

Scaled Agile



→ **How to implement transformation**

Questions and answers,
expert opinions,
and real-life experiences

Preface



At Enalean, we understand agility and the actual human and managerial qualities involved, the qualities that help you move forward, that help you enjoy working and progressing alongside your colleagues. We understand what it means on a day-to-day basis. Anyone who's come to spend even a day at Enalean has experienced it, felt it. Clients, technology partners, students, new employees— they've understood the ways we are different, why we see our customers as individuals, and how we succeed in helping them.

So, when—being immersed in this agile culture and market for agile tools—I felt that scaled agile was the next big thing and was becoming a real lever for transforming organizations, I told myself, “Wow, it's a great idea; every team will benefit from the virtuous circle in which I get the chance to develop professionally.” Then, very quickly, I asked myself a whole bunch of questions, from the most simplistic to the most operational.

That's the reason why I'm now offering you this white paper, which considers a good many questions that I've dealt with myself. To answer those

questions, I asked four experts in the field to help me and share their experiences. I am sincerely grateful to them for their contributions.

Of course, the purpose of this white paper is to make us aware of what we at Enalean can offer, our expertise, our Tuleap product. I have to admit, though, that if it were also able to play even a little part in making the foundations and effectiveness of agile approaches the new norm in large organizations, my goal would also be achieved.

As you read the white paper, please therefore pick out what speaks to you and let me know what you thought. I'd be happy to hear your views.

Manon MIDY

Agile Evangelist, Tuleap Marketing Manager at Enalean



[Contact us](#)



[Follow Tuleap](#)



Introduction

We are now in the era of the scaled agile approach. Agility at scale is also one of the major challenges for 2021 and the years to come in large organizations. This is because enterprises that have seen the benefits of an agile approach within individual teams are eager to reap the benefits for the whole organization and its projects.

However, “becoming an agile enterprise” means, above all, rethinking the way the business works by reorienting it toward its customers, and changing the corporate culture to bring about greater collaboration between teams and between individuals.

This book is aimed at all of those working with the enterprise who are, or will be, involved in an agile innovation process. Whether you are the one initiating this new business approach or are working with an operational team, this e-book will help you better understand the challenges and benefits of agility at scale.

The aim of this book is to help you understand the potential choices, challenges, and consequences involved in implementing a scaled agile approach. Once these foundations have been shared and understood by all,

we can move on to the phase of applying the scaled agile methodology in practice. That aspect will be the subject of a second white paper to be published in the coming weeks.

For this e-book, four agile experts and coaches have contributed and answered by giving their answers to no fewer than 23 questions. In a “round table” format, each of the experts shares their opinion, knowledge, and experience. The e-book is a great resource for enriching your learning about the topic, for reading, and for coming back to.

Contributors

In alphabetical order



Laurent Charles

 Enalean - Tuleap

I'm the co-founder and CEO of Enalean. I spent the first 16 years of my career working for technology businesses, being particularly involved in the software developments and software revolution

experienced by those companies. Since 2005 I've taken a particular interest in how good software can be developed and in the practices and processes to be implemented—including agile approaches.

I've become profoundly convinced that software has become a strategic issue for businesses. This led to the creation of Tuleap and Enalean and prompted me to engage with organizations to help them turn their software applications into a competitive advantage.

I've also drawn inspiration from the practices followed by free and open-source communities, and I promote this culture of cooperation and “co-opetition” as a driver of innovation and efficiency.

By deploying Tuleap as a tool to support agile and DevOps processes, I get

the chance to work alongside people in a variety of roles and teams, whether focused on software or not, and in organizations of all sizes at different stages of their agile journey. I also have to deal with successes and failures, with people who are enthusiastic and those who are fearful. Sharing this experience is another of the reasons why I'm contributing to this white paper.

Alexandre Cuva

 SoCraAgile

As a recognized agile specialist with over 20 years' experience in international environments, I'm an Enterprise Coach, Agile Coach, and trainer at SoCraAgile Sàrl, a company based in Nyon, Switzerland.



I have extensive experience in IT as a developer, architect, product owner, Scrum Master, coach, CTO, and CEO in the areas of finance, government, luxury watches, and insurance. I've been a certified SAFe Program Consultant 5 (SPC5) since 2017.

As well as coaching, I give training and presentations on agility, Kanban, team dynamism, remote collaboration, happiness, agile leadership, Management 3.0, agile transition, agility at scale (SAFe and Scrum@Scale), DevOps, and software craftsmanship.

I'm passionate about technological change, agile development techniques in Java, C #, Python, Elixir, and JavaScript, and about OpenUP, XP, and

Scrum practices. I work as an Agile Organizational, Software Craftsmanship, and DevOps coach.

Having attended a number of events worldwide (such as XP Day, SoftShake, Agile Tour, and ScrumBeers), I'm a co-founder of the ScrumBeer in Geneva and Da Nang (VN) and a member of the Agile Vietnam committee.

Jean-Claude Delagrange

As a project manager from 2003 to 2010 and then an agile coach since 2011, I have over 13 years' experience in agile. Since 2007, I've implemented and managed agile projects in Scrum, Kanban, and other frameworks. I've trained and supported IS project teams (Scrum Masters, product owners, development teams) in their agile transformation; I've also worked with operational teams (real estate, budget, and HR) on both IS and non-IS projects. More specifically, I've been involved on the ground in projects facing major challenges, guiding teams toward approaches for resolving these (with some failures along the way, which were good learning experiences).



As a team coach, I've also worked with executive boards and strategic programs (involving more than 100 people) on transformation projects; I've been SAFe-certified since August 2020 after working in the field for nearly three years on the Banque des Territoires (French regional investment bank) and the Mon Compte Formation (personal training) platforms.

As part of my role as an agile and transformation support coach, I've also implemented product owner and Scrum Master training (designing and leading courses), design thinking (UX) projects, co- construction (project framing and launch) workshops, and individual* coaching activities for managers on their managerial role, all in an agile context.

* I completed the "Systemic Coaching" training course provided by MetaSysteme (Alain CARDON).



Laurence Hanot

 Zenika

Having accidentally stumbled on to the agile scene back in 2007 at Kelkoo, I've been involved in the agile community as an active member of Club Agile Rhône-Alpes and Agile Grenoble for more than 10 years. I've also trained and supported multiple teams and organizations, first as a Scrum Master and then as a coach, trainer, and facilitator.

To date, I've trained and supported more than 2,000 people, in very different contexts and sectors, from a simple Scrum team buzzed about developing a brand-new software product (the dream!) to the huge initiative I'm currently supporting. This involves taking a SAFe approach to software, embedded systems, and physical products, as well as the hardware and IT support team that has benefited in broad terms from a Kanban approach. I also have the opportunity to create, roll out, and deliver multiple agile training courses.

Holding SAFe 5.0 certification, I share my professional expertise and my “soft skills” at a number of agile events nationally. I support teams and organizations with the complexity of agile approaches both at scale but also within teams, in the different facets of their agile practices (product, flow, and interaction management).

Table of contents

| | |
|--|-----------|
| Understanding scaled agility | 15 |
| What does scaled agile mean? | 15 |
| What are the benefits of scaled agile?..... | 18 |
| What's the difference between a lean approach and a scaled agile approach?..... | 20 |
| What main challenges are involved in scaling up?..... | 21 |
| What are the key factors involved in making scaled agile practices work? | 22 |
| How should you go about selecting your scaled agile framework? | 30 |
| What is the minimum number of people or teams needed to consider moving to a scaled agile framework? | 35 |
| Can scaled agile practices be implemented if everyone is working on their individual project? | 36 |
| How can the effectiveness of scaled agile approaches be measured? | 40 |
| Is scaled agile a top-down or bottom-up approach, or both? | 43 |
| How does a scaled agile approach differ from a V-cycle with user feedback? | 46 |
| What is the connection between DevOps and scaled agile approaches? | 49 |
| How can scaled agile practices be implemented with distributed teams? | 52 |
| In concrete terms, where should we start if we want to initiate a scaled agile approach?..... | 57 |
| What role do customers play in a scaled agile approach? | 63 |
| What new roles are needed in order to implement scaled agile practices? | 66 |

| | |
|--|----|
| What additional skills are needed in order to implement scaled agile practices? .. | 68 |
| Is SAFe really an agile methodology?..... | 69 |
| Is SAFe truly safe and secure for my organization? | 72 |
| If we've never used Scrum techniques, can we still implement SAFe? | 73 |
| Is a software application needed in order to take a scaled agile approach? | 75 |
| What should we do if our teams are already using different tools? | 77 |
| What should we do if our teams have different workflows? | 79 |

The 10 points to remember... 82

To continue making progress 83

| | |
|-------------------------|----|
| Webinar..... | 83 |
| Client case study | 84 |

About... 85

| | |
|--------------------------|----|
| Tuleap and Enalean | 85 |
| SoCraAgile | 86 |
| Zenika | 87 |



Understanding scaled agility

What does scaled agile mean?



Laurent Charles

“Scaled agile,” as it’s termed, involves scaling up agile approaches within an enterprise.

We might say that this scaling process can involve three different approaches:

- Having an increasing number of agile teams in the organization. This involves implementing tried and tested practices, typically with some pilot teams leading the way, so that more teams—or even all of the teams—take an agile approach.
- Making projects more and more agile. This is a less well-recognized aspect of the scaling process as it’s probably less visible. Here, teams that are already agile increase the number of agile practices they deploy. This is a fairly natural development, since an agile approach

naturally encourages progress and the implementation of new practices. However, we're still talking about scaling up in this case, as extending the use of such agile practices involves more and more people, technologies, customers, and suppliers, thereby complicating the smooth operation of an ideal Scrum team of seven people.

- Implementing increasingly large-scale agile projects. Often the concept of scaled agile is used to refer only to this particular aspect, in terms of having 50, 100, 500, or 1000 people work in an agile way on the same project. And that is a real challenge. However, it's interesting to note that success on this point often depends—or possibly always depends—on the progress made in respect of each of the other approaches.

In fact, from my perspective, scaling up an agile approach actually tends to relate to iterations in any one of these three aspects, with the team then moving to another aspect, and then proceeding with iterations based on the context or objectives involved.



Alexandre Cuva

A scaled agile approach also enables different departments to become aligned, such as any departments that share a common goal (for example IT, sales, and support). Scaled agile doesn't necessarily involve the application of a specific framework, such as SAFe, Scrum@Scale, LeSS, or Nexus¹... You can scale up using basic practices like Scrum.

1. Scrum@Scale, LeSS (Large-Scale Scrum) and Nexus are scaled agile frameworks that build on the core principles of Scrum and are designed to bring multiple Scrum teams together.

“



Jean-Claude Delagrange

In addition, a scaled agile approach can be applied to an important project, an entire program, or even all or part of an enterprise.

“



Laurence Hanot

Like Laurent and Alexandre, I see scaled agile as a multidimensional approach.

“Horizontally,” it enables multiple business units, departments, and teams to collaborate on aspects ranging from gathering user requirements to delivering the systems or applications that meet those needs. Here, we can implement a number of practices, such as design thinking, lean startup, Kanban and/or Scrum, DevOps, and so on.

I see scaled agile as a multidimensional approach: horizontally, it enables multiple business units to collaborate and, vertically, it allows a large number of teams to work on one product.

Its “vertical” dimension comes in when we need to have a large number of teams collaborating so as to deliver a product. Here, scaled agile frameworks are a helpful way of embracing the complexity that Scrum alone cannot always address.

Finally, an enterprise-wide scaled agile approach will leverage these two

axes at the same time, expanding the approach by also adding in the rest of the ecosystem, including HR, finance, commercial, and so forth.

What are the benefits of scaled agile?



Laurent Charles

According to Gartner, in 2020 approximately two-thirds of IT directors and R&D managers in technology companies prioritized the rollout of scaled agile practices. Agile represents a major transformation of an enterprise's culture.

Scaled agile involves a profound transformation of the way that an enterprise is organized and operates. Organizations that take this step will be more flexible in their way of operating, more capable of absorbing the impacts of change, and more responsive in terms of bringing out products that meet customers' expectations.

6 ways in which scaled agile adds value

Originally, the goal of scaled agile was simply to put in place a framework that enables multiple agile teams to work cooperatively. A number of effectiveness-related benefits have resulted from this type of approach:

1. **Better responsiveness** with regard to strategic business issues
2. Establishment of an **innovation-oriented approach** across all business units
3. Greater **cross-team collaboration**
4. Adoption of a **"user"-centric approach** for a better customer experience
5. **Better management of teams** on a day-to-day basis
6. More effective **prioritization of tasks**

Agile practices give more space to teams and people, more breathing room and autonomy, and greater collaboration between the enterprise's departments.

“



Alexandre Cuva

Businesses today need to change if they are to remain competitive in an unpredictable and constantly changing market. Even if we talk about strategic issues like digitalization, having an organization with a complex structure doesn't allow us to be innovative. Innovation requires creativity, and a highly constrained environment doesn't allow for creativity. Similarly, an organization can hire the best talents under the sun yet still not achieve the magical results it's looking for. Nowadays there are some enterprises that are generators of talent in that they provide a fulfilling environment, one that allows creativity. These organizations are today's market leaders.

“



Laurence Hanot

An agile and—to an even greater degree—a scaled agile approach enables businesses to combine the organizational and managerial transformations required for the digital transformation of the world and therefore of enterprises and their markets. It gives (back) power, visibility, and ownership to the people actually “doing” the work—the same people who will then be able to innovate, organize, and adapt to this changing world.

What's the difference between a lean approach and a scaled agile approach?



Alexandre Cuva

A lean approach is a scientific approach that involves endeavoring to optimize an organization's workflows. This might be done by adding or removing processes. With a scaled agile approach, we're seeking to do the same thing while adding in the human factor. With regard to the latter, we set to work on the relationship between the different parts of the same system. These parts are the members of a team, who are also members of a department and an organization. A scaled approach will in any case analyze value streams to understand the relationships between systems.

A SAFe framework, for example, is a hybrid between a lean and an agile approach, in the sense that the organization is lean and the operation is agile. All other currently available frameworks are agile approaches.



Jean-Claude Delagrange

In addition, lean thinking has fed into the agile approach and the two systems' values are still very close to one another. The scaled agile methodology, however, has introduced rules and toolsets for aligning and synchronizing teams, as well as roles and responsibilities specific to it that are derived from Scrum in general and not defined in the lean framework.

“



Laurence Hanot

So, yes, while lean has fed into agile, both approaches incorporate a very strong human dimension. They are rooted in the same values, but their views of the enterprise and their principles and tools aren't the same. Unfortunately, the defenders of these two approaches sometimes clash, each arguing that one or other of the two systems is better. From my point of view, they can be complementary. This is what SAFe tries to highlight, albeit not always in a very clear way. SAFe sometimes gives the impression of being a ragbag of every lean or agile approach that's ever existed.

What main challenges are involved in scaling up?

63%

The corporate culture is in conflict with agile principles

34%

There is insufficient training

20%

Equipment, data, and monitoring are fragmented

45%

Lack of support at management level

Source: Harvard Business Review 2020

What are the key factors involved in making scaled agile practices work?

52%

In-house agile coaches

41%

Consistent practices and processes

48%

Support from the corporate hierarchy

36%

Implementation of a tool shared by all teams

Source: Harvard Business Review 2020



Jean-Claude Delagrance

First and foremost, a product strategy is needed, driven by a trusted management team. Based on my own experience, this involves:

- The appointment of a pioneer (and future director) who is able to play the agile game often in the face of backward-looking values and reflexes; however, don't forget to train him or her up properly and, above all, ensure that your agile coach focuses initially on this person, who'll play the role of Business Owner¹

1. Business Owner (BO): in SAFe terminology, Business Owners have the primary business and technical responsibility for governance, compliance, and return on investment (ROI) for a solution developed.

- The designation by the leader of a trusted person, trained in agile techniques, to act as Product Manager¹, working with the Business Owner as efficiently as possible.

Secondly, the assignment of properly designed roles, with clear boundaries, is vital; most importantly, you need a Product Manager who is capable of guiding the product development strategy, who can act as a single, available, and legitimate point of contact (for both teams and his or her managers), who is brave enough to innovate, and who can understand what's involved in initial prioritization, which may evolve over time and with product deliveries.

Another important responsibility involves dealing with issues that cut across business units. Working hand in hand with the Product Manager, a person designated the “super Scrum Master” or other title will be responsible for managing cross-functional accountabilities (which should be developed gradually) and synchronizing teams.

Thirdly, clearly identified teams should be created, focused on the target product, with an agile culture already well established. If some teams aren't yet acculturated in the approach, it will be up to the agile coach to ensure this happens, and they can be trained by the most experienced teams. The road to an agile culture can be a long one, as people cling on to old ways of working (mini V-cycles, the project owner-project manager divide, etc.), and it takes a lot of pragmatism for the agile coach to move forward.

1. Product Manager (PM): Is responsible for identifying customer needs, prioritizing product functionalities, managing work, and developing the vision and roadmap.

This is how a current and future organization might look that's capable of aligning with this target (the achievement of a feature team-based structure being a distant objective):

- The maintenance teams for existing applications (e.g., back-office) who have to organize themselves so as to provide services and data...
- ... to teams responsible for new applications (e.g., smartphone apps)
- The teams that need to transform the applications they've already installed, in order to integrate them seamlessly into the new platform.

SAFe offers a very comprehensive framework, its main contribution being a way of naming, formalizing, schematizing, and integrating ideas from various sources (such as lean, Agile Manifesto, Scrum, and Kanban).

SAFe schematizes and integrates ideas from a variety of sources: lean, Scrum, Kanban, Agile Manifesto.

“



Laurence Hanot

As coaches, we unfortunately don't always arrive at the right time in this type of transformation, and often have to act as “firefighters.” In other words, “We've tried things, we've embarked on an agile transformation on our own (at scale or otherwise) and we're experiencing difficulties—please help us.” Depending on the state of play (identifying this is the crucial first phase in a coaching project), we then propose a variety of options that may, however, take a long time to

implement depending on the extent of the damage or shortcomings observed. I'd therefore say that obtaining training and support from the outset is essential.

Defining and explaining the reasons for such a transformation is the second key step, to be done with leaders and sponsors. Why change things? For what purpose? Agile isn't an end in itself. Rather, it helps in the achievement of goals that might be very distinct in different enterprises, departments, and teams.

Examples of transformation objectives mentioned in some of my projects include improving customer satisfaction, increasing the level of employee autonomy and engagement, developing employees' cross-functional skills, improving the quality of our products, reducing time to market, and re-engaging and re-enthusing teams.

It's clear that, depending on the objectives to be achieved (and measured throughout the transformation), we won't rely on the same levers and the same practices. Once the objectives have been established and shared, we are then able to work with the teams on identifying the way ahead— in other words, the focus areas to work



on—to achieve them. We then adopt an agile approach to our agile transformation, involving small steps and regular feedback loops to adapt our plan or continue on our way.

In order to involve as many people as possible so that they become active participants themselves, it's therefore necessary to make the transformation visible, to communicate widely and, often, to obtain the necessary support and sponsorship from the management team. In situations at scale, such as the SAFe context in which I'm currently mainly involved, this requires the creation and mobilization of a transformation team composed of different participants, such as coaches, managers, operational staff, and sponsors. The diversity of the people in the team is an asset and allows support and transformation to take place “from within,” involving for example teams, senior managers, the portfolio, solutions, Agile Release Trains¹, cross-functional roles, and so on in a coordinated and coherent manner. The goal is, little by little, to establish transformation representatives who work to change things in their role on a day-to-day basis. This will anchor agile practices, principles, and mindset to an ever-greater degree and therefore also enable the corporate culture to evolve.

Once the coaches and trainers have been appointed, the objectives and potential ways of achieving them established, the transformation team created and supported at the highest level, off you go!

1. Agile Release Trains (ARTs): in SAFe terminology, the Agile Release Train (ART) is a long-lived team of Agile teams, which, along with other stakeholders, incrementally develops and delivers solutions using a series of fixed-length iterations within a timebox of one Program Increment (PI). The ART aligns the teams on the basis of a shared business and technological mission.

Obviously, in practice things aren't so sequential or ideal and, in a scaled agile context, this necessarily involves a large number of people. There's a constant need to juggle between the different levels of maturity of the teams and people concerned, alleviate their fears and doubts, move forward with the "early adopters," communicate a lot, lead the transformation team itself, and so on.

Taking charge of the roles of Scrum Masters, Release Train Engineers¹, and Product Owners is the focal point of agile transformation.

To finish, I'd say that both in Jean-Claude's very good report and in my own experience, I remember one key point: the need to take charge of the roles involved, especially those of the Scrum Masters and the Release and Solution Train Engineers within SAFe or other equivalent scaled approaches. These people are the drivers of transformation, without which transformation can neither evolve, nor be viable in the medium to long term, nor become anchored in the organization.

1. Release Train Engineer (RTE): the Release Train Engineer is a servant leader and coach for the Agile Release Train (ART). The RTE's major responsibilities are to facilitate the ART events and processes and assist the teams in delivering value. RTEs communicate with stakeholders, escalate impediments, help manage risk, and drive relentless improvement.



Laurent Charles

A scaled agile approach can lead to a profound transformation of an enterprise; indeed, it must result in such a change. Managing change is important. There is a risk of failure.

How do you feel about these conversations?

- We're in the midst of an agile reorganization.
- Great, how's that going?
- I don't know yet; we haven't yet been notified of the new structure.

Or even:

- We've completed our agile transformation.
- What results have you achieved?
- HU... well, now we're "agile," but we've lost lots of our agility.

And also:

- We've decided to adopt agile practices.
- Good news; how are you going about it?
- We're putting in place tool X.

Without going into detail, here are some points we thought were key to the success of rolling out a scaled agile approach:

- Seeing agile first and foremost as a culture and only then as a process or practice.

- To achieve big things, learn to do the little things right. Large organizations need to relearn how not to go back into their silos, how to work incrementally, and how to identify the little important things to be done first, for example.
- Being supported by one or more coaches is vital. Ideally, these coaches will be external to your organization and will have the necessary perspective.

How should you go about selecting your scaled agile framework?

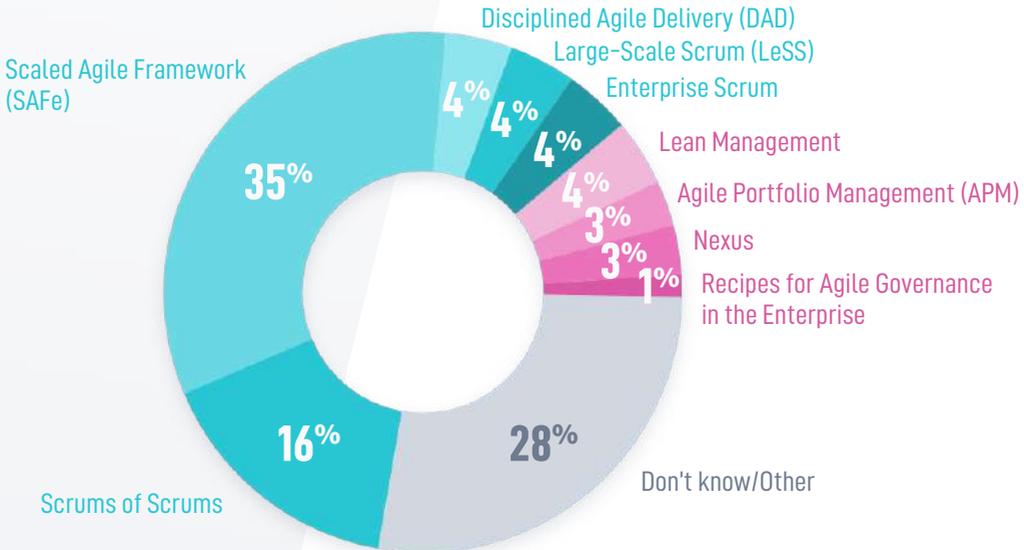


Laurent Charles

A number of frameworks are available to give you a structure for your work. Examples include SAFe (Scaled Agile Framework), LeSS, Scrum@Scale, and others besides—you have a choice.

These frameworks map out practices drawn from the experience of what the perfect agile organization might look like. They are full of examples of good practice and can guide you toward the method you might adopt or that might inspire you.

For your information, here are the most often-used frameworks in 2020:



Source: Annual State of Agile Report 2020

“



Jean-Claude Delagrance

The choice depends on the teams' and managers' levels of affinity with agile values and principles, as well as the degree to which decision-makers are committed to the transformation:

- SAFe retains a strong managerial and decision-making structure while firmly circumscribing the agile structures to be used (roles, milestones, organization of teams, decision-making methods, etc.). SAFe is therefore reassuring to organizations that have been slow to start their transformation process.
- LeSS, or other scaled organizational structures designed on the basis of context-appropriate practices (such as Sociocracy and Nexus), are indicated when the transformation takes place during the project, when the decision is already highly decentralized, and when the agile teams are well established. The generally autonomously organized transformation process will require less involvement from managers but will have to prove its worth, with particular attention given to communicating about the transformation actually carried out.

“



Alexandre Cuva

We have extensive experience of scaling up at an organizational level. Whether SAFe, Scrum@Scale, LeSS, or other frameworks are involved, each has a common denominator: stable Scrum teams focused on a single product. Scaling can only be achieved if this simplest element is handled in a sufficiently disciplined way. Only then can the organization opt for one or other of these frameworks, or a hybrid version.

A pragmatic coach will help you choose a solution tailored to your organizational needs. In one of our recent major transformation projects in the watchmaking sector, our customer contacted us regarding a SAFe transformation. After the client discussed the situation with one of our experts, we understood that their needs involved “doing as others do” and “wanting PI Planning¹” rather than really using SAFe. We’re now supporting the client with their implementation of Scrum@Scale, with planning releases in the format of PI Planning. There’s nothing to stop the client from moving to SAFe if that’s what they really want.

It’s therefore really important that you make your choice with the aid of a coach, who will help you choose the right framework for your environment and not simply follow fashion.

“



Laurence Hanot

Why choose? From my point of view, it’s important to look for practices and roles that will help in the achievement of transformation goals and, if they don’t exist, to invent them. It may sound a little provocative, but I actually think that announcing we’re going to deploy a particular method is a mistake that frequently leads to a “by the book” application, whereas in fact the very principle of agility involves adapting to a particular context. As I often say, I’ve never seen the

1. PI Planning: in SAFe terminology, a Program Increment (PI) is a timebox during which an Agile Release Train (ART) delivers incremental value in the form of working, tested software and systems. PIs are typically 8–12 weeks long. The most common pattern for a PI is four development Iterations, followed by one Innovation and Planning (IP) Iteration.

same agile approach adopted in two different teams, departments, or organizations—and that’s a good thing!

Having said that, it can be helpful, at least at the start, to choose a direction—the “main” framework to use as the basis. However, pay attention to the discussions concerning that choice. Even if, for example, you’re looking at a particular toolbox or a particular set of principles in your search, it doesn’t mean you have to accept everything or that you shouldn’t look elsewhere.

I’d say that two of my biggest projects over the last three years have been based on or inspired by SAFe but with two different approaches:

- A project conducted in 2017–2018 (a department of 50 people), in which Scrum was effectively used to bring agility to five teams in the same department on a gradual basis and then, when the issue of scaling became crucial for coordinating those five teams, to the rest of the department (managers and cross-functional roles). We then relied on SAFe to bring about this synchronization and scheduling: by syncing only the Agile Release Train and PI Planning before working on “in-house” roles to involve the whole department;
- My current assignment, for the past year (a program involving about 250 people), in which an attempt to deploy SAFe at the solution level was made without support and initially on a “top-down” basis; in this case, we needed to dismantle bad practices and then rebuild “from the bottom up” (SAFe Essential with adaptations, of course) in order to solidify the foundations and really involve the teams and not just their representatives. Currently, we’re leading this project alongside others

on the higher Solution and Portfolio levels; these are also required in order to structure and manage the complexity of this program. These levels also enable issues at the senior management level to be addressed and incorporated in this agile approach (portfolio prioritization, adjustment of roadmaps for going to market, incremental deliveries, involvement of teams in choices and decisions, etc.).

In the French agile community, we frequently see passionate debates between defenders of SAFe and LeSS. Yes, these approaches are different: SAFe is very descriptive and prescriptive, whereas LeSS is simple and gives teams a lot of autonomy in terms of organizing themselves. However, they're not in opposition provided that you don't apply things to the letter, and they can even complement one another: in fact, we now include practices from LeSS in our SAFe Agile Release Trains!



The constraints of any method come only from the way they are applied; nobody is forcing you to implement everything described in a particular method. On the other hand, it's vital that you explain the reason why a particular practice is being used and give coherence and consistency to a scaled agile approach.

I've never seen the same agile approach adopted in two different teams, departments, or organizations—and that's a good thing!

What is the minimum number of people or teams needed to consider moving to a scaled agile framework?

From two teams upwards, you're at scale.



Alexandre Cuva

We talk about a product being at scale as soon as more than one team is working on the same product. But that doesn't mean we have to immediately embark on a framework like SAFe that's recommended for 50 or more people. Two teams are two entities that coordinate their work with the help of a Scrum of Scrums, for instance.



Jean-Claude Delagrange

It's important to monitor the headcount within agile teams, since as soon as a team has close to ten people, it's easier to have two small teams working than one large one; even then this can be described as a scaled agile approach. However, as soon as a product in development is integrated into a wider environment, it's important to identify all the stakeholders and all the IT systems involved, so as to set up, as quickly as possible, the structure that will enable work to be synchronized most effectively.

“



Laurence Hanot

As soon as I join my co-authors, with two or more teams, we're at scale. Even a single Scrum team developing an application and working with other teams such as hardware, marketing, mechanics, physical products—for example, a mobile app for remotely controlling the radiators in your home—will, in my view, already be taking a scaled agile approach. In this example, I'm talking about scale in a “horizontal” sense, as I described in response to the first question.

You could say that you're working at scale from the moment when a team needs to synchronize and schedule their activities with those of other teams or functions.

Can scaled agile practices be implemented if everyone is working on their individual project?

“



Laurent Charles

Yes and no. If the teams are truly independent in that they're working on different products, the development and delivery cycles can carry on separately from one another. It will therefore not be necessary to set up synchronization meetings.

Nevertheless, we shouldn't forget that an agile approach also involves integration between people. We can therefore encourage exchanges

between project teams and set up communities of practice in which different developers and product owners can share ideas, as can be done in the Spotify culture, with its “chapters” or communities focused on specific topics with “guilds.” This sharing of ideas is essential as it creates a dynamic within the enterprise, helping all of the teams to progress.



With some of our clients, we also see that team members are encouraged to move from one project to another. This ensures that individuals have a richer and more diversified experience, as they collaborate with more people.



Alexandre Cuva

Among other things, in the agile world, we consider a project to be the start or an increment of a product. Even before embarking on a scaled agile approach, we recommend that our clients move on to having stable teams on a product or solution, in which those teams are initially given Epics (the business need to be delivered in the project) that will be broken down into smaller needs.

“



Jean-Claude Delagrange

For me, no—scaling up implies that all teams feel committed to the same strategic vision. And the concept of the project takes a back seat to that of the product, or even customers.

The contribution of each will form part of the features, the customer journeys (depending on the framework adopted), either within specialist teams for particular components (such as back-office, data, and customer interface) or, for more well-established setups, cross-functional teams organized around the customer journeys involved (feature teams).

As part of a development program, we can still speak of a project, in financial and management terms, for integration into the enterprise's overall vision, especially if it's not yet ready for an agile organization at the overall corporate level.

The concept of the project takes a back seat to that of the product, or even customers.

“



Laurence Hanot

Yes, scaled agile in a “horizontal” sense if we assume that a project equals a product.

Some rituals don't make sense if each team is working on a different product

That said, I also agree with Enalean’s point of view: some rituals don’t make sense if each team is working on a different product that has no interaction with the one next door. Nevertheless, it can be interesting to consider a scaled agile approach in these cases at the department or enterprise level, bringing it to all business areas (agile for HR, sales, marketing, etc.), but also cross-functionally to

teams so as to promote continuous improvement on a collective basis, constructive exchanges, learning through communities of practice, internal events, hackathons, “live my life”-type internships (working for a few days in another team,



another business area), and so on. A long time ago, an organization I was working for succeeded in accepting the following challenge: to move all its servers in just one month from two data centers that had just been sold and were no longer available. Working in multiple separate teams, the 70 people in this enterprise built a backlog, held a stand-up involving 70 people every morning for one month and everyone was able to contribute to achieving the goal. Although it had looked like an impossible task to achieve in such a short time, they succeeded. That’s the most inspiring story of a scaled agile approach that I know, and back then the term didn’t even exist.

How can the effectiveness of scaled agile approaches be measured?



Laurent Charles

To get straight to the point, identifying the much-discussed ROI (return on investment) that every leader hopes for after an agile transformation can be a tricky task. We might even say that, if agile transformation is guided only by a quest for this ROI, it often fails, as the focus is no longer on the development of the corporate culture required. To measure the effects of a move toward scaled agile practices, the usual KPIs are inadequate. We can start by looking at the impact on people, on employees' commitment, and motivation levels. Regular interaction with teams allows us to “measure” or at least “sense” their state of mind. We can look at their degree of involvement, of sensitivity to the approach in progress, their acceptance of the right to make mistakes, and discuss these.

It's therefore possible to measure process improvement, but drawing conclusions in monetary terms on the savings made would be a matter of guesswork. On the other hand, more efficiently managed projects, earlier deliveries, and incidents identified prior to production are more factual elements that bring real savings to the organization and help to give it a high-quality image.

If we really want to analyze this, we can assess the number of people collaborating in an agile way, the level of maturity of the teams and the prospects for growth. Remember, however, that the customer is the focus of

every aspect of the agile approach, and so measuring customer satisfaction seems the best measure.



Alexandre Cuva

The first thing to do is to go back to basics as regards the KPIs, which are there for us to learn from, as we only measure what we want to improve. Once this improvement has been achieved, you change the metric. It's also very important that these steps are taken by those who wish to learn more. A systemic view of these measures is required. A Scrum team measures its system, and an executive measures the organization and the relationships between the parts of his or her system.

Nowadays, there are all sorts of tools that allow an organization to better understand its adoption of agile, how a product is progressing, the returns from end customers, and so on. Each metric provides a benefit for a given system. As Enalean points out, measuring revenue involves hypothesizing about a choice that will have to be validated after a production increment has been implemented.

More and more organizations are adopting OKRs—Object Key Results—as a means of measurement. In this case, the organization sets a strategic goal. We then measure the Key Results that support it. Everyone in the enterprise pursues these goals in their own way.

Leaders often place too much emphasis on the results achieved and not enough on the potential for learning.

Francesca Gino, Bradley Staats

“



Jean-Claude Delagrange

I fully agree; I'd also add one point from my own experience: the need to consider what value a scaled organization can deliver to its customers, and how quickly.

This value will be measured directly by the use of the apps delivered, and indirectly by the effect on the revenues derived from the activity supported.

For example, let's say my team had been commissioned to implement an application for managing contracts awarded by the government. Thanks to the simplicity of the tool, its flexibility, its functional coverage, the speed of our response (rarely more than two months for the first delivery), our involvement as customers of the public sector procurement organizations, and the quality of our solutions, in the space of three years, the government entrusted us with all of its new contracts. And the business more than doubled its sales within five years.

“



Laurence Hanot

Once again, starting from the goals that have been set: why do we want to take a scaled agile approach? What problems are we aiming to solve? What do we want to improve?

From there, we can define relevant indicators for each of the objectives and measure and adjust them regularly if the objectives change, if we think we've achieved them, or if new ones appear.

As far as the “how” is concerned, we should base the approach on

appropriate methods and tools (OKRs, measuring the quality of the code, tools that support agile practices, user surveys, mood of the teams, and so on).

Is scaled agile a top-down or bottom-up approach, or both?



Laurent Charles

This is a question at the heart of how we “implement” a scaled agile approach. Becoming agile can be a strategic corporate ambition, and it’s actually much more effective if it is.

But we see everywhere that agile can’t be decreed; it can’t be forced. It’s based on motivation and it spreads throughout an organization on the basis of conviction. It can only be done with the teams—and only deployed by the teams themselves. Why? Because it’s the very autonomy, transparency, and accountability of the teams involved that make this approach unique.

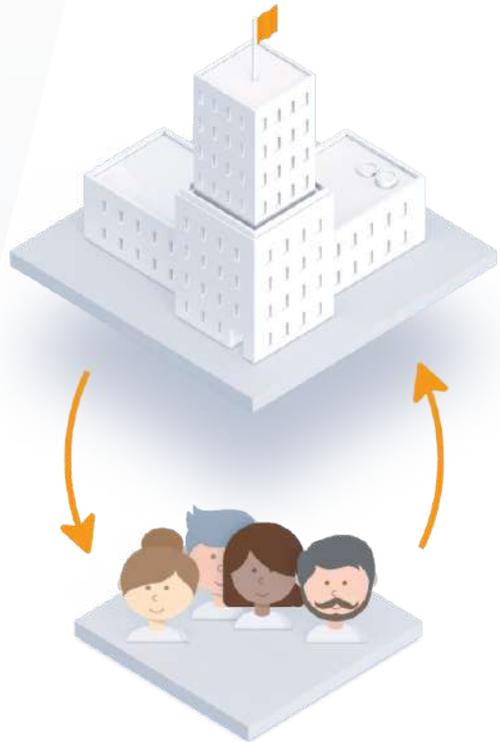
Within your organization, if you already have teams that are convinced of the merits of agile, it will be that much more effective. Build on these early adopters and motivate them to play a role in the bottom-up dynamic. As a showcase for the success of their project and for their well-being as a team, they are the ones that can really drive deployment. Word of mouth can act as a tidal wave in promoting the effectiveness of agile.

It won’t have a magic wand effect, and will necessarily take some time, but the change of mindset will be profound, and good practices firmly

established. Uptake will follow on from the successful example set by some teams and the new experiments tried by others. We've noticed that the clients that succeed are those that prioritize the facilitation of test projects, pilots with motivated teams.

A bottom-up approach on its own soon reaches its limits without the support of management. The reflexes of the enterprise and systemic resistance cannot really be overcome without the support of the management teams. As studies show, successful transformations are all based on strong sponsorship from management.

It's therefore a matter of striking the right balance. Because scaling up agile is a bit like our brain: it works much better if we use both the right and left hemispheres. The SAFe approach, for example, includes both top-down management (Strategic Agility) and confident bottom-up operational teams (Scale Up).



“



Alexandre Cuva

In addition to the comment from Enalean, I'd like to add that it's very important that such an adoption is understood by everyone, from sponsors at the executive level to operational teams, along with the various departments of the organization. Some organizations have created new units that respond to an internal need, such as internal marketing or an on-demand coaching service. Whatever they choose, a change cell has to be created to monitor the adoption process and restrict certain appropriations that could create more complexity for end users.

“



Jean-Claude Delagrangre

I fully share these points of view. I'd add another point that I feel is important (proposed by SAFe) about the need for managers to always ask themselves the question, including during transformation, as to what things can be decided as close as possible to ground level (rapid, local decisions, etc.) and what things need to be kept at the highest level of decision-making (strategy, financial issues, choice of transformation, etc.).

“



Laurence Hanot

Both are needed, of course! Just like my co-authors, I think one can't work without the other. I've experienced each of these situations: bottom-up scaled agile that hits the glass ceiling (middle management, other departments, other roles...) and runs out of steam; and top-down scaled agile that doesn't really take hold because of

the lack of clear intentions and resources provided (training, coaching, time). The syndrome of “yet another new method coming from above,” without people understanding the how, what, or why, and the lack of ownership or involvement for people on the ground. (If you don’t know it, I strongly recommend this video—“Start With Why?”—by Simon Sinek: [long version](#).)

Yes, successful and long-term agility at scale requires both approaches.

How does a scaled agile approach differ from a V-cycle with user feedback?



Alexandre Cuva

It’s difficult to compare a V-cycle that’s “project-oriented” with a scaled agile approach that’s “organizational.” Let me compare the V-cycle with an agile practice such as Scrum, for example. The V-cycle is a series of project stages, each of which is validated in FILO (First In Last Out) mode, so ultimately, it’s a sequence of stages.



Jean-Claude Delagrange

A V-cycle is still a structure based on the predictability of solutions, in which user feedback will have difficulty being heard as people won’t take the risk of questioning forecasts, or will do so only to a marginal degree. A scaled agile approach integrates the

customer so as to guide functional objectives and take account of the other important factor: allowing the product to be adapted as it's developed.



It is possible to envisage organizations at scale with a few teams organized in V-cycles on the margins, but integrating these into the synchronization milestones is always tricky, even if they are based on mini V-

cycles.

Organizing sprints into mini V-cycles in this way is a perversion of agile, especially when it predominates, in that it organizes developments in sequences (development specifications—testing— production) without the teams actually being multifunctional. This way of organizing things is fairly quickly exposed by visual management practices (Kanban), in which there's a requirement to track the stages.



Laurence Hanot

Ha-ha, good question—sounds like it's based on real experience! Obviously, these ought to be quite different, as my co-authors have explained.

In reality, and especially at large scale and with top-down approaches, the two techniques aren't that easy to distinguish.

For example, you might find yourself doing SAFe with a specification PI (Program Increment), a development PI, and a validation PI. And given the magnitude of the task, and at risk of being booed by the whole agile community, that can be a way of starting. Prioritizing collaboration between teams above frequent and incremental delivery is one possible approach. The next step might involve working on a prioritized portfolio, linked to offer and solution strategies, while integrating customer feedback. Another might consist of investing in integration flows, verification, validation, and continuous rollout in order to frequently deliver increments.

With a bottom-up approach, we should see a real difference and remain truly agile if we've scaled up intelligently.



Laurent Charles

Let's keep in mind that the V-cycle or waterfall (cascade) cycle is an approach that seeks to systematize project

implementation in view of the methodological experience acquired over time, and in particular in line with two key assumptions: that we know what we want to do, and that we know how we'll do it.

However, in our digital world, where technologies evolve very rapidly, and where needs are not defined or only poorly defined and change even faster, applying approaches based on the above assumptions is often akin to building a house of cards. You just can't define everything in anticipation of something you don't know. Or rather, in order to be able to define things

beforehand, you have to choose to work with obsolete technologies and not innovate too much. I'm not sure it's a recipe for success.

What is the connection between DevOps and scaled agile approaches?



Laurent Charles

What exactly is DevOps? Gartner, for example, proposes this definition: “the ability to deliver customer value faster.”

Strictly speaking, an organization can have an agile approach with the goal of delivering “more precisely” without yet having a goal of delivering “faster.”

This doesn't mean abandoning the implementation of those things that will allow value to be delivered more quickly. DevOps and process automation are elements of agile, and must be one of the goals, especially at scale.

Indeed, the more people that are involved, the more complex the task and the longer it will take to build, integrate, deliver, and so on. Without taking into account the vital need to optimize such a situation, a large organization might find itself facing the paradox of developing better, but less quickly—and that's a shame.

We have found that an organization in which everyone keeps DevOps at the

team level is less effective than an organization that handles DevOps at a scaled level.

We need to watch out though, for possible mistakes in the order of the phases. If you're aiming to deliver faster before delivering better, you just risk delivering faster a... well... a load of mess...

“



Alexandre Cuva

Strictly speaking, there's no link between DevOps and scaled agile. You can do DevOps without scaled agile and you can do scaled agile without DevOps.

Nowadays, we recommend that our customers also adopt DevOps and software craftsmanship, which are complementary to one another. Often, an organization's aim in adopting a scaled agile approach is to speed things up; for example, Sandro Mancuso, founder of the London Software Craftsmanship Community, said in his presentation “Software Modernisation—2020”: “An organization cannot accelerate if it continues to follow IT practices from another time.” The practices of DevOps and software craftsmanship promote those approaches that will allow such an acceleration.

“



Jean-Claude Delagrance

I think that scaled agile gets very difficult without DevOps, and all the more so without automated tests, as the complexity of the overlaps between software blocks makes production difficult (non- regression tests in particular), unless deliveries

are only made in an intermediate environment and integration testing is performed at the end of the cycle. (We then lose one of the major benefits of agile: time to market.)

For me, teams need to welcome an “Ops” specialist from the outset in order to integrate the release- related tasks into the backlog and encourage development teams to think about the whole cycle, incorporate it in their “definition of done,” and ultimately show it in their velocity measurement.



Laurence Hanot

Here, too, I agree with the views of my colleagues and with those I outlined in the preceding paragraph. A DevOps

approach to agile and scaled agile is essential even if all too often it’s forgotten or set aside. This only pushes the problem downstream and causes disruption to the people involved in the deliveries to the users (Ops, country rollout, sales reps, user trainers, operational marketing, sales channels, and so on).

And here too, there’s one essential question and focus of work within an agile transformation: what do our users need in terms of delivery? What are our current practices and frequencies? What do we need in terms of approach and toolsets? What investments are needed in this area and when?

In any case, let’s not forget that DevOps is first and foremost a collaborative approach, which is a fundamental aspect of scaled agile. My first DevOps experience before the word existed was back in 2007 when we included an Ops person in our Scrum team!

How can scaled agile practices be implemented with distributed teams?

“



Jean-Claude Delagrange

In decentralized structures on multiple sites, even if they are a few metro/tram/bus stops apart, the scaled agile structure has to cope with an additional distance-related constraint. This impacts one of the main qualities of agile: “working together.”

The distance might be between members of the team (e.g., developers and project management, teams and users), or between teams.

If we add in a subcontracting relationship, often involving a development team, we are adding a contractual constraint, which—if not provided for at the time the contract is signed—will have to be treated as an additional risk.

It’s essential to set up events as soon as possible that bring everyone together once every one or two sprints (typically review and retrospective events), to test the organization of roles, and to encourage feedback and cross-functional learning. (SAFe recommends PI Planning only every 3 to 5 sprints.) The first few of these will be a total mess, but perseverance is vital.

Here, dedicated logistics and tools are important, under a person with identified responsibility, including:

- Communication tools for video conferencing, information-sharing tools (wikis, project management tools), virtual moderation (using tools such as Beekast);
- Dedicated meeting rooms equipped with video conferencing facilities/large screens, supplemented by individual equipment;
- The availability of a large space for reviews;
- A tool for sharing Tuleap-type or other backlogs.



Particular attention will need to be paid to the quality of the links between teams and to supporting people performing a cross-functional role. This is why the alternation between virtual and real-life meetings has to be carefully organized.



Alexandre Cuva

And yes, of course, now more than ever with lockdown and home working, it's been shown that distributed ways of working are possible. But this requires organizations to be equipped with tools for remote collaboration. A product like Tuleap is

already a solid foundation, one that can be complemented by products for desktop virtualization, whiteboarding, and other things.

“



Laurence Hanot

A scaled agile approach with distributed teams will always be less easy to implement than with teams in the same location.

The human factor is essential in agile: we've all had some experience of working with someone remotely—including in the corridor or the department next door—and finding it complicated.

“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”

Agile Manifesto, 2001

This principle can be proven in all contexts, in all organizations.

However, when faced with an international scenario or the simple fact of teams distributed across multiple sites, the issue remains unresolved.

The use of tools then becomes indispensable. We should distinguish between two types of tools:

- Project management tools are the ones that are most often indispensable, especially in a scaled agile context, where visual management is no longer sufficient. In addition, consistency and continuity in the monitoring of activities, from management of the backlog to the deployment of stories— including all good practices for

development, testing, and integration—are a real help for agile teams. A tool like Tuleap can meet these needs very well;

- Collaborative tools for developing, building, and sharing ideas (Google Suite, Mural, Klaxoon, Miro, etc.); although these will never replace the effectiveness of a team working together in a room with a whiteboard and Post-it notes, they are necessary nowadays.

In addition to the tools, here are some further tips:

- Hold regular physical meetings, even if it's expensive (as we don't measure the cost of not doing this, and I'm convinced of the ROI!);
- Turn on your cameras when video conferencing;
- Use communication tools that allow small groups and large groups to interact (Zoom breakout rooms, Microsoft Teams channels);
- Introduce virtual cafés (informal gatherings where you talk about something other than work);
- Set up random virtual meetings between all members of the organization (speakeasy);



- For kick-off events, workshops, PI Planning sessions and agile days, there are some great tools such as Remo (used for the Agile Online Summit in April 2020) or Sococo, which allow you to (almost) believe you're in the same place;
- Encourage cross-functional initiatives such as hackathons, communities of practice, sharing experiences between teams, agile days, and so on.

Note also that, in SAgile, the geographical factor should be taken into account when defining Agile Release Trains. Finally, above and beyond the geographical distribution, the cultural factor is also important in an agile transformation: European, North American, and Asian cultures aren't the same and this can create obstacles, misunderstandings, and tensions if it's not taken into account.



Laurent Charles

We all generally agree that it's preferable to have people physically present. But in reality, it's extremely rare that all of those involved (sometimes hundreds of people) can be brought together in one place. Appropriate tools are, in fact, indispensable at scale. So, allow me to share some tips for success with distributed teams:

- Rule #1: see yourselves, show yourselves, and show others what you see using video conferencing, dashboards, and digital backlogs;
- Rule #2: have confidence and inspire confidence;
- Rule #3: help one another.

But these points are obvious, aren't they, as we're talking about agility, the agile culture.

In concrete terms, where should we start if we want to initiate a scaled agile approach?



Jean-Claude Delagrange

Jean-Claude Delagrange shares his experience as an Agile Coach within the IT department of France's Caisse des Dépôts (public sector financial institution), on the implementation of the "Banque des Territoires" platform.

When launching a new structure that was designed to establish a single banking platform irrespective of the type of customer and proposed product involved, the scoping phase and the definition of the MVP were facilitated by a team of three in-house agile coaches, with the assistance of two external coaches:

The Product Manager, the future director of this regional investment bank, was trained and coached at the same time as the teams.

A product vision was defined: originally, a slide taken from a board meeting launching the new platform (the seven promises of the platform), shared, and outlined with all teams (some created for the new platform and other pre-existing ones, comprising 80 people in total).

The backlog was defined from a blank sheet, in conjunction with “team”-based Product Owners (with the teams initially organized into Component Teams), and prioritized (by identifying the MVP right at the outset by asking: “In four months, what do you want to present to your customers?”

On this basis, we were able to establish a dynamic weekly PO meeting, under the responsibility of the Product Manager, and gradually consolidate the new backlogs and the existing backlogs.

The self-designation of Scrum Masters in the teams, complemented by new hires, enabled a community to be instituted, which very early on set up a weekly session for sharing experiences and syncing. They were the points of contact for implementing tools for managing the backlog and for agile project management.

Cross-functional roles were taken up by committed individuals, whose responsibilities included event logistics.



Alexandre Cuva

shares his experience at SoCraAgile as an Enterprise Coach for the Haute Horlogerie watchmaking foundation in Switzerland.

Just like always when supporting organizational transformation, we first took the time to fully understand the customer’s requirements. This point should not be ignored, as we’ve often met leaders who want to scale up a framework, more for tool-related reasons than because they have a real need. We take a pragmatic approach and believe that frameworks can always be adjusted to the needs of the organization as long as it’s in line with the agile manifesto.

In this case, our client had contacted us to help them implement SAFe 5 in their organization. After understanding their real need, we realized that, in reality, they were only attracted to SAFe in the sense of it being “flavor of the month” in that, if it’s been successfully implemented by another company, it surely must be the best framework! We finally proposed that Scrum@Scale be adopted (also known as Scrum of Scrums), allowing them to scale up as needed without adding further complexity to an already highly complex system.

The first step involved creating product-based teams that are independent of one another. This is important as we’re introducing a radical paradigm shift. The organization is no longer driven by “projects” but by “products” that are monitored over the long term. A Product Owner will be in charge of a product (a set of applications grouped under a common product); he or she will manage the strategic requirements, development needs, technical maintenance, and any incidents.

We’ll then start by setting up agile teams and increase their level of



experience. This is also important for the future of the project. Teams that don’t behave as Scrum teams will have difficulty scaling up later. If the “easy” part of the system cannot be adapted, then what’s the point of trying to scale up?

The first scaling tool in place will be the “Scrum of Scrums” (a kind of daily Scrum) and the “Scaled Retrospective.” With the aid of these two events, we are bringing greater transparency and continuous improvement between teams.

The rest will come in small steps, depending on the maturity of the teams and the system overall.

Laurence Hanot

Laurence Hanot shares her experience as an Agile Coach on a complex program involving around 250 people in the field of energy.



I’ll talk about this particular situation, where the business has been trying to deploy SAFe at the Solution level (or its own misinterpretation of this), without support and without success. It should be noted that this program is very complex,

in terms of both the number of components, interfaces, technologies, and different organizations involved and the geographical distribution of the teams.

The first step involved undoing the implementation that wasn’t working, simply by explaining to the people concerned that what they were doing (PI Planning for 15 people who plan specification activities and then monitor whereabouts they are each week in respect of those specifications—in other words, the Solution aspect from their point of view) was neither agile nor SAFe.

We then tackled the most obvious aspect for me at that time: the fact that multiple component teams needed to work together to deliver a solution. However, one-on-one interviews quickly helped me understand that not only were the teams not working side by side and didn't know one another, but also that each team was pursuing its own particular objective (to deliver its component) rather than a collective one (to deliver the solution).



We therefore set up weekly synchronizations involving representatives of those teams; this soon resulted in some interesting initial discussions aimed at trying to deliver and integrate the components on an as-and-when basis—since the teams mainly worked in Scrum but with different iteration durations. The first collaborative demos took place.

We then arrived at a consensus, following several weeks of negotiation, whereby the teams would apply the same cadence (iterations of three weeks).

Next, we organized a first “actual” PI Planning session, which took place in January 2019 with 80 people but few team members, as we hadn't been able to convince them of the benefits and importance of their presence. Despite many people's doubts, the PI Planning went well (at least for a first

session), not least because it resulted in some important issues being highlighted regarding a central team. Synchronization sessions focusing on visual management, retrospectives, and shared product demos have now become standard practice.

As a result of some decisions (or non-decisions) by management and the subsequent lockdown, a period of drift set in, in which little changed.

Anyway, over the past two months, this particular transformation has sped up and broadened with a new energy, thanks to a push from senior management, a common and shared vision, the creation and alignment of a transformation team, a leader-type manager who plays a part in that team, and some recent reinforcements in terms of coaching.

We've just launched—or rather, have formalized the existence of—three Agile Release Trains (at least four or five others will be launched next year, all already scheduled at the same iteration and Program Increment cadence) delivering three solutions (obviously, not on a one-to-one basis as there are dependencies). This means organizing three actual PI Planning sessions (with the developers present!), one pre-Program Increment and one post-Program Increment, and formalizing, training,

and supporting the key Train-related roles (the Release Train Engineer / Product Owner / SAFe Agilist trios) and solution-related roles (Solution Train Engineer / Scrum Master / SAFe Agilist), while supporting the teams where there are any gaps.

At the same time, a lot of work will be done for the end of this year to establish the portfolio: with a shared multilevel backlog, prioritization,

rituals, tools, etc., as there's a real shortfall at this level that's impacting the teams.

We're also expecting other projects next year: working on incremental deliveries, involving the customer to a much greater degree, setting up communities of practice, harmonizing and improving tooling, and so on.

What role do customers play in a scaled agile approach?

“



Jean-Claude Delagrange

A central one of course, as is set down in the agile values and principles!

Depending on the steps (defining the global feature backlog, the product-user stories backlog, sprint reviews, and testing), customers will be involved differently, with the overall organization based on the network of POs, coordinated by the Product Manager.

The difficulty is less about reconciling the views of different customers than gaining the agreement of those who represent them (digital marketing and sales teams). The other difficulty comes from integrating internal users, who have a more or less direct relationship with customers, and whose points of view may be opposed (with a multiplication of checks, data needed for back-office processing or for control purposes, and habitual ways of doing things).

At the stage when the global backlog is defined, the “digital marketing”

departments are involved from the outset in a UX-oriented approach and design thinking sessions; they jointly identify the features and develop the product strategy under the guidance of the Product Manager. Epics can emerge at this stage, but in the form of major themes.

All of the elements, even if detailed, have to be noted and saved for the next phase.

"The need is what the customer is missing in order to resolve their problem"

François Delivré

“



Laurence Hanot

Yes, it should be central! But even though it's sometimes difficult to involve customers with a Scrum team, it can be even harder in a context where customers are even "further" away from the teams.

SAFe incorporates several interesting concepts on this subject:

- the role of the Business Owner in defining the value of team deliverables from a user perspective;
- the measure of predictability in a PI, based on this value rather than on the velocity (which is of no use if you have a good delivery cadence but are not delivering the right product: "Do the Right Thing, Do the Thing

Right, and Do it Fast”, as per the Henrik Kniberg video “Agile Product Ownership in a Nutshell”—a must see if you haven’t yet watched it).



Needless to say, “customer centricity” has been placed at the heart of SAFe 5.0, but if you look closely, these are simply design thinking practices and so aren’t really anything new.

In any case, in a scaled agile context, it’s just as important for an organization to involve its customers throughout the

product development process so as to get their feedback: gathering their needs or bringing out a new product (the design sprint is great for this), regular demos, managing their feedback, delivering at the right pace, and so on.

And of course, to make sure that the key roles conveying the customer’s vision and voice are properly connected with both the customer and the teams: Product Managers, Business Owners, Epic Owners, Product Owners, etc.



Laurent Charles

The customer? The organization for which we’re doing all this and without which everything we do wouldn’t exist?

One thing is certain: if you forget to put the customer at the center, and

especially the value delivered to the customer—as it isn’t always possible to have access to the customer—you’ll fail. But this isn’t only true for agile approaches: whatever your approach, if you forget about the customer, you’ll fail.

On the other hand, having the customer at the center doesn’t mean you have to do everything the customer asks for. Keeping the customer at the center doesn’t mean saying yes to everything. Knowing how to say no—“no, not like that,” “no, I can’t do that,” “no, I don’t know how to do that”—is also a way of being loyal to your customer.

What new roles are needed in order to implement scaled agile practices?

“



Jean-Claude Delagrange

While avoiding simply repeating what existing frameworks specify (SAFe goes the furthest in this area), I’d say that it’s absolutely necessary to put in place roles focused on synchronization, whatever form or name it takes, and on each of the levels (product, customer approach, teams, architecture, go-live, etc.). In fact, the overall vision depends on this.

But implementation, which is based on execution-related tasks (organization of events, monitoring of consolidated workflows, consolidation of backlogs, etc.), will also require operational resources,

including secretarial/support roles, PMOs, etc., and this definitely shouldn't be overlooked.

My experience of two platforms was as follows:

- In the first, trust was placed in a secretarial pool that was treated as a full team, with its own backlog, producing value for all other teams, and gaining control of room bookings, meetings, support software, equipment for reviews, minuting meetings, ordering trays of food and drinks: in this case, every PI Planning session was virtually a party;
- In the other, lots of money was spent on hiring PMOs who didn't want to lower themselves to subordinate tasks and delegated them to the secretarial teams, resulting in postponements, absences, time wasted getting projectors to work, flip charts going missing, CRs coming in late, and so on; the first stages were tough.

“



Laurence Hanot

The larger the scale, the more you'll need to synchronize, apply cadences, coordinate, align people and practices, and delineate requirements at multiple levels and for a large number of people.

Whatever the approach undertaken or the underlying framework, it'll be necessary to “scale” the roles: with Super Scrum Masters or RTEs / STEs, Super Product Owners or Product Managers / Solution Managers, multilevel architects with different perspectives and backgrounds, and so forth.

Apart from the specific roles of Business and Epic Owners in SAFe, I'd say that every context will be different and that new requirements will emerge

during the scaling process that will need to be supported by existing roles (agile or not), and that this is part of adapting agile to each organization's situation.

I've often seen people—managers in particular—lose their way or feel reticent about an agile transformation because they don't fall into any of the roles described by the method. It's a good idea to take account of their apprehensions, work with them to take stock of their current responsibilities / activities, compare those with agile roles, and see what they can move toward, if they want to, or whether they can “invent” their own role in this new way of working.

What additional skills are needed in order to implement scaled agile practices?



Jean-Claude Delagrance

At present, “scale agilists” are pioneers; it should be possible to judge the expected results in the not- too-distant future.

Mastery of the tools and framework used is at least as important as the management skills of the people who occupy the roles of Product Manager, Release Train Engineer, and so on: strategic vision, a manager-coach approach, fully taking on board agile values, decision-making abilities and, above all, the ability to delegate decision-making, etc.

A good deal of questioning is necessary to get rid of the reflex behaviors acquired through the use of project methods such as V-cycle, waterfall; the

project owner-project manager divide, a detailed predictive vision of products, specifications before any development, keeping a distance from customers, and so on.



Laurence Hanot

I'd say, above all, soft skills: adaptability, humility, open-mindedness, the ability to listen, knowing how to delegate, the ability to take a global view, to take a step back so as to see the big picture (a systemic view)... And courage!



Laurent Charles

And let's add trust: the trust of the management team in their teams, the teams' trust in management, and trust in one another. But again, that's not specific to the agile approach. It's very general, as trust is more powerful than either hierarchy or processes.

Is SAFe really an agile methodology?



Laurent Charles

This is an issue that can lead to debate and discussion. From the enterprise's point of view, it's true that, in order to take a global view of several projects being performed simultaneously, it seems vital that the work involved is orchestrated on the basis of coherent processes and shared progress indicators. From the teams' point of view,

this notion of “structuring the work for everyone” might seem to contradict the principle of autonomy inherent in the agile approach. Keeping in mind the principle of adaptation in agile, we can say that these processes and indicators should be open to evolution and modification whenever teams are no longer adhering to them or are suggesting something else.

Is SAFe agile? I’ve asked myself the following question: couldn’t SAFe be a first step on the road to agility?

This is paradoxical, as SAFe is sometimes seen as the ultimate goal of a large, agile organization, and this is how it’s promoted.



However, SAFe isn’t a method to be adopted in blocks but rather a set of practices to be used as a basis. So, why not take from SAFe only those things that are relevant to the organization’s goals and culture; take what isn’t agile before then taking a first step toward agile when you come from a traditional culture; and take the things that are scaled agile, like the Train, when you come from a Scrum team culture?

After all, moving forward in this way by iteration is an agile approach.

“



Laurence Hanot

At first glance, SAFe looks like a diagram of a gas processing plant. And when you look more closely, it still looks like a very, very detailed and highly structured diagram of a gas plant.

When we attend an SPC (SAFe Program Consultant) training course, we hear the trainer bore us for hours with the principles and values taken straight from the Agile Manifesto and the House of Lean, twisted and expressed in different ways (the 10 principles, the 4 core values, the 7 skills, and so on). I use the word “bore” because when you’ve been an agile coach for nine years, it does seem to go on a bit.

JSO, let me be clear: the question is not whether SAFe or Scrum or “the” agile method is truly agile. The question is what we do with those approaches.

I’ve seen a lot of Scrum implementations that didn’t really involve agile at all. (In one example, “we measure velocity per person so as to eliminate the weakest”—a true story!) And it’s the same with SAFe.

Obviously, you won’t cook up a very good dish by simply throwing together the list of ingredients in a recipe. The key is in the way you implement the recipe, interpret it, the way you taste it as you go along so as to adjust it, the way you add or remove an ingredient or spice, and the time and love that you devote to it with the aim of making people happy.

Is SAFe truly safe and secure for my organization?

“



Jean-Claude Delagrance

The strength of SAFe lies in organizing things in a secure way by integrating both traditional hierarchical roles and new roles, thereby ensuring both horizontal coverage and production synchronization on a non-hierarchical basis. And in doing this while setting strong, even restrictive, rules guaranteeing the effectiveness of the system as a whole.

The security of this way of organizing depends greatly on the human factor: in other words, both the willingness of those in the traditional roles to accept and make room for those in new roles and a good alchemy between the men and women who'll be members of the teams.

The main risk is that the weight of the organization generated by the framework slows down teams and makes decision-making more cumbersome, especially since it's difficult to ensure that all managers will truly play the agile game.

“



Laurence Hanot

From my point of view, SAFe doesn't define any hierarchical roles. Rather, its rationale lies in its provision of a virtual organization (“the second operating system” in SAFe 5.0 terms) to deliver products and solutions without changing the structure of the department or enterprise.

The risk is therefore precisely that it won't go far enough, and it won't help the enterprise to challenge its ecosystem or its own actual structure. However, I don't know of a scaled agile approach that does this.

Another risk, but one shared by other approaches, is that of applying the framework "by the book" without adjustment or support, in a wholesale and unreasonable way. I haven't yet encountered this on the ground.

If we've never used Scrum techniques, can we still implement SAFe?



Jean-Claude Delagrance

Yes, absolutely. Provided you're fully aligned with agile values, it's possible to acquire the basis foundations quickly, for the smooth rollout of events, of Scrum milestones, of the language and the concepts: certification training is available, which should be completed by a good coaching team.

The difficulty will be greater for those taking the roles of facilitators, coaches and, of course, RTEs ("Scrum Master of Scrum Masters").

Personally, I'd never used the Scrum framework before developing my skills as an agile coach.

“



Laurence Hanot

It's a question I asked the trainer during my SPC training since, in the Roadmap Implementation, “Waterfall / ad hoc agile” is written in the very first box (and therefore seen as a starting point). I don't know what “ad hoc agile” means, but that was the only time in four days of training that I felt that Andrew (who was otherwise an absolutely brilliant trainer) was expressing doubt, as he answered that starting with purely waterfall-oriented teams seemed difficult to him and that deploying a minimum set of agile practices for a few months before starting would be better.

I believe that too, as it seems to me that it's too big a step to embark on everything at once and such an approach risks being abandoned or rapidly reversed. I would say that getting started with agile practices within teams for at least four to six months before scaling up is a minimum requirement.

“



Laurent Charles

Because agility, at scale or not, is primarily a culture, I feel that implementing SAFe without having experienced agile approaches—whether Scrum or anything else—is more complex. The important thing is to have practiced the principles and values beforehand.

Is a software application needed in order to take a scaled agile approach?

A tool without a process is ineffective.
A process without a tool is inefficient.



Laurent Charles

As we've already noted above, for 36% of people, the implementation of a tool shared by all teams is a key factor in the success of a scaled agile approach.

For me, the answer to this question is the same as for agile on a smaller scale. It's the same as "do we need a tool to do Scrum or Kanban?"

I always respond to our clients: "Not at the start. But yes, very soon, you will." A tool without a process is ineffective, while a process without a tool is inefficient.

No, because agility, whether on a small or large scale, first and foremost involves a culture, a set of values, which can be transmitted through games, simulations, failures, and attempts of varying degrees of conclusiveness.

Then, when the people involved all share the same mindset, the software tool becomes indispensable: in order to work and manage projects remotely, aggregate information, have better

36%

Implementation of a tool shared by all teams

40%

traceability and history of the project life cycle, automatically have charts that are helpful

Are you among the 40% of people who use Excel or Calc spreadsheets to keep track of their projects? For scaled agile, there are more suitable tools

in monitoring the project, and so on. And because the Post-its are falling off the board and are put back at random by the person who comes to do the cleaning! We ourselves have witnessed this : -)

“



Laurence Hanot

I very much like the point made by Enalean: “Not at the start. But yes, very soon, you will”. That’s exactly right. I’m not talking about tools to facilitate the developments that seem essential to me (source manager, code reviews, continuous integration, etc.), but about tools to manage backlogs, spreadsheets, indicators, and so on.

Visual management is awesome but has its limitations in a scaled context, while remote working has, I’m afraid, buried this practice for a while.

Tools are therefore essential in a scaled agile context but should support practices and principles rather than constraining or limiting them.

What should we do if our teams are already using different tools?

“



Jean-Claude Delagrange

You'll need to very quickly consider migrating all of the workflows to a single tool.

The difficulty here will depend on the degree of discrepancy between teams, whether in terms of the tools used or, in particular, in terms of workflow sophistication and the detail of the data involved—as some people will use data for different purposes such as the test automation repository.

It's important that synchronization can be carried out within a single framework, with continuity between the modules used by teams and those used for coordination; along with syncing of the USs, consolidation of the USs into Epics, as the customer journeys and dependencies between backlogs will require this.

The option of extracting an Excel file from an old tool so as to re-include it in a new tool after reformatting is relatively straightforward, with a dedicated role (an expert in the new tool) given responsibility for this, and with support (training or mentoring). However, teams are reluctant to waste time on the transfer (especially if data (re)entry is involved), adopt new standards, modify their workflow, unify their metrics, and sometimes lose information (such as their initial numbering system or their archives).

The need for psychological preparation by the coaches should not be ignored and must be part of the support package.

The consolidation rules and the results expected for coordination purposes must be established with the management team and the product manager, but must also involve the teams themselves, since it's their working tool that's at stake. All this will enable the PMOs to play a part again.

“



Laurence Hanot

This is the case with the program I'm supporting at the moment. We've overcome this issue by defining the “master” tool that contains all of those items—Business Epics, Enablers, Capabilities, Features, and User Stories—that describe and break down the requirements from the portfolio into Trains / teams, with hyperlinks in the User Stories to the element(s) of the teams that aren't using the same tool (and that cannot migrate).

And I agree with Jean-Claude that, beyond the tool issue, the greatest difficulty lies in unifying the workflows and therefore, by implication, the practices followed by the different teams and people, even in terms of their understanding of agile.

What should we do if our teams have different workflows?



Jean-Claude Delagrance

Workflows can remain different in detail as long as they involve shared stages that correspond to shared milestones or to project management requirements. The “Definitions of Done” can also be different at the outset but will gradually have to get closer to one another.

The first action to take is to bring the teams together to agree on a workflow that, as a minimum, allows for consolidation and consistent metrics. The efforts to achieve these goals will need to be clearly displayed and integrated into the team’s workload (an investment that can have an impact on the velocity).

In my experience, teams with the most complex workflows tend to try to maintain these. However, as a minimum, the milestones can be aligned with the workflow of those teams with the simplest one.

This also provides an opportunity to give ownership and a role to the PMOs.



Laurence Hanot

As I said earlier, different workflows are generally an indication of different practices and ways of working. These are the things we need to focus our attention on and see whether or not they are impacting on the degree of consolidation at a higher level.

From my point of view, workflows at the team level can be different and, indeed, we can even have teams using Kanban in a scaled context. However, if we're in a situation involving multiple Trains, multiple solutions, and portfolios, all with a lot of dependencies, it seems important to harmonize workflows so as not to complicate the view and timing of the overall project (in terms of Kanban and backlogs at the Portfolio, Solution, and Program levels).



Laurent Charles

In an industrial context involving multiple business units (even if the activities are purely software- focused), workflows will always be different (unless the workflow is so generic that it doesn't mean that much in the day-to-day work done by the teams).

Our experience shows the importance of having team synchronization points. And such synchronization must, of course, not only be synced to the specifications and deliverables, but has to continue for the entire life cycle of the teams involved.

An approach like the SAFe Train is interesting because it forces a synchronous rhythm between different teams.



The 10 points to remember...

1. Be aware that scaled agile is a journey and not a destination.
2. Accept that you'll need to unlearn your old ways of working and change your mindset.
3. Involve all stakeholders including the senior management team.
4. Focus on the satisfaction of your customers, both internal and external.
5. Choose an inspirational model but don't refrain from adapting it.
6. Mobilize the early adopters.
7. Facilitate collaboration between teams with the aid of shared tools.
8. Give project teams real decision-making power.
9. Communicate and convince virally, on the basis of your successes.
10. Focus on training and coaching.

To continue making progress

We have provided you with the following resources to help you move forward on your scaled agile journey:

Webinar

The banner features the Tuleap logo and the word 'WEBINARS' in orange. The main title is 'Scaled Agile: Scrum of Scrum, SAFe®, LeSS...'. On the right, a diagram shows a 'Scrums of Scrums' box at the top, connected by lines to three boxes below: 'Agile Team 1 (PO, SM, Devs)', 'Agile Team 2 (PO, SM, Devs)', and 'Agile Team 3 (PO, SM, Devs)'. At the bottom, it says 'Join our next Webinars!' and 'www.tuleap.org' with a mouse cursor icon.

[View all the webinars](#)

Client case study



[!\[\]\(41ce11edeec1381a4e9c966de16a76b8_img.jpg\) Vitam: How 3 French ministries build an app to manage large volumes of data with Scrum of Scrum](#)

About...

Tuleap and Enalean

Agility is a rewarding journey, with its ups and downs and the way it continually evolves. We'll be by your side throughout your scaled agile journey.

Laurent Charles, CEO Enalean

Tuleap is the leading Enterprise solution for Agile Management and 100% open-source DevOps. Within just a few years, the French solution has won over large corporations and emerged as an alternative to better-known tools such as Azure DevOps, the Atlassian-Jira suite, HPE ALM Octane and CollabNet VersionOne. With stronger “business customization” capabilities than its counterparts, Tuleap helps organizations modernize their project management processes and software development chain by favoring agile and DevOps approaches. Customers include CEA, STMicroelectronics, the French Directorate General of Armaments (DGA), Orange, JTEKT-Toyota, and Saft-Total. In a situation in which IT priorities are being reviewed in 2020, Tuleap’s three main benefits—its all-in-one solution, advanced customization, and open-source qualities—make it a product that can bring a rapid ROI.

Enalean is the publisher and developer of Tuleap.



Tuleap and Enalean also embody the concept of the [Open Roadmap™](#), a unique service offering based on open innovation and customer collaboration.

SoCraAgile

SoCraAgile Sàrl was set up in response to its founders' observation that today's digital organizations need ingredients that will trigger innovation, ingredients that are usually brought in by high-profile talents responsible for the success of their organization. We are based in Nyon, a small town in Switzerland's Vaud region, located on the lake shore between Geneva and



Lausanne. We carry out most of our work with clients in the French-speaking part of Switzerland.

We are firm advocates of an approach that combines the best elements of agile, lean management, software craftsmanship, and DevOps. Such a culture can be applied to an entire organization and can impact on multiple business units that collectively contribute to its success, such as IT, business, human resources, marketing, and more besides.

SoCraAgile's team can draw on more than twenty years' experience in the fields of technical excellence, agile lean coaching, and the agile transformation of organizations.

Zenika

Founded in 2006, Zenika is a consulting firm specializing in technology, management, and organizational innovation that supports organizations with their digital transformation.

Zenika's ten offices are located in Bordeaux, Rennes, Grenoble, Lille, Brest, Nantes, Lyon, Paris, Montreal, and Singapore. We offer services covering ten areas of expertise: agile, big data, consulting, Dev, DevOps, design, AI, labs, and security.



zenika

< animés par la passion >

Zenika is an anagram of “Kaizen,” the Japanese term for “continuous improvement.” This philosophy, combined with the three fundamental values of transparency, sharing, and friendliness, enabled the company to earn top place in the Work® France awards for organizations with 50–500 employees in 2018, the fifth time it had participated in the initiative.

What distinguishes Zenika is its desire to disrupt an IT consulting firm's standard business model in its management practices by becoming a responsible company, regularly training its employees, paying attention to the things that motivate each person, giving them time, and encouraging the establishment of “tribes” so that teams can be kept on a human scale. With sales of €40 million in 2019 and a workforce of more than 400 people by the end of 2018, Zenika is committed to always offering an environment that encourages initiative and entrepreneurship.

The vast majority of our consultants also act as trainers within our Training Division, whose portfolio of services is constantly refined in line with technological developments. In September 2018, the Zenika Academy

opened its doors to offer a program of training dedicated to those seeking professional development in open-source technologies and innovation-related roles. The training program is based on our culture of transparency, friendliness, and sharing. Zenika's consultants also have access to "technoZaures," which are in-house conference events; "Lunch & Learn" sessions, which are open lunches focused on a topic suggested by a particular employee; and "NightClazz" sessions, which are Zenika meet-ups that consider a topic in depth for several hours. Zenika also organizes "Learning Expeditions" and takes part in national and international conferences.

It is an outward-looking company that supports a number of communities and dedicated client events (morning sessions and "What's Next" events).

We support large organizations in their agile and digital transformation by training and coaching their teams and implementing an agile enterprise approach inspired by the best practices from the SAFe, LeSS, DAD, and Spotify frameworks. Here, the aim is not to apply these models dogmatically but rather to gain inspiration from them and adapt them to the client's context and history in a process of continuous learning and improvement.

Finally, we work with our clients to co-build their products, from launching MVPs (Minimum Viable Products) and implementing our Zenika Labs approach, for example on Blockchain projects, to working on large-scale projects, such as the creation of data lakes, with the establishment of feature teams or squads.

Brands mentioned in this book

- SAFe® and Scaled Agile Framework® are registered trademarks of Scaled Agile, Inc.
- Tuleap®, Tuleap Enterprise Edition®, Enalean®, and Open Roadmap® are registered trademarks and logos of Enalean SAS.
- Zenika and its logo are registered trademarks of Zenika SARL.

Other product or service names, logos, and brand names are the property of their respective owners.

Copyright © 2021 Enalean, Jean-Claude Delagrange, SoCraAgile, Zenika.

e-book v1.4—May 27, 2021

Published by Enalean SAS, 87 Avenue de Chambéry, 73230 Saint-Alban-Leysse, France.

Tel.: +33 (0)9 81 41 96 16—Email: contact@enalean.com

Business identification (SIRET) number: 53179662100047