

Better Scrum through Essence

Ivar Jacobson¹ | Jeff Sutherland² | Brian Kerr³ |
Barbora Buhnova⁴

¹Ivar Jacobson International, Switzerland

²Scrum Inc., Cambridge, MA, USA

³Ivar Jacobson International, UK

⁴Masaryk University, Brno, Czech Republic

Correspondence

Ivar Jacobson, Ivar Jacobson International
Email: ivar@ivarjacobson.com

Funding information

None

We live at an exciting time where software has become a dominant aspect of our everyday life. Although software provides opportunities for improving various aspects of our society, it also presents many challenges. One of them is development, deployment and sustaining of high quality software on a broad scale. While agile methods (Scrum being one of the most prominent examples) ease the process, their popularity deteriorates the clarity and simplicity they were once meant to bring into software development. This article explores the synergy of Scrum and Essence, a domain model of software engineering processes, intending to become a common ground for software development methods, bringing clarity into the composition of methods from individual practices. This short communication motivates the interplay of Scrum and Essence, being accompanied with a set of videotutorials and 21 Scrum Essential cards to further guide more effective team's way of working.

KEYWORDS

Essence, Scrum, agile, methods, software engineering practice

1 | INTRODUCTION

Essence describes the essential elements of any software development and helps teams understand where they are, what is missing or what needs to be addressed [1, 2, 3]. Essence is method-agnostic and unites the myriad of disconnected, conflicting, costly, contradictory and transitory methods and practices that exist in most large organizations.

Essence leverages the fact that the software world needs an ecosystem from which teams could select the practices they need to succeed in their mission to develop software. And in this way it complements the works on situational method engineering, which focus on the in-house construction of an organization-specific or project-specific methodological approach [4].

In the Essence view, every method is a composition of a set of reusable 'mini-methods', or practices. Examples of such practices are user story, component-based development, test-driven design, continuous integration, pair programming, self-organizing teams. Through Essence, we envision that all useful practices should be made available in an ecosystem. As teams would use the practices, they would also provide feedback to improve them, and possibly replace them with better practices.

Today, outside Essence, there are a big number of methods in the world. However, teams can hardly mix and match practices from different methods, because practices are presented in a way specific for the method they belong to. They use specific vocabulary and structure, preventing teams from reusing practices from different methods. The practices are locked into their methods, in method prisons [5].

The first step towards our open ecosystem is a standard vocabulary of what software engineering is and a standard way of describing what is important about a method and its practices so that teams could be sure that the practices they selected are managed by an open community of competent individuals, instead of being under the control of a guru. Such a standard, a widely-agreed upon platform/foundation, a common ground, or as we say a kernel was developed over five years with participation of about a hundred method experts resulting in Essence [1, 2], in 2014 adopted by Object Management Group (OMG).

Since then, Essence has been establishing itself as a domain model of software engineering processes, intending to become a common ground for all software development methods, to be used to describe any method or practice used to develop or operate software. And the growing world of agile methods is no exception [6, 7, 8].

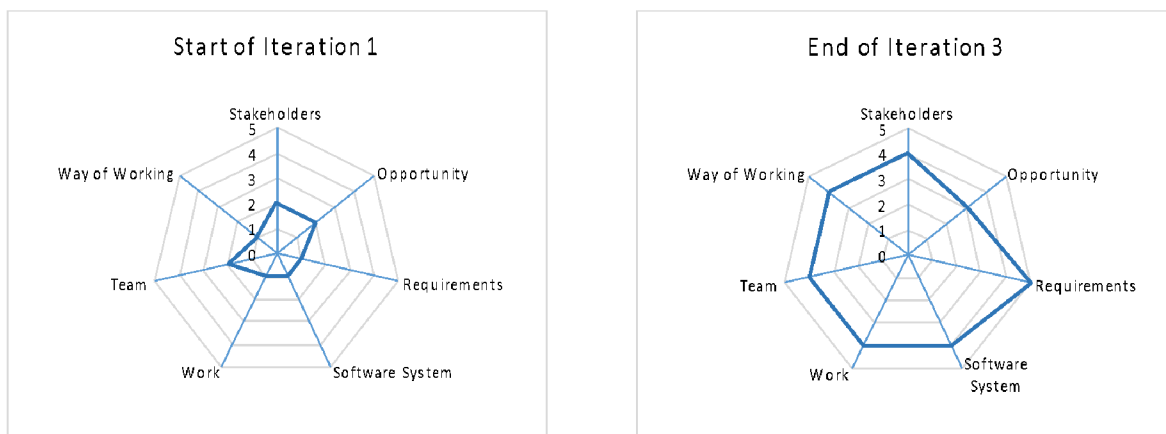


FIGURE 1 The progress of an endeavor in terms of alpha states over time (from iteration 1 to 3)

In this article, we share how Scrum can be leveraged with the help of Essence, building synergy between the two concepts. To this end, we briefly introduce the key ideas, and then demonstrate the benefits of combining Scrum with Essence, resulting in an improved way of working, planning and tracking team's progress within software development. This short communication is then complemented with a series of Better Scrum through Essence blog posts [9, 10] and videotutorials [11, 12, 13], where we have elaborated on the presented idea.

2 | THE ESSENCE OF ESSENCE

Essence is many things, dependent on who you are. For a software practitioner, Essence provides a language for defining practices with just a few different kinds of elements. The key element is the concept of an alpha, described in this section together with additional language constructs needed for proper capturing of practices.

2.1 | The seven kernel Alphas

To be successful with a software product, getting the right software system, one needs to understand the value it will provide, to whom it will provide value, and what that means in terms of requirements. One also needs to set up a project with a functioning team doing work, while providing the team with an efficient way of working. Software engineering is multi-dimensional. We identified seven dimensions, the alphas, as prevalent in any endeavor. They are defined as part of the Essence kernel.

Related to the customer

- 1) **Opportunity:** There is a problem or an Opportunity to address
- 2) **Stakeholders:** There are Stakeholders who will fund, use and benefit from the solution produced

Related to the solution

- 3) **Requirements:** There are certain Requirements to be met
- 4) **Software System:** There'll be a Software System to develop

Related to the endeavor

- 5) **Work:** We need to kick off the Work to be done
- 6) **Team:** We need an empowered Team of competent people to perform the work
- 7) **Way of Working:** The team needs a good, responsive Way of Working

Typical for an alpha is that it has clearly defined states and during its lifetime it progresses from state to state. During an iteration, many alphas, sometimes all of them, will progress from one set of states to another set of states, as shown in Figure 1. In the figure, the left radar diagram shows the states of the kernel alphas before start of iteration 1. The right diagram shows the states after iteration 3.

Such a set of states is called a wave, so the endeavor moves in waves and for each wave there must be some kind of balance in how far each alpha has progressed. For instance, if in one wave you just moved along one alpha, for instance only creating code on the Software System alpha, you may be building code that doesn't give any value, code not of interest to the stakeholders, with a team that doesn't collaborate well.

2.2 | Essence is more than the alphas

Since Essence is a standard, it clearly cannot contain specifics of practices. It does though describe additional language constructs beyond the concept of alphas, to be able to describe specific practices. The main types of item are:

1. *Alphas*, the key elements essential to an assessment of the project's progress and health
2. *Work Products*, the things that get produced like documents or code
3. *Activities*, the work to be done like Sprint Planning or a Daily Scrum
4. *Patterns*, for describing other things and their relationships, like roles or general principles

For more specific examples of Essence and its usage, see [1, 2, 12]. In the following, we elaborate on two use cases specific to the value of Essence in the context of Scrum.

3 | APPLYING ESSENCE FOR SCRUM

When Jeff Sutherland, the co-creator of Scrum, first encountered Essence, he described it as a way “to create a simple glossary of the events, roles and artifacts of the Scrum framework”, which is what was needed to more easily and concisely describe Scrum.

The first attempt of studying how Essence can power Scrum has been made in [14]. Recently, however, we have elaborated on the topic via introducing a set of 21 cards supporting better Scrum through Essence [15], together with a series of videotutorials on the topic [11, 12, 13].

Overall, development teams can benefit from the synergy thanks to improved ways of learning new practices and enhancing training events, team retrospectives improving the team's way of working, planning activities while efficiently coordinating the team and other stakeholders, agreeing on responsibilities within the team and with external stakeholders, tracking progress of the key items under development, and determining the current status of the endeavor [15]. In this section, we elaborate on two use cases of Essence support of Scrum, to demonstrate the synergy between the two.

3.1 | Use a practice

Practices and methods have been used since the early days of software engineering. Back then, they were tacit, not explicit, had no description, and during the last 50 years there have been periods when the method authors and software-engineering leaders were advocating documenting them, followed by periods when they recommended them being tacit and presented to the teams by coaches with similar traditions. Today, the leading methods for scaling agile advocate some form of descriptions motivated by consistent application when using the practices over many teams. The need for consistency in usage is one of the reasons for Essence:

1. Up until now every method author/guru has come up with his/her own way of describing practices. 20 years ago the fashion was to write lengthy descriptions, sometimes even books, on each practice. Nowadays, descriptions are brief, sometimes with very little content, but nevertheless each author has a special way of doing it. The serious problem is that you cannot mix and match practices from different methods in a simple way. With Essence we have a simple, standard way of describing practices eliminating that problem.
2. Describing too much or describing too little is not useful. With Essence the descriptions focus on what is needed to be understood to support team work. More knowledge may be needed, but knowing the essentials is the start.
3. Learning practices from papers or books has traditionally not attracted more than a small fraction of the user community. Essence provides a modern user experience based on poker-card sized cards¹ used to play serious

¹The examples shown in this paper have been based on physical cards but we have also worked on an electronic online environment for teams to share,

games which makes the learning experience naturally attractive. Using cards is not a new idea, but Essence cards have deep meaning which helps you better understand the key concepts and to play many practical games, e.g. to help the team agree on the progress within the project, guiding them to use the practices [16, 17].

Essence provides use cases for teaching, describing, modifying practices, but one of its most important use cases is Using a Practice in daily work.

Describe a Practice

Thus, most modern methods advocate some form of descriptions to ensure consistent use of practices. Essence supports doing that in a way adopted as a standard.

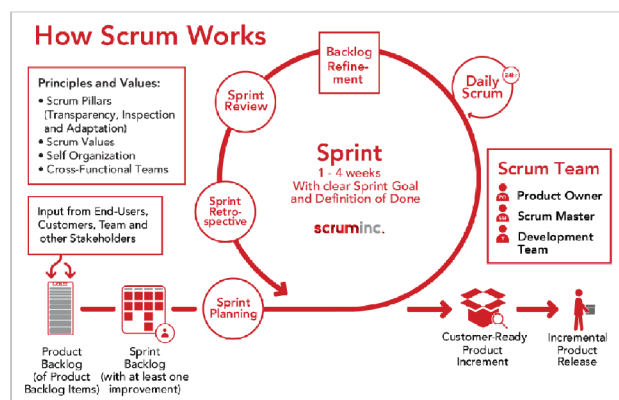


FIGURE 2 Scrum – the big picture

As of today, December 2020, Ivar Jacobson International has described around 90 practices and of those published close to 30 on its web site using Essence as a vocabulary and created cards for all essential elements [3]. We say we have essentialized these practices. One of the most well-known practices of all times is Scrum, which is described as in Figure 2 by one of its co-creators, Jeff Sutherland.

Together with Jeff Sutherland this practice has been essentialized, resulting in 21 cards representing Scrum Essentials, as shown in Figure 3.

The cards are grouped in four categories, each category being represented by its own icon. The alpha icon looks like a stylized alpha or like a fish, the work product looks like a sheet with a folded corner, the activity looks as an arrow and the pattern icon is a rectangle with a broken line through its centre.

1. The *alpha cards*, which represent the alphas the team works with during an endeavor. Scrum has three alphas: Product Backlog Item, Improvement and Sprint. Each alpha has a series of states defined, for example the Sprint alpha progresses through three states: Scheduled, Planned and Reviewed.
2. The *work product cards*, which represent the tangible elements, such as reports or documents, that you work with when using Scrum. Scrum provides five work products, one of which is the Product Backlog. Work products are related to an alpha and in this case we can see at the bottom of the card that it describes the Requirements alpha. Work products have a number of levels of detail described and in this case it has just one, that of Items Ordered. The individual items on this Product Backlog are represented with an alpha, the Product Backlog Item, as we want

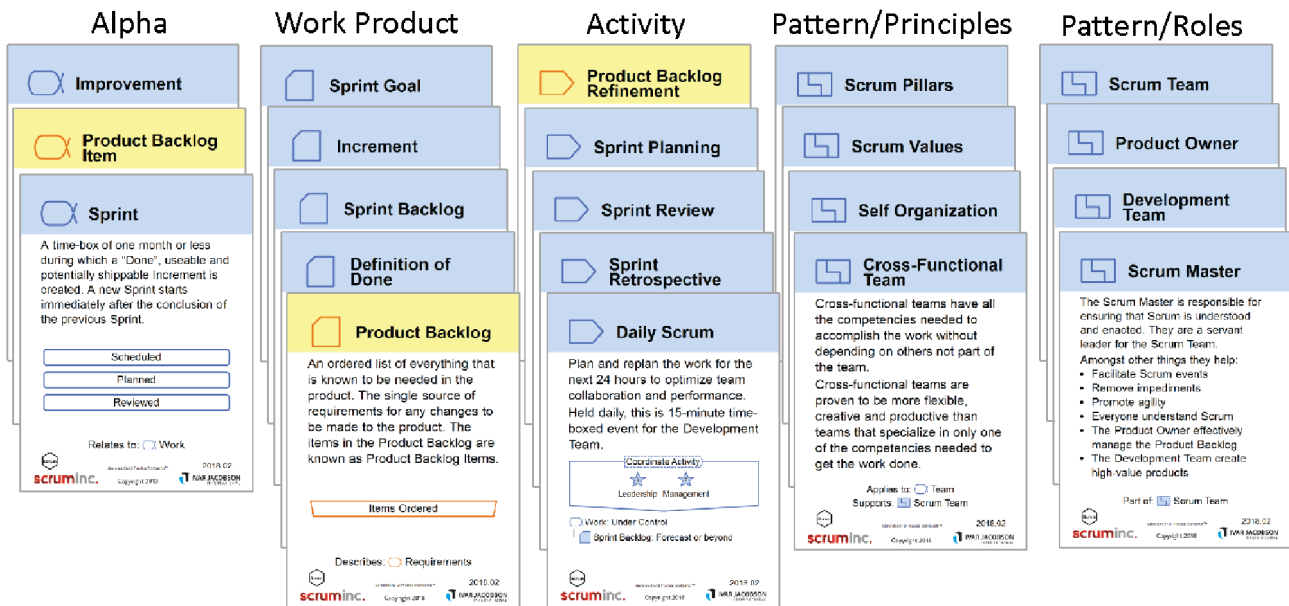


FIGURE 3 Scrum Essentials represented as cards

to progress and track them.

3. The *activity cards*, which represent activities, the things to do during an endeavor. Scrum has five activities, one of them being Daily Scrum. An activity can be defined by the alpha states or work product levels of detail that it expects before starting, and that result from performing 24 hours to optimize the activity. In this case the Daily Scrum activity ensures that the Work should be in the Under Control state, and that the Sprint Backlog should be at the Forecast level of detail or beyond. An activity card also specifies which competencies – the star icons in the activities in Figure 3 – are needed to do the job.
4. The *pattern cards* are represented by the two columns on the right in Figure 3. A pattern is a generic mechanism for describing other elements and their relationships. Scrum provides two kinds of patterns – principles, such as Cross-Functional Team, and roles, such as Scrum Master.

In his blog [9] Jeff Sutherland says: “I use the cards in many of my Scrum training courses today and they prove to be very popular with both students and training partners. The cards add exercises and interactive games to enhance the learning experience. Teams often represent their actual work items with the cards or sticky notes. These are placed on their team boards to visualize, prioritize, track and make decisions about the work. Using the cards as a means of representing key parts of Scrum gives the team the facility to work better together as a team.”

The use of the cards constantly reveals impediments. After a ‘Build Your Own Scrum’ exercise with thousands of workshop participants we find on average that teams implement only about a third of the components of Scrum well, another third are implemented poorly, and the last third are not implemented at all. It clarifies why many teams have hard, slow, and painful Scrums even though Scrum is designed to be fast, easy and fun [13].

From Descriptions to Usage

Until now descriptions of practices have for most methods just served one use case – teaching a practice. This is not the case with essentialized practices. Thanks to the alphas with states and checklists, the practice is coming alive every time you use it. The cards can for instance be played as the way to agree on the progress of the project, in

waves from states to states with engaged teams. The other kinds of cards can also be used in other games [16, 17] which we will now see in another example.

3.2 | Assess the Use of a Practice

In the old days of software engineering, playing games when developing software was not seen as a serious activity. Modern software development has gamified much of the work because of the value it gives to engage the team members and to enable them to collaborate in a better and structured way.

With every practice several useful games can be played, so playing games is a natural use case of Essence. The cards invite the team members to have serious conversations on important subjects related to their practice. There are numerous such placeholders for conversations when the cards are used to play games. For instance, the alpha state cards have checklists to discuss; every check is discussed to agree on the meaning and then if the check has been achieved or not. To further illustrate the value of games we will play one of the many practice-based games – the Practice Patience game.

An important part of Scrum is for the team to regularly inspect itself and create a plan for improvements. This team game utilises the Essence cards to identify and prioritize these improvements, and in this example we will do it for how the team uses the Scrum practice itself, but it would be equally effective for any practice.

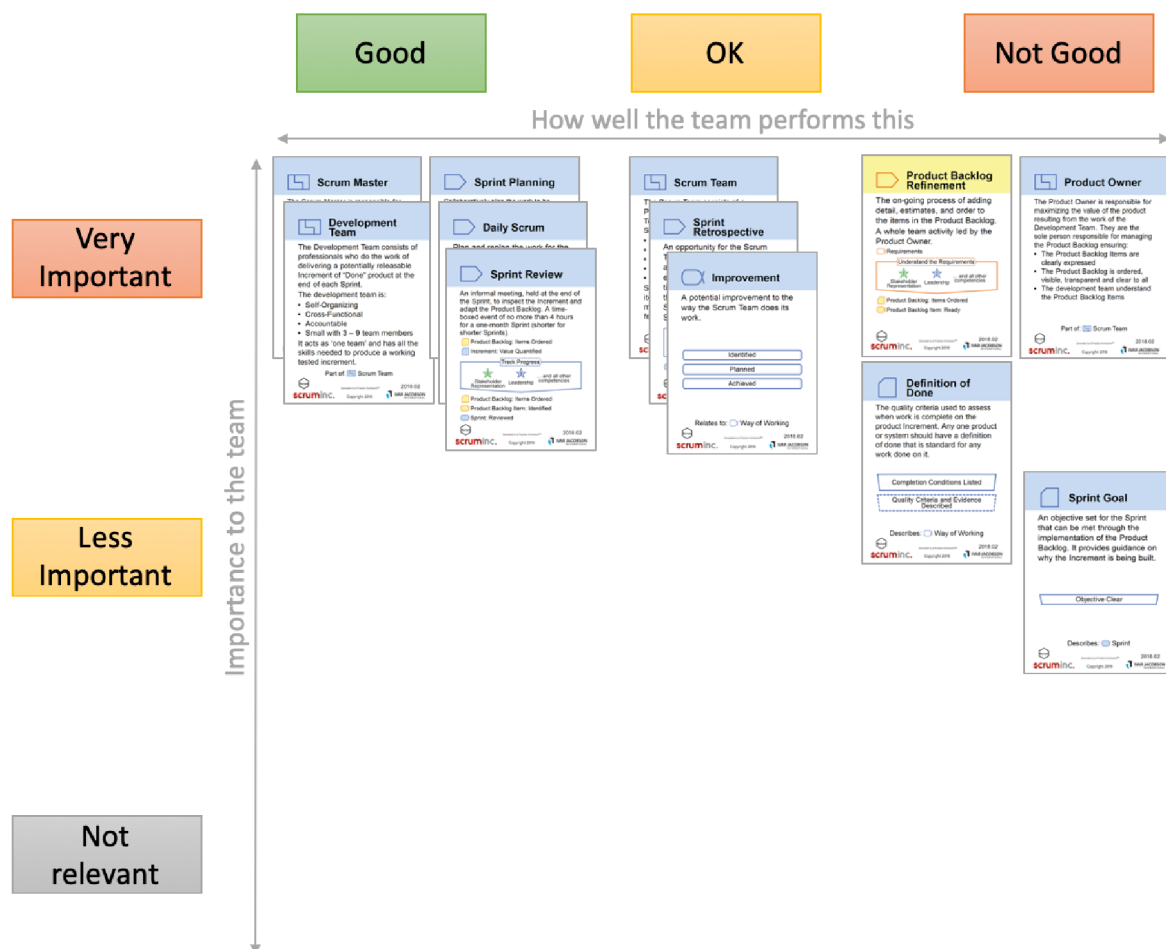


FIGURE 4 Scrum cards laid out on the Practice Patience board

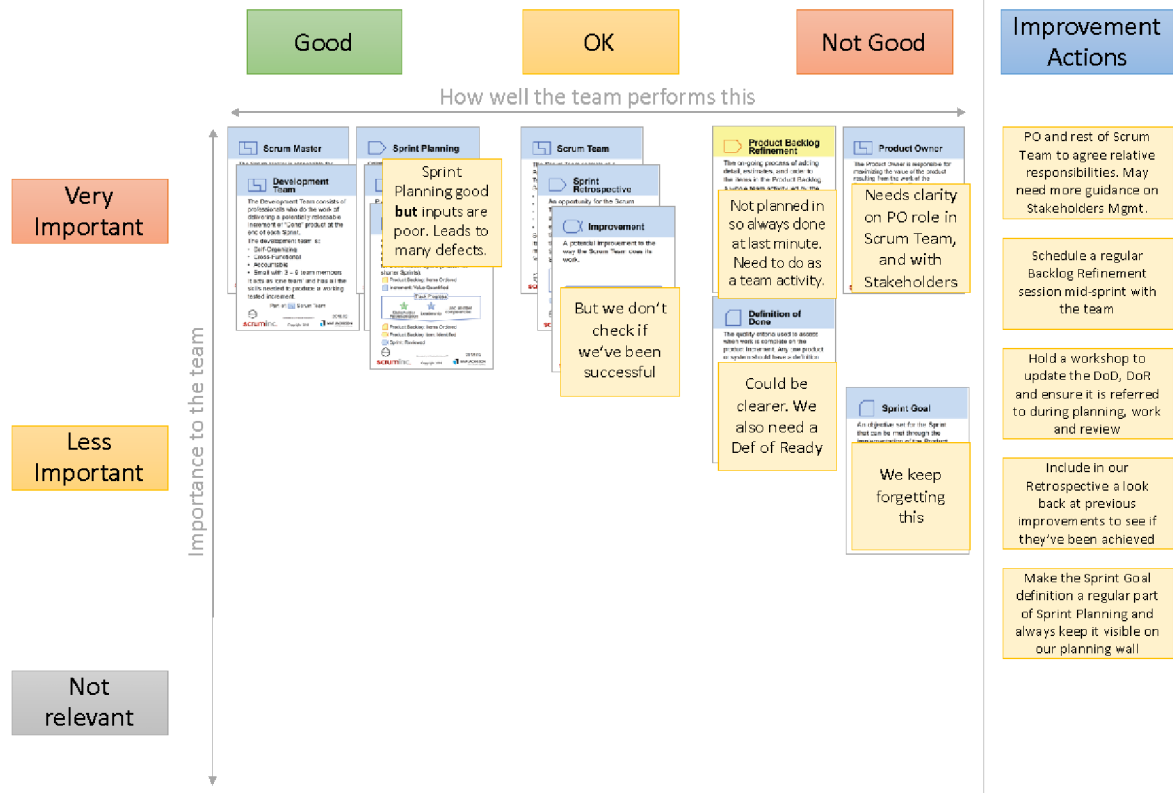


FIGURE 5 The Practice Patients game with problem comments and improvement actions

1. Create a board as suggested by Figure 4. The vertical axis is how important it is to the team. The horizontal axis is how well the team feel they perform that concept.
2. Lay out the cards from the practice on the board, discussing and moving the cards as a team
3. The cards at the top right are the ones for the team to focus on as they are considered important with the most room for improvement. If there are many cards in that area then the team can prioritize them to focus their efforts.
4. Finally the team discuss the nature of each problem and come up with some improvement actions which can be taken into the next Sprint Planning. See Figure 5. A column on the right-hand side is created to guide improvement actions rather than just talk about the problems faced.

This organised retrospective game can help teams who are fairly new to Scrum or for experienced teams who perhaps have not looked back at the essentials in a while. It also prompts teams to take action and plan in their improvements.

Although only a glimpse of better Scrum through Essence can be presented in this short communication, we invite you to learn more about the topic via blog posts [9, 10] and videotutorials [11, 12, 13] we have prepared on the topic to guide software engineers through freeing Scrum and other agile practices from method prisons.

4 | DISCUSSION

The value proposition of Essence for Scrum is broad and diverse as it provides many complementary advantages to a range of stakeholders. In summary, let us consider Essence from a number of these different perspectives.

Individuals gain the ability to see the big picture of software engineering that will always be useful to them throughout their career, even as specific practices change and evolve around them.

Teams are able to flexibly mix and match practices from a number of sources that suit their situation and needs. Crucially this guidance focuses the team on the essential items and provides actionable advice in the form of simple checklists to guide and measure progress. The card-based nature enables the whole team to be involved in bringing their practices to life and reasoning and improving their way of working through serious games.

Teams of Teams can improve their overall performance by having clear and shared practices for coordinating teams where it matters, while still allowing the flexibility for local practices within the teams. New practices that have been pioneered and proven in some of their teams can then be shared and used by the other teams. Having a common way of seeing and measuring progress gives a clear view of status across all the dimensions that are important for success.

Organisations have the opportunity for lean and agile governance, where the teams transparently apply the states and checklists to their work, avoiding much overhead and non-value-add bureaucracy. The common language and way of thinking about process makes it easier to form teams and have them coordinate with others. The organisation can still have common practices without being constrained within a one-size fits all process. This allows for industry standard process frameworks that can evolve on their own over time, practice by practice, as the organisation learns from within and from the outside world.

Coaches and Service Providers gain some common ways of expressing, teaching and putting their ideas into practice. Essence enables this knowledge to be plugged into an industry standard framework for context and for composing with other practices. For coaches, working with the practices and games helps them foster team understanding and application, giving structure to their services and ensuring the knowledge sticks long after they have gone.

Finally, **Industry as a whole** benefits from having for the first time a common ground of elements to use and a common language to express practices, all focussed on the essentials. This makes it easier to compare, mix and improve practices from various sources and make them available to others. This will accelerate the learning and sharing cycles across organisations and liberate the learning and knowledge from around the world. Academia gains the same structure with Essence to help with research and share findings, while students of the future will hit the ground running in their new jobs, holistically understanding what is important, and speaking and thinking with the same common language.

5 | CONCLUSION

In this paper, we have explored the synergy of Scrum and Essence, via their essential concepts and use cases, to re-introduce the clarity into planning, tracking and working on complex software projects. This short communication is further accompanied with a set of videotutorials and 21 Scrum Essential cards that turned out to be an extremely useful tool available for participants of Scrum workshops to represent their understanding of Scrum, the missing or suboptimal pieces in their implementation of Scrum, and their path towards a higher performing Scrum.

references

- [1] Jacobson, Ivar, Pan-Wei Ng, Paul E. McMahon, Ian Spence, and Svante Lidman. "The essence of software engineering: the SEMAT kernel." *Communications of the ACM* 55(12), 42-49. ACM, 2012.

- [2] Jacobson, Ivar, Pan-Wei Ng, Paul E. McMahon, and Michael Goedicke. *The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons!*. Morgan & Claypool, 2019.
- [3] Jacobson, Ivar, et al. "Essence in Practice" Available online at <https://essence.ivarjacobson.com/>
- [4] Henderson-Sellers, Brian, and Jolita Ralyté. "Situational method engineering: state-of-the-art review." *Journal of Universal Computer Science* (2010).
- [5] Jacobson, Ivar, and Roly Stimson. "Tear Down the Method Prisons! Set Free the Practices!." *Queue* 16, no. 5 (2018): 101-127.
- [6] Greer, Des, and Yann Hamon. "Agile software development." *Software: Practice and Experience* 41, no. 9 (2011): 943-944.
- [7] Mirachi, Samoel, Valdir da Costa Guerra, Adilson Marques da Cunha, Luiz Alberto Vieira Dias, and Emilia Villani. "Applying agile methods to aircraft embedded software: an experimental analysis." *Software: Practice and Experience* 47, no. 11 (2017): 1465-1484.
- [8] Scaled Agile. "Scaled Agile Framework (SAFe)" Available online: <https://www.scaledagileframework.com/>
- [9] Sutherland, Jeff. "Better Scrum Through Essential Cards". Available online at <https://www.scruminc.com/better-scrum-through-essence/#comment-31323>
- [10] Kerr, Brian, and Ian Spence. "Better Scrum Through Essence". Available online at <https://essence.ivarjacobson.com/publications/blog/better-scrum-through-essence>
- [11] Ivar Jacobson International. "Better Scrum with Essence: An Introduction to the Essence Cards". Videotutorial available online at <https://vimeo.com/476004142>
- [12] Ivar Jacobson International. "Essence in Practice – The Guided Tour". Videotutorial available online at <https://vimeo.com/447251988>
- [13] Sutherland, Jeff, and Ivar Jacobson. "A Better Scrum with Essence". Videotutorial available online at <https://essence.ivarjacobson.com/videos/better-scrum-essence-jeff-sutherland-and-ivar-jacobson>
- [14] Park, June Sung, Paul E. McMahon, and Barry Myburgh. "Scrum powered by Essence." *ACM SIGSOFT Software Engineering Notes* 41(1), 1-8. ACM, 2016.
- [15] Sutherland, Jeff, Ivar Jacobson, and Brian Kerr. "Scrum Essentials Cards: Experiences of Scrum Teams Improving with Essence". *Queue*, 18(3), 83-106. ACM, 2020.
- [16] Pieper, Jöran, Oliver Lueth, Michael Goedicke, and Peter Forbrig. "A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects." In *2017 IEEE Global Engineering Education Conference (EDUCON)*, 1689-1699. IEEE, 2017.
- [17] Zapata-Jaramillo, Carlos Mario, Grissa Vianney Maturana-Gonzalez, and Johnathan Mauricio Calle-Gallego. "A Board Game to Simulate the Software Development Process Based on the SEMAT Essence Standard." In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*, 1-4. IEEE, 2020.

Ivar Jacobson has made seminal contributions to software engineering spanning over 50 years starting from components and component architecture in 1968 to the essentials of modern software engineering in 2019. In between he created Use Cases and what became the Unified Process in 1986 and he was the co-creator of UML in 1997. Ivar has authored eleven books and written hundreds of papers on subjects related to software, system and business engineering. Ivar is the founder and CEO of Ivar Jacobson International.

Jeff Sutherland the inventor and co-creator of Scrum, has worked with thousands of companies deploying Scrum and recently launched two global trainer programs for Scrum Inc. Scrum Trainer and Certified Scrum@Scale Trainer, in addition to creating two independent companies, Scrum Inc Japan and Scrum@Scale LLC. He has been VP of engineering and CTO or CEO of 11 software companies. In the first four companies he prototyped Scrum and in the fifth company created Scrum as it is now used in 74 percent of agile software companies in more than 100 countries. In 2006 Sutherland established his own company, Scrum Inc., now recognized as the premiere source of Scrum training in the world.

Brian Kerr is a principal consultant and one of the founders of Ivar Jacobson International in the UK. He is an experienced agile coach, consultant, trainer, and change agent specializing in the adoption of key software-development practices in a pragmatic and sustainable way. He works at all levels with individuals, teams, teams of teams, programs, coaching senior management, and strategic support to organizational improvement programs. He has been involved in the development of the Essence standard since the beginning and in particular formed the early ideas of making the process tangible and representing it on cards to enable serious games. He is currently working on Essence tooling.

Barbora Buhnova is an Assoc. Prof. and vice-dean at Masaryk University, Faculty of Informatics (FI MU), Czech Republic. After her research stays in Germany and Australia, she now leads multiple research teams at FI MU (on software architecture), and Czech CyberCrime Centre of Excellence C4e (on critical infrastructures). She is the Steering Committee chair of the ICSA conference, and acts as a reviewer and editor in multiple scientific journals. Besides, she is a Co-Founding and Governing Board member of Czechitas, a non-profit organisation aiming at making IT skills more accessible to youth and women.