EB

Elektrobit

Scaling agile methods without being dogmatic

There are many theoretical models for implementing agile development principles in organizations. These models often sound logical in theory and quickly tempt you to use them without any modifications. However, these models often describe ideal scenarios that do not exist in reality. This article describes considerations when adopting and scaling agile methods that will allow you to successfully realign your organization.

The number of theoretical models for scaled agile approaches has risen steadily over the past few years, especially for scaled Scrum and Kanban processes. There are many models such as SAFe [SAFE], LESS [LAR16], Nexus [NEXU], SoS [SUT01] or Kanban flight levels [LEAN] (see Figure 1). But does theory actually always equal reality in your organization?

There is a risk that these models are used too dogmatically.
What I mean by this is that you are so convinced of the specific model you have chosen that you try to restructure your company such that you can use the

model - regardless of the consequences. This article shows why you should avoid dogmatically introducing a scaled agile model.

As scaling agile approaches are, in most cases, initiated by management, we will assume, for this paper, that they have been management driven. Experience has shown that teams that introduce such a model always require management level support.

If you have decided to introduce a scaled process model, I recommend that you consider the following aspects before applying the model.

Figure 1: Models for scaling

Models – wrong, but useful?

The British statistician George Box (1916–2013) once said: "Essentially, all models are wrong, but some are useful." This statement allows us to identify two key aspects for scaling agile methods that help us assess the models:

1. Know the models' elements
2. Use those elements that are useful to you

The first aspect states that we need to know different models first before we can evaluate their differences and assess their benefits to the company. Therefore, at management level, identify the reasons why you favor a chosen model. This will also help you explain the model's introduction to your employees.

Dogmatically introducing the model would mean that you decide on a model and intend to use almost all aspects of it as described in theory. However, this approach is an extremely unfavorable starting position for a transition with the objective of introducing agile methods throughout the company. The Status Quo Agile Report 2014 [KOM14] shows that only one quarter of those interviewed actually think that they use an agile approach as described in theory (another 15 percent still use a classical approach). In this context, agile process models do not mean scaled approaches but the process models at the team level. A scaled approach requires more synchronization, coordination, artifacts, events and roles than a model at the team level, which increases the percentage distribution of hybrids. You first need to find the theoretical model that is fully tailored to your company's needs.
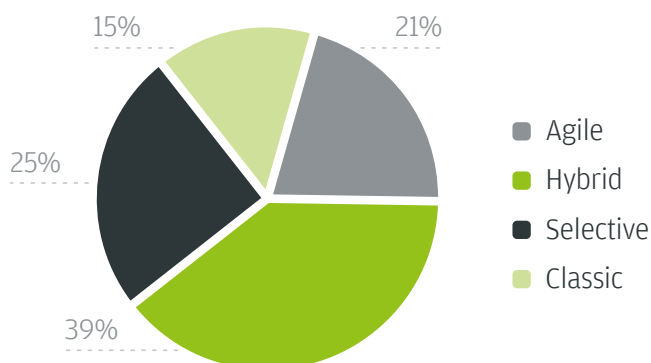

Figure 2: Status Quo Agile Report 2014 [KOM14] survey result

This does not mean that we want to rule out theory from the start, but you should be aware of the following: Experience tells us that every second agile approach is a hybrid of different theories.

Therefore, it is important to be aware of other approaches, and I encourage participating in agile discussion groups and regional meetups. Any kind of lessons learned from across company boundaries can provide you with additional insights.

Agile Manifesto

The Agile Manifesto [AGIL] forms the basis for agile processes. Follow its principles when introducing the model to lead by example. The manifesto shows us that the values on the left-hand side have priority over the ones on the right-hand side (see the highlighted Agile Manifesto key statements). However, that does not mean that the right-hand side is not important.

Individuals and interactions over processes and tools

The dogmatic introduction contradicts this statement. If we introduce a model dogmatically, we focus on the model (the processes and tools) and not on our employees and interactions in the company. If the introduction fails to consider individuals and interactions in the company, you will see that the employees will not live the Agile Manifesto values. As has already been mentioned, one management principle should be to always lead by example.

Working software (products) over comprehensive documentation Customer collaboration over contract negotiation

At first glance, these two statements appear to have no relation to the scaling of agile methods. Nevertheless, they do have an indirect impact on the change. Despite the transition, companies must supply products that work. In addition, they need to make sure that the transition process does not noticeably affect customer collaboration.

My experience has shown that customers very often also influence internal processes, especially when customer collaboration is intense or intended to be intensified. However, customer availability is often limited. This makes dogmatic approaches involving extensive planning meetings or reviews not as feasible as desired since it is not possible to get direct customer feedback. Can customers really participate in a two-day planning meeting (e.g., Program Increment Planning, see [SAFE]) or do they need representatives in the change process who take their role and deputize their position?

Responding to change over following a plan

Be open to change from the beginning of your transition. Follow a plan without being dogmatic. A wise old saying from the German-speaking area teaches us: "Planning replaces coincidence with error."

Without planning, everything we do is coincidence or defined by others. The consequences of our actions cannot be assessed before we follow a plan – we can be right or wrong. If we find out we were wrong, we need to change our plan. This can also mean changing the process model we have chosen.

Don't regard this as a failure. View it as a new experience and continue to change your plans based on the new insights you have gained. Take this comparison between planning and reality as your first opportunity to learn for the future. Thomas Edison famously said he didn't fail: "I've just found 10,000 ways that don't work." This is the attitude that today's companies should have and view errors as merely a detour on our way to a successful scaled agile approach.

The Agile Manifesto describes important principles you should follow if you want to introduce scaled agile methods. In particular, it opposes the introduction of methods based on a theoretical model as this can only serve as a basis for a plan, discussions with customers or a first analysis of the processes. Such a model allows you to roughly plan the way, but what your scaled approach ultimately looks like cannot be defined at the outset. Rather concentrate on what you want to achieve with this change.

Incremental experimental introduction

In a study conducted by McKinsey (see [KINS]), only 30 percent of organizations achieve their initial change targets. Dogmatically adhering to a scaled agile process model results in a 70-percent probability of not attaining these targets. Incrementally introducing a scaled agile process model supports us in minimizing this risk. Lean Change Management [LIT14] can help incrementally introduce changes with small experiments and establish them on a small scale before their company-wide rollout (see Figure 3).
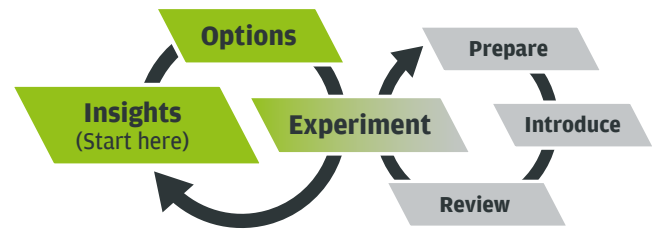

Figure 3: Process steps in Lean Change Management [LIT14]

The experiments are supported by one or more experts for agile processes, which allow the team to learn the iterative approach from them. If the experiment was successful, not only the expert but also the pilot team should support the further rollout. The diagrammatic representation in Figure 4 shows how a designated target area can be reached using an iterative approach. It is important to build a team of experienced persons from your company and experts who drive the scaling process. If you want to make sure that the scaling has top priority, this team's only task should be to introduce and support these experiments.

What fits here is the Shu-Ha-Ri model, an Asian martial arts concept applied to software development by Alistair Cockburn [COC06]. Shu-Ha-Ri also translates as learn-break-create (see Box 1).

> During the **Shu** phase, we learn and follow the process model's individual practices exactly as conveyed by a "teacher". During the **Ha** phase, the teams start applying variations to given practices and adjusting them to their own situation. The teams slowly detach themselves from their "teacher". During the **Ri** phase, the "students" develop their own practices.

Box 1: A. Cockburn, Shu-Ha-Ri model

This suggests that the teams' transition from the Shu to the Ha phase gives a new and different impetus from theory and practice. This does not necessarily mean that these practices are from the process model you have chosen. However, your theoretical model will be modified to a greater or lesser extent at the latest when the teams reach the Ri phase and go their own way. Make sure to check whether these modifications are still in line with your objectives or whether you need to adjust your plan.
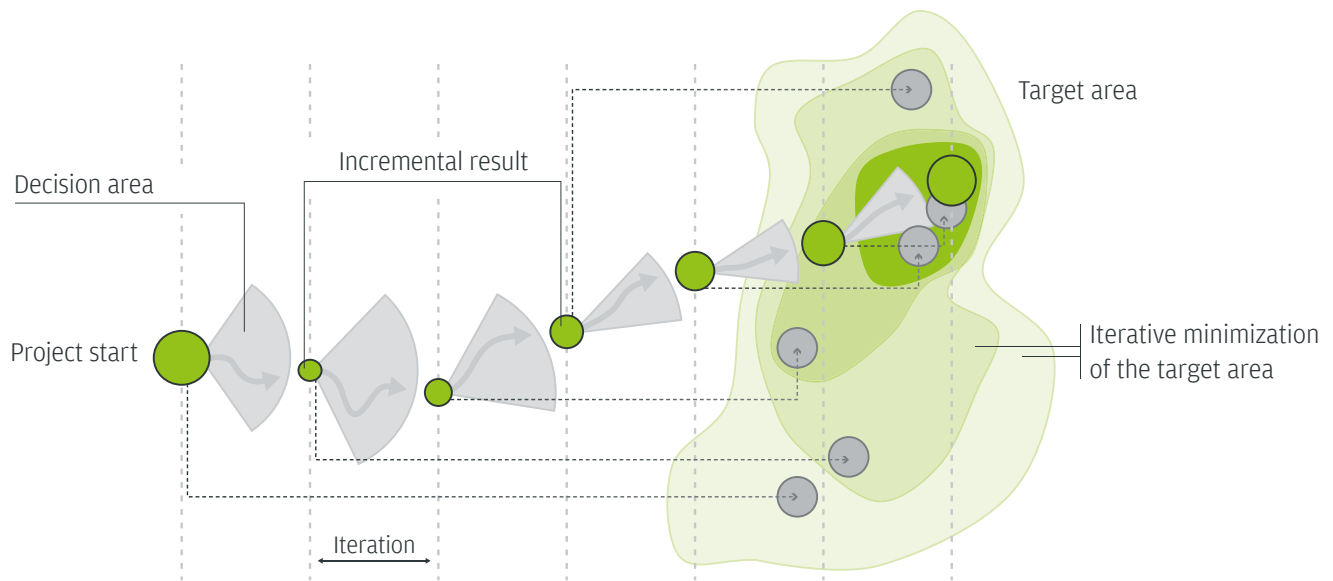
Figure 4: Incremental introduction and minimization of a target area

System view

By introducing a scaled approach, you will want to change your product development for your entire company or a part of it. To do this, you need a clear understanding of the system in which the change is made. If you focus on optimizing one aspect without considering dependencies and side effects, the change can result in a degraded overall system [RUB12].

Does the theoretical model actually solve all your problems and their causes or only some of them? Identify gaps in the model and ways to close them.

If practices of your model are already being used, verify that they produce the desired outcome and do not cause any side effects. There is a risk that the model merely eliminates symptoms. However, the actual goal is always to eliminate the cause of the problems.

In lean production, there is a practice called GEMBA (Japanese for "the place where production happens"), which means "go and see for yourself". On site is the only place where you can collect unfiltered information and analyze for yourself whether your theoretical assumptions match reality. Especially in a process where, sometimes, the goal is profound change, you must check whether your assumptions are actually true.

As a manager, don't just rely on the reports you get. Take the time to let your employees show you how they work or try to understand their problems. This gives you greater insight into the individual areas, which allows you to integrate them into your overall view. You should not only attend the team's presentations, but also visit the members during their everyday activities. What keeps you from being directly with the teams for half a day per week and receiving first-hand information?

Keeping your objective in view

The introduction of a specific process model should never be the objective of a transition. Introducing a process model can only be one step towards achieving an overall objective such as shorter installation cycles and therefore better market opportunities.

Constantly remind yourself of the actual objectives. Continuously communicate these goals to make sure that you reach every employee. Focus on few but clear objectives.

This means visualize your objectives and publish them in the organization so that the objectives are always kept in view [COV13] and the right decisions are made.

Give the employees the opportunity to provide feedback on the practices used. This reality check from innovation [MOR14] gives you different views. It provides you with a clearer picture of where you are on the path to achieving your objectives.

Values and culture

Sustainably changing the way an organization works, especially toward agile values, always means changing the corporate culture.

By dogmatically introducing a scaled agile approach, you will not automatically change a company's culture. In many areas, you will need to take time to enable experienced employees to learn the practices and the process model.

In his keynote, delivered at the 2014 Scrum Gathering in New Orleans, Ken Rubin described the passive attitude of many employees around change. A team member once told me: "This is just another process being introduced. We've already had this happen several times. Just keep calm and nothing will change for us." That's when I realized that more work is required with the people on the team so that they change their attitude and values and recognize the value added by the change. We have to give all employees the time they need to get used to and accept the change.

This shows that a dogmatic approach does not allow us to engage all employees of a company. The reality is that we need to address the employees' needs and wishes, and try to solve their problems. This takes time and close collaboration with the employees to empower them to master the required practices and understand the reasons behind them. We must enable them to learn that the aim of the practices is to help them and that the practices are up for discussion if they are not useful. In this way, they constantly change their values and culture. In particular, we must reassure the employees that their jobs are secure, even if the new process makes their jobs more transparent. The only thing that changes is the responsibility. They must transition from experts who protect their knowledge to experts who freely encourage the dissemination of knowledge in an organization. The managers in your company must ensure that these actions are included as criteria in performance reviews and that not only the individual's pursuit of excellence is assessed.

Another process you need to influence is the recruitment of employees. Discuss the future values you strive for and your company's culture with the candidates. The reason for this is that companies attract individuals who share the company's values [LEN12]. When you are transparent about values and culture, candidates know what they sign up for when working for your company. You will not win over those who are not willing to go

along with your company's transformation. But you will find those who will commit and contribute to the transformation.

In his book, Lencioni writes that organizations exploit only a fraction of the knowledge, experience and intellectual capital that is available to them [LEN12]. In many transitions, I was often told that agile had already been used "in the old days" before the company became big. This means the employees of your organization already know some of the practices and would like to reintroduce them – which is often prevented by processes that are too rigid. Try to include this experience in your transition and ask your employees for their opinion. You will find that most of them are willing to make improvements to their work.

Technical factors

The scaling often neglects the influence of technical factors. In most cases, the theoretical models are based on a greenfield situation, which means there are hardly any legacy issues. However, legacy issues almost always exist. You usually have an existing product with an evolved organizational structure, program code and IT infrastructure.

This means: If you dogmatically follow a theoretical model, it is very likely that you will not be able to profit from it from the start and reinforce your decision in favor of the model. In agile software development, we let tools make our everyday life easier – we do not allow them to dictate it. Due to this, we achieve shorter cycles for testing, developing and installing our software. This requires creating the necessary structure and technical basis. You need to lay the technical foundation in order to actually achieve the short cycles required by the theoretical models. You must allow room for compromise and resolve, and revise your legacy issues. We can conclude that the dogmatic introduction of the model is not possible. This means that the introduction can only be implemented either in entirety or not at all. Therefore, you not only have to plan the introduction of a process model, but also revise your technical foundation, which can differ for each product and team. This, in particular, requires that you involve your employees and keep in mind that this too will take time.

Unfortunately, taking a source code file with external dependencies and more than 1,000 lines of code that has evolved over years and splitting it into several source code files or classes based on responsibility to

make it more maintainable and readable is not something that can be done overnight. This example shows that we have to pay off this technical debt. Include this in planning the scaling as it can affect the overall system.

Systemic misjudgments

Does an expert for the theoretical model provide support? Have you attended trainings on how to introduce the theoretical process model and do you now feel like a company-wide expert? Do you support or even manage the introduction? If so, be aware of systemic misjudgment: overestimating yourself and confirmation bias. Many aspects that are implicit in models can only be understood when one has worked in an agile environment for a longer period of time.

If you overestimate yourself, you tend to put your own knowledge above that of others. As a result, you could miss important information from other employees or demotivate employees if you do not give space to other people's opinions. However, do not confuse this overestimation of your abilities with the determination with which you perform a task.

Moreover, monitor how you interpret your collected data and information. Confirmation bias leads persons to interpret data and information such that they meet their expectations. To be successful, it is not sufficient to analyze the data that meets your expectations. You also have to visualize and assess the data that is most unfavorable – for you and the project.

One agile innovation approach is to quickly verify assumptions by different feedback (see [MOR14]). Organizations such as the armed forces never analyze data that supports organizational goals. They assume that the worst case scenario takes place.

These systemic misjudgments of our brain show us that dogmatically adhering to a theoretical model can influence our decisions towards interpreting everything such that the outcome we desire will come true.

As a company, strive for transparency. Don't hide your conclusions from your employees. Give them the opportunity to understand your decision-making process and provide feedback. This gives you an external view by people who know your company. In addition, you can verify whether your conclusions match reality. If you find that employees are not happy with some of the decisions, have them describe these decisions in

greater detail and invite them to enter into dialog. This is the only way that allows you to see which side made a misjudgment and therefore take corrective action.
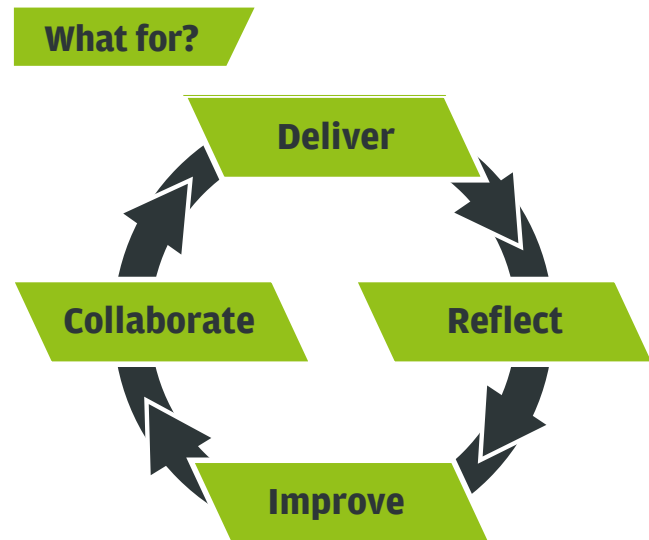
Figure 5: "The Heart of Agile" by A. Cockburn [HEAR]

As mentioned at the beginning of this paper, there are many scaled agile models you can introduce in your company. Therefore, always ask "what for?". Analyze not only the theoretical model, but also each meeting, each method and practice used. Analyze why the roles prescribed by the model exist.

In his article, "The Heart of Agile" [HEAR], Alistair Cockburn states that even agile process models have become overly decorated. This made me realize that the goal is not to introduce a practice but to first understand it: "For what purpose do you want to introduce the practice, process model or meeting?" There is no dogmatic approach that leads to success – we need to understand which cause of the problem we want to solve and approach the solution by using the four key concepts. This must be done in a repeating cycle (see Figure 5). According to Cockburn [HEAR], the following key concepts are at the center of each agile development: Deliver, Reflect, Improve, Collaborate.

When scaling agile process models, make sure to implement the four key concepts when introducing the model. This makes it independent of the practice you are actually using. It is important that the practice used promotes and supports the key concepts. This moves the discussion about which practice should be used away from a dogmatic discussion and into a discussion about how to best promote the four key concepts.

Summary

It is essential to recognize that theoretical models provide us with a starting point for scaling agile processes. Use theoretical models. Get to know and use them. However, keep in mind that there is no model that works for all projects and organizations. Don't decide too early on a specific model. Use the models as long as they are useful and produce noticeable improvements.

From my personal experience, I encourage you to avoid theoretical discussions about the pros and cons of the individual models. Instead, discuss the practices always within the context of your organization. At the end of the day, is it really crucial whether you use Kanban or Scrum? Find your own way. Like Steve Jobs said in his 2005 Stanford commencement address: "Your time is limited, so don't waste it living someone else's life. Don't be trapped by dogma – which is living with the results of other people's thinking."

Set a goal for your transition and follow it. Include your employees in the scaling and give them time to solve legacy issues. This will motivate your employees to be a part of the change and you will get a model that is tailored to your company and your company's needs.

References & links

[AGIL] A. Cockburn et al., The Agile Manifesto, see http://agilemanifesto.org/

[COCO6] A. Cockburn, Agile Software Development – The Cooperative Game (2nd Edition), Addison Wesley Professional, 2006

[COV13] S. R. Covey, 7 Habits of Highly Effective People, Simon & Schuster UK, 2004

[HEAR] A. Cockburn, The Heart of Agile, see http://heartofagile.com/

[KINS] C. Aiken, S. Keller, The irrational side of change management, see http://www.mckinsey.com/business-functions/organization/our-insights/the-irrational-side-of-change-management

[KOM14] Prof. Dr. A. Komus, Status Quo Agile, BPM Laboratory at the Koblenz University of Applied Sciences, 2014

[LAR16] C. Larman, B. Vodde, Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum: Successful Large, Multisite and Offshore Products with Large-scale Scrum, Addison Wesley, 2008

[LEAN] K. Leopold, Kanban Flight Levels, see http://www.leanability.com/de/richtig-abheben-die-kanban-flight-levels/

[LEN12] P. Lencioni, The Advantage: Why Organizational Health Trumps Everything Else In Business, John Wiley & Sons, 2012

[LIT14] J. Little, Lean Change Management: Innovative Practices For Managing Organizational Change, Happy Melly Express, 2014

[MOR14] L. Morris, M. Ma, P. C. Wu, Agile Innovation: The Revolutionary Approach to Accelerate Success, Inspire Engagement, and Ignite Creativity, John Wiley & Sons, 2014

[NEXU] K. Schwaber, Nexus, see http://www.scrum.org/Resources/The-Nexus-Guide

[RUB12] K. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison Wesley, 2012

[SAFE] D. Leffingwell, SAFe – Das Scaled Agile Framework, see http://scaledagileframework.com/

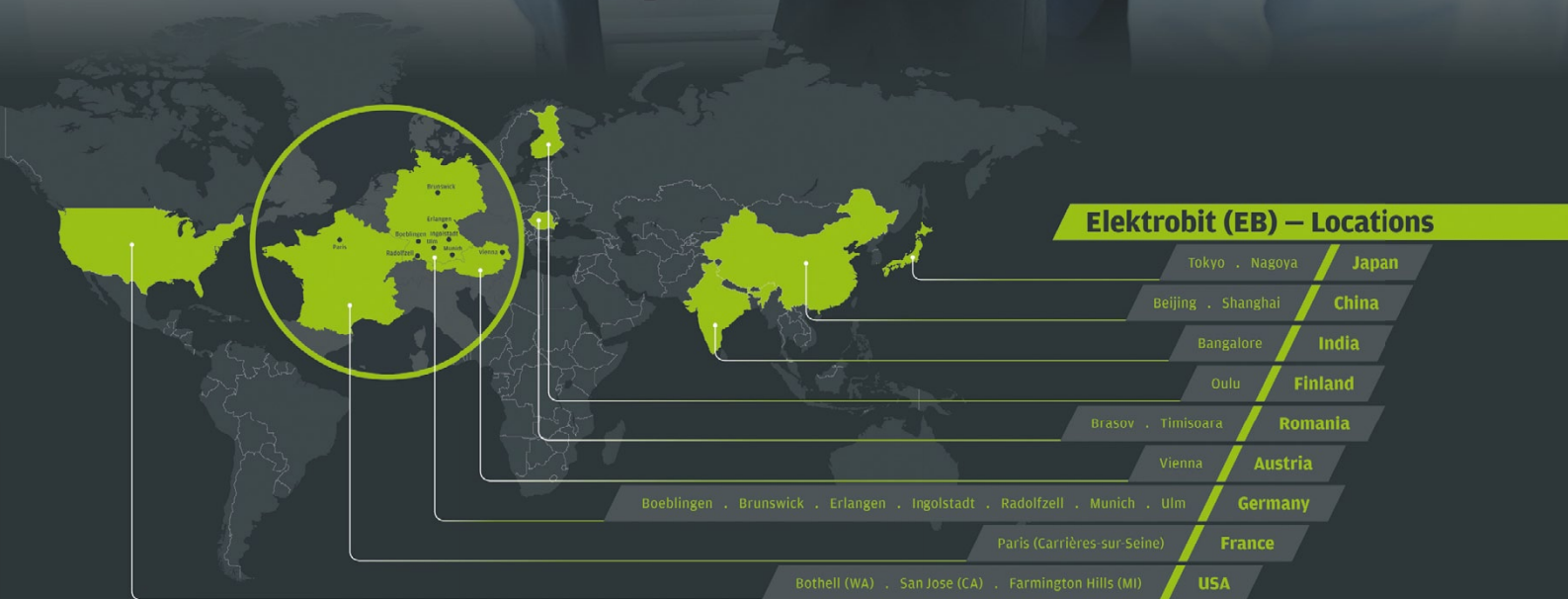[SUT01] J. Sutherland, Agile can scale, IT Journal Vol. 14 No. 12, 2001

Author:
Martin Hillbrand
(martin.hillbrand@elektrobit.com) is an expert for agile processes at Elektrobit Automotive GmbH. As a Scrum Master, Agile Coach and Software Craftsman, he has supported several companies in their transition to agile processes for eight years.

Notes

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

▶ Scaling agile methods without being dogmatic ....

....

....

Elektrobit (EB) – Locations

Tokyo . Nagoya	**Japan**	
Beijing . Shanghai	**China**	
Bangalore	**India**	
Oulu	**Finland**	
Brasov . Timisoara	**Romania**	
Vienna	**Austria**	
Boeblingen . Brunswick . Erlangen . Ingolstadt . Radolfzell . Munich . Ulm	**Germany**	
Paris (Carrières-sur-Seine)	**France**	
Bothell (WA) . San Jose (CA) . Farmington Hills (MI)	**USA**	

About Elektrobit (EB)

EB is an industry leading supplier of automotive software and has a proven record in embedded and connected software development for over two decades. As a dedicated partner to the automotive industry, EB provides technologies and flexible software platforms, tools, and services to help automotive manufacturers and their suppliers to deliver the best products and services in order to meet the needs of drivers.

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
sales@elektrobit.com
www.elektrobit.com