjustment) are as important as the method itself. In my judgment that if there is little adherence to the Agile principles, there is a strong probability that the project would be better served by a formal method like Waterfall. A low rating (0 : 20 % support of agile traits – 1: 40% support of agile traits) will tend to indicate that Agile would not work fine. A high rating (3: 80% support of agile traits – 4: 100% support of agile traits), on the other hand, will tend to indicate that an Agile method is likely appropriate. A middle-of-the-road rating (2: 60% support of agile traits) road rating would likely lead to a method that includes elements of both Agile and Waterfall could work best. See Section 3.5 Applicability of Agile Principles for more information.

[6]**Team expertise and experience (TEAM):** Having a team that is experienced and

knowledgeable in IM/IT and business aspects is the key to project's success. Agile team members should be more "experienced" than those of traditional/waterfall methods because in agile the team is given an "open book" and must rely on its own capacities. Formal methods are fully documented and supported by software engineering practices and can, therefore, be applied by less-experienced people. They can "follow the recipe" much more so than in the case of an Agile method. When the team is strong and experienced enough, any chosen SDM will be more likely to be successfully applied. However, to follow an agile method, the team needs to be strong enough. When the team self-assessment values are 3: Proficient, 4: Advanced, or 5: Expert, the team is clearly strong enough to use and apply an Agile methodology. When the assessment is 1: Limited or 2: Beginner, formal methods might work better … if they can get some external help regarding applying and managing the formal method. See Section 3.6 Team expertise for more information.

[7]**Maturity of the organizations involved on the project (ORGANIZATION):** This is about organizational capacity and maturity. Developing systems in an organization that operate in an ad-hoc inconsistent manner and successfully delivering system projects will be more difficult. In our experience, when such is the case, a formal method will work better as, by itself, it provides a structure…… follow the recipe, and you will get a product! At the other end, highly mature and capable organizations can handle any type of method, especially Agile as they can easily provide Agile teams with the experience or resources they may not have. When the organizational capacity and maturity values are 2: Established, 3: Predictable, or 4: Optimizing, the organization is likely strong enough to deal with Agile method. When the assessment is 0: Performed or 1: Managed, formal methods might work better as the success is, to a large degree, dependent on following a detailed methodology. See Section 3.7 Organization's maturity for more information.

Each of the seven dimensions that the MAF is based on (Outcomes, Scope, CYNEFIN, Constituents, Principles, Team and Organization), the way they can assessed using the self-evaluation tools and the assessment metrics are discussed next.

3.1 Outcomes – How Critical?

By design, a "system" uses resources as inputs and turns them into outputs to make an organization hopefully more efficient. It is important to highlight the difference between outcomes and outputs. Allen makes the distinction between outputs and outcomes by stating that "Outputs are the goods and services that result from activities. Outcomes are the constructive impacts on people or environments" [19]. When the Chief Executive officer of a company asks his/her Chief Information Officer how a specific project will contribute towards the attainment of the organization's mandate, he/she means how will this project explicitly contribute to the strategic outcomes of the organization. If the CIO has no answer or can only say that it will save some time and money, he may not be listened to for a very long time or paid any money.

In this paper, the goal is to select a suitable SDM methodology for a given project. The degree of fit between the SDM and the project will be demonstrated through the outcome improvements, in terms of social, economic and organizational capacities, such as learning, understanding, benefits and economic changes [5, 19]. According to Saunders, "the means of achieving the desired outcomes in real life situations could be provided by the use of the suitable systems methodology" [20].

The assessment grid shown in Table 4: Evaluation Grid for Factor 1: Outcomes aims to assess the importance of speed, innovation, reliability, security, and efficiency aspects of the project outcomes. The authors argue that the more critical the outcomes associated to the project the more important it is to have solid and proven system development methodology and that the right set of constituents be part to the project. Therefore, the higher a project scores on this dimension (Level 3 or 4), the more likely that a solid, plan-driven methodology will be required.

3.2 Scope of the project

The Project Management Body of Knowledge (PMBOK), defines scope as the "sum of products and services to be provided as a project" to ensure that the project includes all the work required, and only the work required, for completing the project successfully [21]. Scope of the project is related to the complexity profile of the project and it impacts the project constituents as there is a need to develop and maintain a common understanding of what products or services the project will deliver [22].

"The scope of the project is one of the key software process determinants"[8]. It relates to the

project size, competencies and experience it will require, number of organizational divisions to be involved and the environmental forces (political, economic, social, technological, environmental, legal) that the project will be impacted by. Therefore, it is almost a given that the scope of a given project will play a role in selecting the proper methodology such as agile or else. At the extremes of the spectrum lies a simple software update project and the development of a new set of applications with significant and complex linkages. In the former case, Agile might be the method of choice while in the latter, a more formal methodology should likely be used. As Griffiths states "While an agile method can work well on life-critical systems, it takes much more skill and effort to implement. Agile is much easier to use on small, non-life-critical applications" [7]. The projects that seem to benefit most from an agile approach are the ones that develop a new system for a totally new environment [16].

By answering each of the 15 questions captured in the Table 5: Evaluation Grid for Factor 2: Scope, you will identify the base score for each element. Next, you choose an appropriate weight factor for each element and then multiply the base score of each element by its weight factor to obtain a final score of the element. When all the final scores are added up, this will provide the scope assessment for the project. The assessment grid shown in Table 6: Assessment Grid for Factor 2: Scope provides you with an assessment grid to help define the scope of your project. A low score (0 or 1) for the scope assessment would indicate that an agile methodology would be better while a high score (3 or 4) would indicate that a formal methodology like waterfall should be used for the project.

3.3 CYNEFIN of project

Project managers need to deal with the complexities of projects in practice in order to "improve the likelihood of project success or at least to understand the reasons for failure" [23]. However, the assessment of complexity is subjective and will be influenced by how the project manager perceives and responds to it [24]. The complexity level of the project will influence the decision of what kind of methodology would give us the best chance of success with this project [23].

CYNEFIN (a Welsh word meaning habitat) is a conceptual framework that was developed by Dave Snowden in 1999 to help decision making based on the project complexity level [25]. It has evolved over the years, but the project habitat (CYNEFIN) has not changed much. We find that the "simple – complicated – complex – chaotic – disorder" nomenclature can easily be used to assess a project environment. The "CYNEFIN picture" of a project will, with other factors, ensure the selection of the right development methodology.

In this research, CYNEFIN was chosen as the framework to assess the "nature" and the complexity of the project. Knowing the complexity of a given project will help us determine the right system development methodology (traditional, hybrid, agile) to be used to reduce the complexity and to achieve desired project outcomes [23]. There are, of course, other ways and methods to assess the nature of the project and there will always be people arguing the pros and cons of a specific method. The key point to keep in mind is that none is perfect, and their goal is simply to *help* you assess the complexity level of the project.

According to CYNEFIN, a project will be born in the "Disorder" state and go through the "Chaotic", the "Complex", the "Complicated" and end in "Simple" [23]. In traditional methods, stakeholders agree on deliverables, create a WBS, make schedules and execute the plan. Whereas in agile methods we focus on reducing complexity one sprint at a time [23]. So, the waterfall is not suitable if requirements are not well-understood/defined or likely to change during the course of the project [16]. According to Apke, "most software development is complex and that is the reason that agile works well and is generally preferable to waterfall. Those projects that might benefit from Waterfall are those that are complicated, those where all the answers can be known up front and experts are effective" [26].

While agile can be used in "simple", "complicated" and "complex" projects, it works best for "complex" projects, which have some uncertainty around both requirements and technology but not so much that they are chaotic or impossible to get our hands around [18]. According to Griffiths, the "simple" and "complicated" projects can benefit from the benefits of increased collaboration, communication, and visibility aspects of the agile methods, but these kinds of projects can also be run with a traditional approach. "Complex" projects, on the other hand, become a struggle if team tries to use traditional methods. According to Mikkelsen, as we can foresee the future in the "complicated" domain, the waterfall model would be a better choice, whereas in the "complex" domain, a better

choice would be an agile approach with flexibility and adaptability [23]. Boehm & Turner assert that agile is more suitable for "trivial applications where failure of the system results in a loss of convenience, such as losing personal time if a video game crashes or losing work time if a word processor fails [10]. However, for mission-critical or life-critical applications agile would be less applicable.

Table 7: Evaluation Grid for Factor 3: CYNEFIN provides elements, which will help you to assess the CYNEFIN "habitat" of your project to determine if it is going to be a *simple*, *complicated*, *complex*, or *chaotic* undertaking. For *simple* and *complicated* projects (ratings of 0 and 1) waterfall methods should be preferred while for the *complex* or *chaotic* projects (ratings of 3 or 4), agile would be better suited.

3.4 Constituents of the project

Public and private sectors program and project management best practices clearly point to the people aspects of a project as the key elements of success.

It is important to highlight the difference between constituents and stakeholders. Why talk about constituents and not stakeholders? While most IM/IT executives used the term stakeholders as an all-encompassing term, it is not. Stakeholders are those that have a stake in the outcome of the project, and they are affected *indirectly* by the project outcomes. While constituents are those whose voice matter the most, they are the active partners, co-creators who play a direct role in the project and who are *directly* affected by the project outcomes [27].

Constituent relationship management is a structured approach to manage who; what; how; for what purpose of interactions between the project team and the project players. In other words, it refers to all relationships associated with all aspects of a project. To deliver the project outcomes that reflect the actual needs of the constituents, we need to work closely and collaboratively with them to identify what they think they want, produce something which reflects that understanding, get feedback from them, and then update our solution to reflect our improved understanding [28]. Collaboration and knowledge sharing are crucial both in plan-driven/traditional and agile methodologies. However, in agile knowledge sharing active collaboration and communication are viewed as the key components. In fact, the Agile manifesto [29] highlights active collaboration in both its 3rd value ("Customer collaboration over contract negotiation") and 4th principle ("Business people and developers must work together daily throughout the project"). In agile, constituents need to communicate frequently to ensure that everyone is on the same page and kept up to date, since many project failures can be traced back to a failure of communication [18].

There is a distinction between various types of constituents as each has unique attributes, such as advisor(s), business owner(s), client(s), community of interest, executive sponsor(s), partner(s), project enabler(s), service provider(s), stakeholder(s), public, development team. Relationship Management also makes the distinction between the depths of relationships for any given constituent. While the difference between various terms may appear to be only semantic, the differences are real and have an impact on defining and agreeing on who does what and how, and the amount of resources needed to carry out an activity.

- **Approve**: To officially agree to or accept as meeting requirements
- **Consult**: To seek the views of persons or groups of persons on matters affecting them
- **Develop**: To bring to existence or to make more mature a process or system
- **Educate**: To transfer knowledge, skills and habits from an individual/group to others through teaching, training or research
- **Engage**: To reach out to provide selected groups of people with the opportunity to influence the decision-making process as well as the project outcome, objectives, deliverables, design and implementation
- **Inform**: To exchange thoughts, messages or data and/or information by speech, visuals, writing or behavior
- **Involve**: To be included to contribute in the project due to their specific knowledge, competencies and abilities.

It is important to note that the type and depth of relationship varies from constituent to constituent and often from activity to activity. For example, you could simply inform the Community of Interest that training will take place, or you could involve them in developing the training material. Having a clear agreement on who does what, when, with whom and, for what purpose is fundamental to project management relationship and ultimately to project success.

As the number of constituents involved in a project increases, the complexity of the project increases because understanding, managing, and leveraging the relationships between the constituents increases exponentially due to the $n^*(n-1)/2$ formula that calculates the number of communication channels between relationships. Agile development methodology would be more suitable for small and medium size projects with few constituents and only a few basic rules, whereas the large size projects that involve large number of constituents and extended number of roles will benefit more from plan-driven/traditional development methodologies.

The biggest limitation of agile methodologies is how they handle larger development teams. Cockburn and Highsmith both argue that "Agile development is more difficult for larger teams…as size grows coordinating interfaces becomes a dominant issue"[9]. Boehm &Turner also agrees that "teams of less than ten are a great fit for agile approaches as they can communicate face to face, support tacit knowledge by conversations and facilitate simple, visible tracking systems. As team sizes grow, supporting these agile principles requires additional techniques. It can be done but it takes more work and skill" [10].

De Lucia & Qusef also agreed that agile works well for small to medium sized team and he argued that the smaller the agile team, the higher the chances of the project success. Because when the team size grows, communications and requirement changes becomes more difficult and complex [30]. Both Constantine and Martin Fowler also believe that agile with face-to-face communication breaks down and becomes more difficult and complex with development team size that exceed 20. In contrast, plan-driven, traditional methods scale better to large projects with large number of constituents with extended number of roles.

To help assess the depth and scope of the constituents' management, the following, simple evaluation grid has been proposed. The constituents were grouped by affinities to make the assessment faster. With regards to constituent management, there are many methods, processes and management tools widely used. RASIC (Responsible, Approve, Support, Inform, Consult) is one of the best known but there are many others like PARIS, PACSI, RASCI, RASI, RACIQ, and many others[31]. The approach used in this research is unique, but you could replace it with another one that you are more familiar with.

Use the evaluation grid shown in Table 8: Evaluation Grid for Factor 4: Constituents to calculate the grand total score for this element.

Table 9: Assessment Grid for Factor 4: Constituents provides you with the range score that will be used to graphically plot this element. The lower assessment score and the in-range level (0 or 1) would indicate that an agile methodology would work best for the project while the higher score and the in-range level (3 or 4) would mean a traditional/plan-driven methodology should be used.

3.5 Applicability of Agile Principles

This indicator characterizes the agile traits based on the set of twelve principles provided by the Agile Manifesto [32] and examines the support of agile values and principles. This criterion intends to evaluate the degree of agility because there exists certain characteristics that are inherently associated with agile methodologies which can be used as evaluation criteria [6]. As part of the MAF framework, each one of these principles must be reviewed to assess its desirability or applicability to the project, organization, and culture.

Use the evaluation grid shown in Table 10: Evaluation Grid for Factor 5: Agile Principles to calculate the grand total for this principle. Next, use the assessment grid shown in Table 11: Assessment Grid for Factor 5: Agile Principles to identify the In-Range Level. A high rating (3 or 4) would indicate that an agile methodology is better suited for the project, while a low rating (0 or 1) would indicate a plan-driven methodology like waterfall should be chosen.

3.6 Team expertise

The success of agile depends on highly motivated and skilled people because documentation is very lightweight and most of the knowledge is tacit [33]. Actual implementation is left to the developers who work as self-organizing teams, without providing clear guidance and details on what needs to be done. Boehm & Turner suggests "a critical mass of highly talented people" as one of their five critical factors which can be used to determine the suitability of agile or traditional methods for a particular project [10].

This study has identified the following as the key elements of team expertise:

- Knowledge of and experience in IM/IT

- Knowledge of and experience in Project Management
- Knowledge of and experience in system development methodologies
- Knowledge of and experience in project contracting and management
- Capacity to work in a team environment
- Capacity to work under stress
- Capacity to communicate orally and in writing

The self-assessment grid that is shown in Table 12: Assessment Grid for the Evaluation for Factor 6: Team provides the definitions and their associated score value for *Limited*, *Beginner*, *Proficient*, *Advanced* and *Expert* level team members. When the team self-assessment scores are 2, 3, 4, that means that the team is strong and experienced. The stronger and the more experienced the team is the chosen SDM would be more likely to be successfully used and applied. According to Alistair Cockburn [10], agile development demands experienced team members, who can "revise a method to fit an unprecedented new situation" and "tailor a method to fit an unprecedented new situation", perhaps because the tacit nature of information flow demands a higher level of expertise. According to Boehm & Turner [7], agile projects are more likely to go smoothly with a low proportion of beginner developers and high proportion of proficient, advanced and expert level practitioners. If the development team has a higher percentage of beginners then a more traditional approach may be more successful [7]. Therefore, we claim that to follow an agile methodology successfully, the team has to score at least 2.

3.7 Organization's maturity

This indicator aims to measure the capability of an organization to provide the supporting environment conducive to the implementation of an SDM. Kerzner [34] defines the maturity as "the implementation of a standard methodology and accompanying processes such that there exists a high likelihood of repeated successes". According to Kerzner [34] "maturity implies that proper foundation of tools, techniques, processes and even culture, exists".

Mullaly [35] suggests that "the assessment of organizational capabilities is a core dimension of organizational learning and improvement". Assessment of the current capability/maturity level of an organization and its software development and delivery process will provide an indication for whether an agile or traditional or hybrid SDM would be more suitable.

By using maturity models, organizations can carry out an assessment to determine their current maturity level and the list of things they need to work on to improve. According to Fowler, a maturity model is "a tool that helps people assess the current effectiveness of a person or group and supports figuring out what skills and capabilities they need to acquire next in order to improve their performance" [36].

There are several maturity models described in the project management literature that can be used to assess and improve an organization's maturity level such as CMMI (Capability Maturity Model Integration), ISO 9001 and ISO 15504. Most of these models are rooted conceptually on the five-level project management maturity model: the Capability Maturity Model (CMM) developed by the Software Engineering Institute (SEI) of Carnegie Mellon between 1986 and 1993. CMM defines five maturity levels: Initial, Managed, Defined, Quantitatively managed, Optimized. Since then, around 30 different models have been developed each addressing a specific business model or industry context.

Though the CMM model "comes from the field of software development, it is also used as a model to aid in business processes generally, and has furthermore been used extensively worldwide in government offices, commerce, and industry"[37]. However, CMM is considered to be more associated with a document-heavy, plan-driven culture, which is against the nature of agile software development [36].

Organizational and Software-Development Capability Maturity models are two of the many disciplines that have evolved from the original model. It is important to consider both aspects of maturity when assessing an organization's capacity to positively manage all aspects of application's development.

Maturity Models have, in general, five stages and use a similar nomenclature. This research uses the maturity model that was developed by Stanford's Linear Accelerator Center Laboratory, which evaluates maturity of an organization against 3 dimensions called the "golden triangle" (People, Process, Technology) to determine the maturity level of an organization against the five sequence of stages that define a path from the lowest

(Performed) to the highest state to maturity (Optimizing), as explained in Table 2: Five stages of Organizational Maturity.

Table 2: Five stages of Organizational Maturity

Stages of Maturity	Definition		
	People	**Process**	**Technology**
Performed	Success depends on individual heroics - "Firefighting is a way of life." Relationships between disciplines are uncoordinated, perhaps even adversarial.	Unpredictable process that is poorly controlled and reactive.	Despite security issues, no controls exist.
Managed	Success depends on individuals and management system supports. Commitments are understood and managed. People are trained.	Project process is characterized but is often reactive.	Some controls in development with limited documentation.
Established	Project groups work together, perhaps as an integrated product team. Training is planned and provided according to roles.	Characterized process for the organization that is proactive.	More controls documented and developed, but over-reliant on individual efforts.
Predictable	A strong sense of teamwork exists within each project.	Process measured and controlled.	Controls monitored, measured for compliance but uneven levels of automation.
Optimizing	A strong sense of teamwork exists across the organization AND everyone is involved in process improvement	Process improvement focus	Controls more comprehensively implemented, automated, subject to continuous improvement.

The *People* dimension covers the resources and capacity principles examining both the individual capabilities such as education, training, and skills, as well as the organization capabilities such as culture, policy, strategy.

The *Process* dimension covers the methodological aspects, such as the existence and utilization of standards, guidelines, best principles, and quality management processes.

The *Technology* dimension analyzes the supporting technology infrastructure, tools, platforms, systems and services that are used in the organization.

The Information Management (IM) and Information Technology (IT) domain includes five key disciplines: Business/Process, Security, Information, Development, Operations.

Assessing the maturity of an organization for each of these five disciplines can provide an indication of the organization's readiness to successfully implement agile principles and system development methodologies. The higher the maturity level of an organization in the IM/IT domains listed above, the higher the capability of an organization to provide supporting environment conducive to the implementation of an SDM. In our experience, developing systems in organizations that operate in the low level of maturity will be more difficult. So, in that case, a plan-driven SDM would work better as by itself would provide a structure for the project. At the other end, highly mature and capable organisations can handle any type of method, especially Agile as they can easily provide Agile teams with the experience or resources they may not have.

Agile methodologies encourage the continual improvement of the software delivery process. The principle #12 of the agile manifesto states "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly" [32].

This paper proposes that decision makers use their judgment to assess the maturity of each discipline by using the evaluation grid shown in Table 13: Evaluation Grid for the Evaluation of Factor 7: Organizational Maturity and tally up the results. Next, the assessment grid shown in Table 14: Assessment Grid for Factor 7: Organizational Maturityshould be used to determine the In-Range level for the maturity level of the organization.

4. RESULT AND DISCUSSION

Throughout this article, we explained how the ratings for each of the seven decision elements of the MAF framework can be derived. To demonstrate the use of the MAF framework, we present below the results of a project landscape assessment.

Since no project is the same, patterns will differ. We will present a practical application of this framework as a case study which should demonstrate that it provides a structure and a basis for analysis of influential aspect of projects in terms of seven dimensions to determine the best suited SDM.

4.1 Case Study (an EDA tool development project in a university setting)

Background:

A new EDA tool was planned to be developed to migrate any circuit design and porting flow with minimum human intervention and replicate an existing analog/mixed-signal circuit design (reference design). This tool was developed by a team of five PhD students, using JAVA and based on a proprietary algorithm (similar to the backtracking/shrinkage algorithm).

Analysis:

The project was assessed by the project team along seven factors of the MAF framework by using the self-evaluation tools that are shown in the ANNEX: SELF-EVALUATION TOOLS and the scores were plotted on the radar chart to determine whether an agile, plan-driven or hybrid SDM would be more suitable for the project.

The evaluation results for each factor are demonstrated using a scale from 1 to 5, in Table 3: The MAF evaluation results for the EDA tool development project below:

Table 3: The MAF evaluation results for the EDA tool development project

MAF DIMENSIONS	Outcomes – How Critical?	2: Important
	Scope of the project	2: Small with low complexity
	Constituents of the project	1: Few constituents with basic role (*Inform*, *Consult* or *Educate)*
	CYNEFIN of project	4: Complex
	Organization's maturity	3: Established
	Team expertise	4: Advanced
	Applicability of Agile Principles	5: 100% supports agile values and principles

Outcomes of the project were assessed across five aspects: the *speed*, *reliability* and the *efficiency* aspects of the project outcome were important, while the *innovation* aspect was important but not critical and the *security* aspect is not applicable.

The scope of the project was assessed as *small with low complexity*.

This was a small size project with *few constituents, most of which has basic influence* (Inform, Consult or Educate).

For projects that are complex in nature with a lot of uncertainty, an iterative rapid application development method like Agile will have the better probability of success than a waterfall. According to the CYNEFIN framework, the project was considered a *complex* project because it requires a safe environment for experimentation, through which we could probe (explore) and sense (inspect) to create emergent solutions for the development.

The organizational environment within which the project was developed was mature enough at the *Established* level. This highly mature and capable organization will adequately provide the supporting environment and business processes to successfully develop and deliver the project.

The project team was composed of members that were *advanced* in their fields of expertise. However, they were not yet at a level of expert. As this was a research project which aimed to introduce a new software tool, some improvement or help was required in some areas.

When there is high adherence to the Agile principles, there is a strong probability that the project would be better served by an adaptive, agile method. This project fully (100%) supported the agile principles, which means that it was highly suitable for an agile methodology.

As demonstrated in Table 3: The MAF evaluation results for the EDA tool development project, the project scored very low on *Outcomes*, *Scope* and

Constituents dimensions while reasonably high on the *CYNEFIN*, *Organization*, *Team* dimensions and very high on *Agile Principles* dimension.

The scores for each of the seven MAF dimensions were plotted on the radar chart in Figure 3: The MAF evaluation of the EDA tool development project using Radar Chart.

As the scores towards the center indicate a good fit for an agile approach, while scores towards the outside indicate a better fit for a more traditional approach, MAF suggests that an agile methodology is best suited for this project. According to the MAF, projects that have low rating in Outcomes, Scope and Constituents factors, tend to perform better with the use of Agile methodology. In other words, the smaller size projects that involve few constituents that do not have critical substantial outcomes tend to be a good fit for agile use.

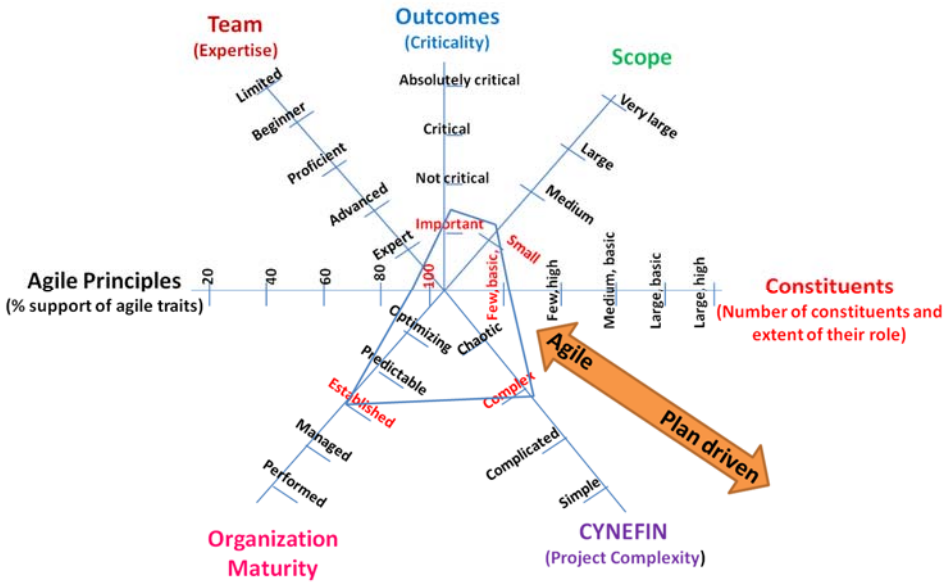

Figure 3: The MAF evaluation of the EDA tool development project using Radar Chart

Result:

The tool was successfully developed by using an agile software development methodology. The software development was based on an iterative life cycle and in one iteration cycle, an increment of the software was analyzed, designed, built, tested and integrated on a continuous basis until the full solution was realized. At the end of each iteration, a subset of functionality was delivered to stakeholders for review and the software was built incrementally, piece by piece.

5. CONCLUSION

The topic of "Which system development methodology (SDM) should we use?" is one of the first decisions faced for project implementations. Upon systematically reviewing the academic literature on available SDMs, we have found that existing SDM evaluation frameworks and comparison tools do not satisfy all the needs of project managers. This paper introduced a new, comprehensive evaluation framework called MAF to help decide on the best suited SDM for a given project. This framework identified 7 elements that

contribute to the choice of a suitable SDM for a project.

Project managers need to select the most appropriate SDM for their projects. A selection and implementation of an appropriate SDM is crucial as it maximizes the chance of project success [10]. Deciding on whether to use an agile or plan-driven or hybrid methodology in a project is not a rocket science but it requires honest answers to difficult questions and courage to make the right decision.

The assessment of each of these 7 elements of MAF is subjective and will be influenced by the project manager and/or the decision makers. Their "Methodology Bias" can influence their assessment and decision making [18]. How they perceive and respond to complexities is more of an individual and interactive consideration than is represented by the current literature. There will never be a perfect method as all projects are different but for the majority there is a best suited method. Success is about making the right choices.

It is important to note that theoretical fit and practical fit are often quite different. While the MAF can influence the choice and success, we need to acknowledge and finally remember that it is just a tool and it should not be a replacement for thought and dialogue with the project stakeholders. The tool should not be used in isolation but instead should be used to start conversations about the suitability of the chosen SDM and build consensus around the method of choice.

It is also important to note that even if the project is a great fit for an SDM (whether it be an agile or plan-driven) in theory, "if management or other stakeholders are against that approach, then its application carries significant risk" [18]. Also, as Griffiths stated, if the project manager just does not believe that a certain SDM would work due to their "Methodology Bias", then project obstacles can appear as vindication of process weakness, not setbacks to overcome [18]. If a team has a strong desire to succeed with an agile (or any other) approach they will likely find inventive ways to make their method work.

It would be worth exploring if MAF is a reliable framework that can be used as a precise evaluation and assessment tool. The practical application of MAF is necessary to prove itself in real life situations as a framework that provides a structure for assessing and evaluating a project to determine a suitable SDM. Upon its application, the framework elements might be subject to more analysis.

Further research might be of value to improve the MAF. Defining and using quantitative metrics for as many of the seven factors (as opposed to qualitative metrics) would make MAF a more comprehensive evaluation framework. Multiple Criteria Decision Making (MCDM) methods especially Analytic Hierarchy or Analytic Network Process (AHP/ANP) developed by Thomas Saaty [38] can be used in prioritizing the seven factors that MAF is based on. Prioritizing these factors with MCDM methodologies (AHP/ANP) could be a great contribution to the subject area.

This research would benefit from an empirical study to refine and update the proposed MAF and apply it to the assessment of various projects of different size and complexity. With empirical feedback that can be acquired from real project situations, the evaluation criterion and the metrics used can be improved in order to provide more precise results.

REFERENCES

[1] Chow, T. and D.-B. Cao, *A survey study of critical success factors in agile software projects.* Journal of Systems and Software, 2008. **81**(6): p. 961-971.
[2] Misra, S.C., V. Kumar, and U. Kumar, *Identifying some important success factors in adopting agile software development practices.* Journal of Systems and Software, 2009. **82**(11): p. 1869-1890.
[3] Micic, L. *Agile methodology selection criteria: IT start-up case study*. in *IOP Conference Series: Materials Science and Engineering*. 2017. IOP Publishing.
[4] Jones, C., *Evaluating agile and scrum with other software methodologies*, 2013, InfoQ.
[5] Mullaly, M. and J.L. Thomas, *Exploring the dynamics of value and fit: Insights from project management.* Project Management Journal, 2009. **40**(1): p. 124-135.
[6] Taromirad, M. and R. Ramsin. *Cefam: Comprehensive evaluation framework for agile methodologies*. in *Software Engineering Workshop, 2008. SEW'08. 32nd Annual IEEE*. 2008. IEEE.
[7] Griffiths, M., *Agile suitability filters.* Retrieved January, 2013.
[8] Alqudah, M.K. and R. Razali, *Key factors for selecting an Agile method: A systematic literature review.* International Journal on

Advanced Science, Engineering and Information Technology, 2017. **7**(2): p. 526-537.
[9] Cockburn, A. and J. Highsmith, *Agile software development, the people factor.* Computer, 2001. **34**(11): p. 131-133.
[10] Boehm, B. and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*2004, Boston, MA: Addison-Wesley.
[11] Geras, A., M. Smith, and J. Miller. *Configuring hybrid agile-traditional software processes*. in *International Conference on Extreme Programming and Agile Processes in Software Engineering*. 2006. Springer.
[12] Datta, S. *Agility measurement index: a metric for the crossroads of software development methodologies*. in *Proceedings of the 44th annual Southeast regional conference*. 2006.
[13] Abrahamsson, P., et al. *New directions on agile methods: a comparative analysis*. in *25th International Conference on Software Engineering, 2003. Proceedings.* 2003. Ieee.
[14] Qumer, A. and B. Henderson-Sellers. *Comparative evaluation of XP and Scrum using the 4D Analytical Tool (4-DAT)*. in *Proceedings of the European and Mediterranean conference on information systems*. 2006.
[15] Taromirad, M. and R. Ramsin. *An appraisal of existing evaluation frameworks for agile methodologies*. in *15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ecbs 2008)*. 2008. IEEE.
[16] van Casteren, W., *The Waterfall Model and Agile Methodologies: A comparison by project characteristics.* 2017.
[17] Strode, D. *Agile methods: a comparative analysis*. in *Proceedings of the 19th annual conference of the national advisory committee on computing qualifications, NACCQ*. 2006.
[18] Griffiths, M., *PMI-ACP Exam Prep*. updated 2nd ed2018: RMC Pubns Inc. . 434.
[19] Allen, W. *More about outcomes – why they are important … and elusive!* 2019.
[20] Saunders, R.G., *Project management: a systems perspective.* International Journal of Project Management, 1992. **10**(3): p. 153-159.
[21] PMI, *A guide to the Project Management Body of Knowledge - Sixth Edition*2017: Project Management Institute.
[22] Darnall, R. and J. Preston, *Project management from simple to complex*2010.
[23] Mikkelsen, M., *In search of project leadership principles for navigating the complexity of IT projects.* 2018.
[24] Geraldi, J., H. Maylor, and T. Williams, *Now, let's make it really complex (complicated).* International Journal of Operations & Production Management, 2011.
[25] Snowden, D.J. and M.E. Boone, *A leader's framework for decision making.* Harvard Business Review, 2007. **85**(11): p. 68.
[26] Apke, L. *Agile Principles: Why you need self-organizing teams*. 2020 16 August, 2016 [cited 2020 March 26, 2020]; Available from: http://www.agile-doctor.com/2016/08/16/why-you-need-self-organizing-teams/.
[27] Keystone. *Impact, Planning, Assessment and Learning(IPAL) Guide 3 Learning with Constituents*. 2018; Available from: https://keystoneaccountability.org/wp-content/uploads/files/3%20Learning%20with%20constituents_0.pdf.
[28] Ambler, S. *Active Stakeholder Participation.* 2018; Available from: http://agilemodeling.com/essays/activeStakeholderParticipation.htm.
[29] PMI, *Agile Practice Guide*. http://agilealliance.org/agile-practice-guide/2017: PMI.
[30] De Lucia, A. and A. Qusef, *Requirements engineering in agile software development.* Journal of emerging technologies in web intelligence, 2010. **2**(3): p. 212-220.
[31] Wikipedia, *Responsibility assignment matrix*, 2020.
[32] Fowler, M. and J. Highsmith, *The agile manifesto.* Software Development, 2001. **9**(8): p. 28-35.
[33] Eberlein, A. and J. Leite. *Agile requirements definition: A view from requirements engineering*. in *Proceedings of the International Workshop on Time-Constrained Requirements Engineering (TCRE'02)*. 2002.
[34] Kerzner, H., *Project management: a systems approach to planning, scheduling, and controlling*2017: John Wiley & Sons.
[35] Mullaly, M., *Longitudinal analysis of project management maturity.* Project Management Journal, 2006. **37**(3): p. 62-73.
[36] Fowler, M. *Maturity Model*. 2020 August 26, 2014 March 29, 2020]; Available from: https://martinfowler.com/bliki/MaturityModel.html.
[37] Wikipedia, *Capability Maturity Model*, in *Wikipedia*2020.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[38] 3Saaty, T.L., *Decision making—the analytic hierarchy and network processes (AHP/ANP).* Journal of systems science and systems engineering, 2004. **13**(1): p. 1-35.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ANNEX: SELF-EVALUATION TOOLS

1. SELF-EVALUATION TOOL FOR FACTOR 1: OUTCOME

Table 4: Evaluation Grid for Factor 1: Outcomes

HOW APPLICABLE ARE THESE OUTCOMES TO YOUR PROJECT?	**0 point:** not applicable	**1 point:** somewhat important	**2 points:** Important	**3 points:** important but not critical	**4 points:** absolutely critical
FAST: the faster we reach new markets, the higher the probability that we will grow or bottom line and capture market share.					
INNOVATIVE: Innovative products are critical to the on-going success of our company.					
RELIABLE: availability 365/12/7/24 without interruptions at the expected levels of performance secures our market's position.					
SECURE: Securing client data and information from unauthorized access and use has a direct linkage on the value of our shares and assets.					
EFFICIENT: To maintain our profitability, we must reduce our cost to the minimum without risking major financial issues.					
TOTAL SCORE	**TBD**				

Note 1: For each outcome, select the appropriate rating in Table 4.

Note 2: Total all points and use the following scale to identify the In-Range Level.

- 0 to 4 points = level 1 (Somewhat important)
- 5 to 9 points = level 2 (Important)
- 10 to 13 points = level 3 (Important but not critical)
- 14 to 17 points = level 4 (Critical)
- 18 to 20 points = level 5 (Absolutely critical)

2. SELF-EVALUATION TOOL FOR FACTOR 2: SCOPE

Table 5: Evaluation Grid for Factor 2: Scope

ID	ELEMENTS	QUESTIONS	BASE SCORE (A) (0, 1, 2, 3, 4)	WEIGHT FACTOR (B) (1 , 2 , 4)	TOTAL SCORE (Ax B)
1	**Competencies**	Does the project involve many IM/IT competencies, such as architects, telecom experts, programmers, other(s)			
2	**Data elements**	Does the system include the management of a very large number of data elements, datasets, data interfaces, data marts and warehouses?			
3	**Dependencies**	Is this project dependent/linked to other projects, databases, and IT infrastructures?			
4	**Environment**	Will the project be carried out in difficult work conditions such as poor accommodations, extra-long work hours, health and safety risks, etc.?			
5	**Financial**	Does the project require big investment; tight and specific financial conditions?			
6	**Geography**	Is the project team geographically dispersed? Different languages?			
7	**Leading Edge Technologies**	Does this project involve a new, leading edge technology?			
8	**Legal**	Are there legal requirements to be taken into accounts in the requirements or project as a whole?			
9	**Multi-disciplinary**	Does this project cross many fields of organizational divisions and businesses such as financial, human resources, legal or other(s)?			
10	**Nature of System Work**	Is this mostly a technology project?			
11	**Process**	Is this project the automation of a complex set of business processes and rules?			
12	**International**	Are there any countries, other than yours involved in this project (e.g.: USA, Canada, China?)			
13	**Security**	Is there a need to secure / encrypt information to limit / control access?			
14	**Technical Integration**	Does this project integrate many technologies or platforms such as cloud computing, business intelligence, other(s)? Is there a need to deploy the system on different hardware such as smart phones, tablets, laptops, PCs and operating systems such as Windows, Apple, Android?			
15	**Time**	Is the project time sensitive?			

TOTAL SCORE	TBD

Note 1: In Table 5, use the following scale to identify the base score for each question. Your score is the selected base score times the appropriate weight factor (1, 2 or 4) for the question.

- 0: Not true, appropriate or applicable to this project
- 1: To some degree true, appropriate or applicable to this project
- 2: Somewhat true, appropriate or applicable to this project
- 3: Mostly true, appropriate or applicable to this project
- 4: Totally true, appropriate or applicable to this project

Note 2: This self-evaluation grid in Table 5 uses weighted scores. The scale used (weight factor: 1 - 2 – 4) reflects and amplifies, based on our work experience, the relative influence of the various factors.

- 1: important
- 2: quite important
- 4: critically important

Note 3: Use Table 6 to determine the in-range score that will be used to determine scope size of the project.

Table 6: Assessment Grid for Factor 2: Scope

TOTAL SCORE	IN RANGE LEVEL
Less than 50 points	1 – Very Small / Very-low complexity
Less than 70 points (0-29%)	2 – Small / Low complexity
70 and 120 points (30%+)	3 – Medium /Medium complexity
120 and 190 points (55%+)	4 – Large / High complexity
More than 190 points (80%+)	5 – Very large / high complexity

3. SELF-EVALUATION TOOL FOR FACTOR 3: CYNEFIN

Table 7: Evaluation Grid for Factor 3: CYNEFIN

LEVELS	DEFINITIONS	THE LEVEL THAT DESCRIBES YOUR CYNEFIN HABITAT
DISORDER (1)	• Complete solution impossible • Attempts to develop solutions are turned down • Requirements are difficult to define • Feelings and emotions run high in the team – it is on the verge of breaking up • Sponsors are becoming uninvolved or becoming negative	
SIMPLE (2) **"Just-do-it"** Example: Cooking an omelet or a steak	• Encompasses some basic issues of technology, techniques, expertise and terminology – scaling is not an issue • Problems and answers are well known • Requirements are clear and stable • The relationship between cause and effect is obvious • There is one or a few right answers • Many similar projects delivered successfully • It can be considered as a standard practice / operation • Often compelled to use a vendor's method	
COMPLICATED (3) **"Plan it"** Example: Building an electronic printed circuit, building a house or car, banks, manufacturing public schools, healthcare providers	• Scaling is an issue but there also is a significant need for coordination, technology and/or specialized expertise • The problem is open ended with a range of solutions • Requirements are clear • Well-defined relationship between cause and effect • Focuses on the content that is knowable and therefore possible to plan • Owners and users are known and numerous • Large number of known requirements and rules • A range of possible answers • Requires significant analysis and investigation	
COMPLEX (4) **"Frame it"** Example: Stock markets, New Product Development, Innovation/Invention	• People, relationships and their properties of self-organization, interconnections and evolution are key drivers and triggers • Problems and solutions are evolving • Requirements are clear • Owners, partners, clients and users are changing • There is no known right solution • Innovation and experimentation are required • Dealing with new technology	
CHAOTIC (5) **"Survive it"** Example: Negotiating a peace treaty in the middle east	• Unpredictable behaviours, often triggered by small changes in conditions, result in a constant state of change • Unclear what the requirements are • Requirements are evolving • No answer / solution seems to satisfy everybody • Constituents are often or constantly changing • Creativity is the only possible avenue	

Note 1: Use Table 7 to identify the level that best describes your CYNEFIN habitat based on the definitions listed in the table below.

- *1: Disorder*

- *2: Simple*
- *3: Complicated*
- *4: Complex*
- *5: Chaotic*

4. SELF-EVALUATION TOOL FOR FACTOR 4: CONSTITUENTS

Table 8: Evaluation Grid for Factor 4: Constituents

ELEMENTS	NUMBER OF CONSTITUENTS (A)	WEIGHT FACTOR (B) *1: Inform* *2: Consult or educate* *4: Develop, engage, involve or approve*	TOTAL SCORE (A x B)
Client(s) / Executive Sponsor / Business owners			
Development Team(s)			
Stakeholders – Community of Interest – Public			
Project Enablers / Partners / Service providers			
TOTAL SCORE			**TBD**

Note 1: In Table 8, for each element, insert the number of constituents (but not the number of people in a constituency).

Note 2: In Table 8, for each element, determine the type of role being basic (*Inform*, *Consult* or *Educate*) or an extended role (*Develop*, *Engage*, *Involve* or *Approve*) which will be used as a weight factor. The scale used (doubling: 1 - 2 - 4) reflects and amplifies, *based on our experience*, the relative influence of the various constituents.

- *1: Inform*
- *2: Consult or educate*
- *4: Develop, engage, involve, or approve*

Note 3: Multiply the number of constituents of each element by its weight factor to obtain a final score for the element.

Use Table 9 to determine the in-range score that will be used for Factor 4: Constituents.

Table 9: Assessment Grid for Factor 4: Constituents

TOTAL SCORE	IN RANGE LEVEL
Less than 50 points	1 – Small size project, few constituents, most of which have basic influence
Less than 100 points	2 – Medium size project, few constituents, most of which have high influence
100 and 200 points	3 – Medium size project, medium size constituents, most of which have basic influence
200 and 300 points	4 – Large size project, large number of constituents, most of which have basic influence
More than 300 points	5 – Large size project, large number of constituents most of which have high influence

5. SELF-EVALUATION TOOL FOR FACTOR 5: AGILE PRINCIPLES

Table 10: Evaluation Grid for Factor 5: Agile Principles

AGILE PRINCIPLES	HOW IT DOES OR DOESN'T APPLY TO THE PROJECT	PROJECT SCORE (0, 1, 2, 3, 4)
#1: Customer satisfaction by rapid delivery of useful software.		
#2: Welcome changing requirements, even late in development.		
#3: Working software is delivered frequently (weeks rather than months).		
#4: Close daily cooperation between businesspeople and developers.		
#5: Projects are built around motivated individuals, who should be trusted.		
#6: Face-to-face conversation is the best form of communication (co-location).		
#7: Working software is the principal measure of progress.		
#8: Sustainable development, able to maintain a constant pace.		
#9: Continuous attention to technical excellence and good design.		
#10: Simplicity—the art of maximizing the amount of work not done—is essential.		
#11: Self-organizing teams.		
#12: Regular adaptation to changing circumstances.		
TOTAL SCORE		**TBD**

Note 1: to assess principles against your project, use the following scoring approach for each principle.

- 0: not true, appropriate or applicable to this project
- 1: To some degree true, appropriate or applicable to this project
- 2: Somewhat true, appropriate or applicable to this project
- 3: Mostly true, appropriate or applicable to this project
- 4: Totally true, appropriate or applicable to this project

Note 2: total up your individual scores to calculate the grand total score

Table 11: Assessment Grid for Factor 5: Agile Principles

TOTAL SCORE	IN RANGE LEVEL
0 to 9 points	**1 :** 20% supports agile values and principles
10 to 19 points	**2:** 40 % supports agile values and principles
20 to 29 points	**3:** 60 % supports agile values and principles
30 to 39 points	**4:** 80% supports agile values and principles

40 to 48 points	**5:** 100% supports agile values and principles

6. SELF-EVALUATION TOOL FOR FACTOR 6: TEAM

Table 12: Assessment Grid for the Evaluation for Factor 6: Team

RANGES AND DEFINITIONS	YOUR SCORE
LIMITED (SCORE 1): the team lacks in most if not all elements of knowledge and expertise.	
BEGINNER (SCORE 2): the team has some of the elements of knowledge and expertise but lack in many.	
PROFICIENT (SCORE 3): the team meets most knowledge and expertise elements, but help will be required to complement some elements.	
ADVANCED (SCORE 4): the team meets all elements although some improvement or help might be required in some areas.	
EXPERT (SCORE 5): the team meets all elements and is considered a model by others.	

7. SELF-EVALUATION TOOL FOR FACTOR 7: ORGANIZATIONAL MATURITY

Table 13: Evaluation Grid for the Evaluation of Factor 7: Organizational Maturity

		MATURITY LEVELS				
		Performed (0 point)	**Managed (1 point)**	**Established (2 points)**	**Predictable (3 points)**	**Optimizing (4 points)**
IM/IT DOMAINS	Business/Process					
	Security					
	Information					
	Development					
	Operations					
	TOTAL SCORE	**TBD**				

Table 14: Assessment Grid for Factor 7: Organizational Maturity

TOTAL SCORE	IN RANGE LEVEL
0 to 3 points	1 – Performed
4 to 7 points	2 – Managed
8 TO 12 points	3 – Established
13 to 16 points	4 – Predictable
17 to 20 points	5 – Optimizing