



SAFe in the Trenches

Collected by - Yuval Yeret, SPCT | PST

Yuval@AgileSparks.com

<https://www.linkedin.com/in/yuvalyeret>

Twitter: @YuvalYeret

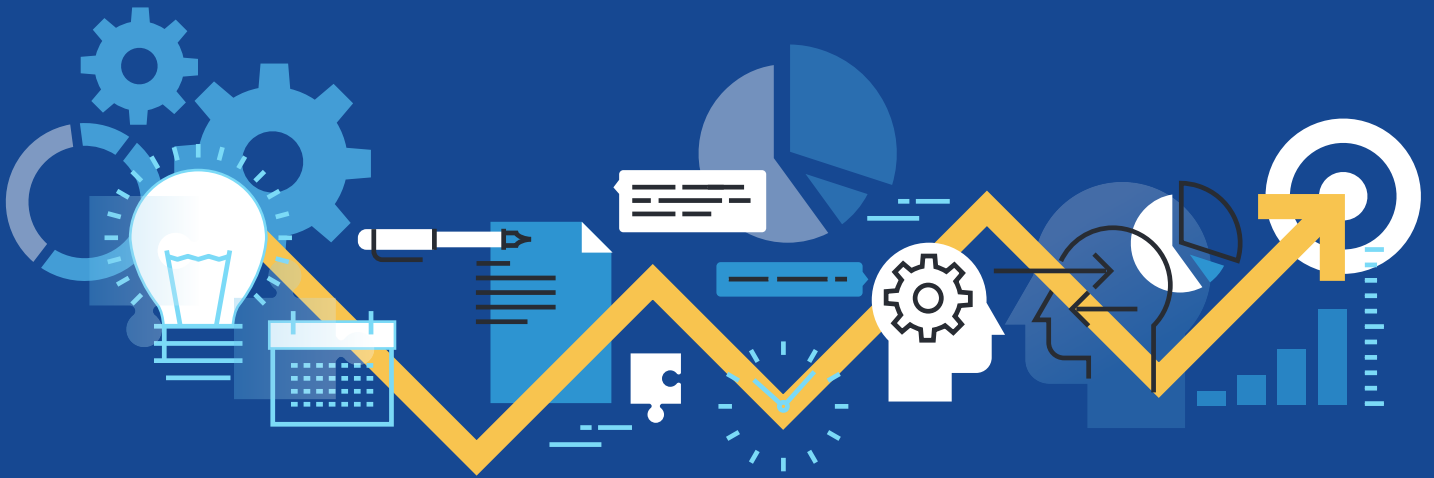
M: 857-218-9585

www.AgileSparks.com





6 WAYS TO TELL IF SAFETM IS RIGHT FOR YOU



6 WAYS TO TELL IF SAFe™ IS RIGHT FOR YOU

These days, it seems that scaling Agile is all the rage. And while Agile truly is the best, most efficient and most profitable way to get a great product to market quickly, **scaling Agile is a road filled with potholes and landmines.**

And if you don't do it properly, your product - and your team - can easily end up in a ditch.

So how do you know if SAFe is right for you?

Well... you could hire a team of consultants to go through what would certainly lengthy (and expensive) process to figure it out.

Or you could just **use this simple checklist and find out in a few minutes.**

Just check the boxes that apply to you and add up your score. At the end of this list, we'll tell you what your score means - and whether SAFe truly is for you.

So let's get started..

6 WAYS TO TELL IF SAFE™ IS RIGHT FOR YOU

1. Do you really NEED to be Agile?

*Agile is a great framework, but it's not for everyone. So do you even need Agile?
Let's find out:*

- ☐ Is there a strong business demand to improve Time to Market?
- ☐ Is there a strong business demand to improve Productivity?
- ☐ Are there serious IT/product development quality problems/complaints that you are struggling with?
- ☐ Is there a serious employee engagement issue that you are looking to address?
- ☐ Do you believe Agile is going to be a key ingredient in maintaining/extending your competitive edge or closing up on your competition?

2. Do you need to SCALE your Agility?

*Has your team gotten too big for your current Agile implementation?
(Do you need to achieve the Agility improvements discussed above in one of these contexts.)*

These questions will give you the answer:

- ☐ A program/product/system where 1-2 teams are not sufficient to deliver at the pace required by the business.
- ☐ A program/product/system where 1-2 teams are not sufficient to maintain the variety of systems/modules/knowledge needed to deliver value in this area.
- ☐ Dependencies between teams are repeatedly coming up as a real struggle - causing waste, delays, and frustrations all around.
- ☐ It is pragmatically impossible at the moment to solve the majority of the dependencies through reorganizing from component teams to feature teams. Coordinating / Collaborating between different teams seems to be a necessary evil for the foreseeable future.

6 WAYS TO TELL IF SAFe™ IS RIGHT FOR YOU

3. Is your organization looking for detailed guidance on how to scale Agile?

Feeling a little lost in trying to scale Agile? See if a helping hand is right for you:

- ☐ You prefer concrete prescriptive guidance for how to scale Agile beyond the team level rather than figure it out on your own from books/case studies/creating a custom framework with a consultant.
- ☐ You are looking for an approach that has a solid documented track record. This will make you feel safer going into this high impact initiative and will help you convince others.
- ☐ You are looking for the “industry standard” which will help you with bringing in experienced practitioners/coaches/trainers, communicating with the wider Agile community, etc.
- ☐ You prefer “Buy” over “Make”. For you “Not Invented Here” is an advantage not a turnoff.
- ☐ You prefer “Tailor” over “Make”. You are not a big fan of working “by the book” but are looking to leverage a good solid coherent set of best practices that you can work with.

4. Is SAFe Agile Enough for you?

SAFe isn't a one-size-fits-all Agile solution. Check the boxes below to see if SAFe is enough for you.

- ☐ Coming from a “Waterfallish” development lifecycle having a planning cycle every 10-12 weeks (every PI) provides sufficient business Agility.
- ☐ Replanning at the Feature/Program level every couple of weeks is not required or deemed to be a dysfunction that will be resolved by moving to a steady reliable cadence of 10-12 weeks.
- ☐ Delivery cadence will be somewhere around 10-12w. (NOTE: Yes. SAFe does say Deliver on Demand. But typically organizations that deliver every sprint or more often have lots more feedback to use for replanning which will drive them more towards a more frequent planning cycle - leading either to shorter PIs or an alternative more pull-oriented planning approach).

6 WAYS TO TELL IF SAFe™ IS RIGHT FOR YOU

5. CAN you implement SAFe? ?

Implementing SAFe is a pretty big commitment. So be honest about whether your organization is ready for it.:

- ☐ Are you serious about transforming your organization? SAFe might be SAFE but it's certainly not EASY. It will be hard work.
- ☐ Do leaders have the time/motivation to lead this change?
- ☐ Do you have authority to drive change in the organization?
- ☐ Do you and whoever is relying on the delivery of this development value stream have the patience for a period of slowing down while investing in building the capability to win later on? (Think the sports team that says we are focusing on building the team this year, not aiming for the playoffs/finals, etc.).

6. Are you willing to go for “Big Room Program Increment Planning”?

Can you make it work?

*An effective SAFe implementation requires your entire team to participate.
Are they up for the challenge?*

- ☐ Is “Whole program of 50-125 people doing program increment planning together in one big room” sound like something you can undertake every quarter or so?
- ☐ Is this something you are open to exploring the value of?
- ☐ Is this something you think you can pull off from a logistics perspective?
- ☐ If the set of teams that needs to work together is distributed across several locations/timezones - do you have, or can arrange, effective tele-presence solutions to enable these teams to have a “virtual big room” multi-day planning event every quarter or so?

6 WAYS TO TELL IF SAFe™ IS RIGHT FOR YOU

So - is SAFe the right approach for you?

See, we told you that was pretty quick...

Now it's time to find out what your score means. If you checked:

16-25

SAFe is a very good fit for your organization and, with proper implementation, you will definitely see considerable impact on business agility, productivity, quality and overall team happiness/engagement.

For you, It is probably time to get some SAFe training, run an implementation strategy workshop and launch SAFe in an Agile Release Train or even a full Value Stream.

11-15

SAFe still may be a great fit for you. What we typically do with organizations at this level is take a deeper dive and examine your organization a little more closely. This typically means an assessment together with a deeper implementation strategy workshop.

0-10

SAFe may not be the right choice for you, but other Agile frameworks can give you the massive results you're looking for. If the main reason for low scores is centered in the NEED area you're probably either doing well or have some blind spot that you're unaware of that is worth probing for. What we do in these situations is a more generalized Lean/Agile Management Workshop where we figure out the real NEED, understand the Lean/Agile options and help organizations figure out what options are worth pursuing for them.

**TO LEARN MORE ABOUT HOW WE CAN HELP YOU IMPLEMENT SAFe™
EFFECTIVELY, EFFICIENTLY AND PROFITABLY, CONTACT US
FOR A FREE STRATEGY SESSION:**

**safe@agilesparks.com
(857) 250-0479**



www.agilesparks.com

By *Yuval Yeret* Posted *January 25, 2018* In *Scaling Agile*

TL;DR – Looking for SAFe exam questions? Sorry, move along, nothing for you here... If otoh you're looking for some tips on how to properly study for the exam that seem to have worked for dozens of my students for the SPC exam and other SAFe exams, hang around....

I've recently been teaching quite a bit of [Implementing SAFe](#) classes. Students are always interested in some tips and tricks on how to prepare for the SPC certification exam, especially since it's a non-trivial exam even if you attend a class with trainers that know what they're doing and if you listen and participate throughout. The vast majority of my students pass the exam, but it doesn't hurt to know how to study.

Here are some ideas that seem to be working for my students, beyond the guidance already provided as part of the class by SAI (Which is good guidance in my experience)

- Spend time perusing the Big Picture for the different configurations at <http://www.scaledagileframework.com>
- Review the whole workbook again. For topics you're not sure about – find them in the big picture and read the relevant article
- For the Implementing chapters in the workbook, your homepage is <http://www.scaledagileframework.com/implementation-roadmap/>. It has articles that more or less align with the different lessons/stages.
- review the first and last slide of each lesson (learning objectives) and make sure you know to explain them.
- Review the glossary.
- Print the full big picture and implementation roadmap (on large paper if possible) and write down notes as you review. Just the act of writing helps you prepare.
- Try to draw the big picture and the implementation roadmap on your own from memory and see how much you can get right. Repeat until you like what you see 😊
- There are Kahoot quizzes created by my friend Inbar Oren (A SAFe Methodologist at SAI) a while ago that are publicly available. Some SPCTs (Including myself) run them during the Implementing SAFe class to help refresh topics from earlier lessons and drive some discussions about nuances/intricacies. You can run them on your own or get together with some others preparing for the exam and have more fun together 😊

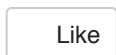
- 2 – lean/agile mindset <https://play.kahoot.it/#/k/60377928-5d32-4391-9237-8cda17cb2f13>
- 3 – safe principles <https://play.kahoot.it/#/k/84ffc33f-9e00-4348-971d-d87dafeab9e4>
- 4 – experiencing PI planning A (Building ARTs) <https://play.kahoot.it/#/k/0288bf37-e609-47d1-ba59-a6d4f7b1c9a7>
- 5 -experiencing PI Planning B <https://play.kahoot.it/#/k/3cd78066-5815-41b1-97aa-a0a7afb8c2c7>
- 5 – program execution <https://play.kahoot.it/#/k/d8b11dc3-720e-491b-a10a-c8e1b1d06024>
- 7 – solutions (old 8 value stream level actually) <https://play.kahoot.it/#/k/fa000fd6-cb7e-44da-98d6-2c5dc3d8ef2c>
- 8 – portfolio (old 7 portfolio) <https://play.kahoot.it/#/k/8caab32d-5d09-4638-96fb-20c99e16611f>
- I created another Kahoot that covers some of the Implementing SAFe lessons as well as SAFe 4.5 stuff (The original Kahoots cover 4.0 – they’re still mostly applicable)
<https://play.kahoot.it/#/k/df65a590-d545-4cb9-bf9f-09d16f3d0906>

If this is useful or if you have other tips you want to share, let me know in the comments.

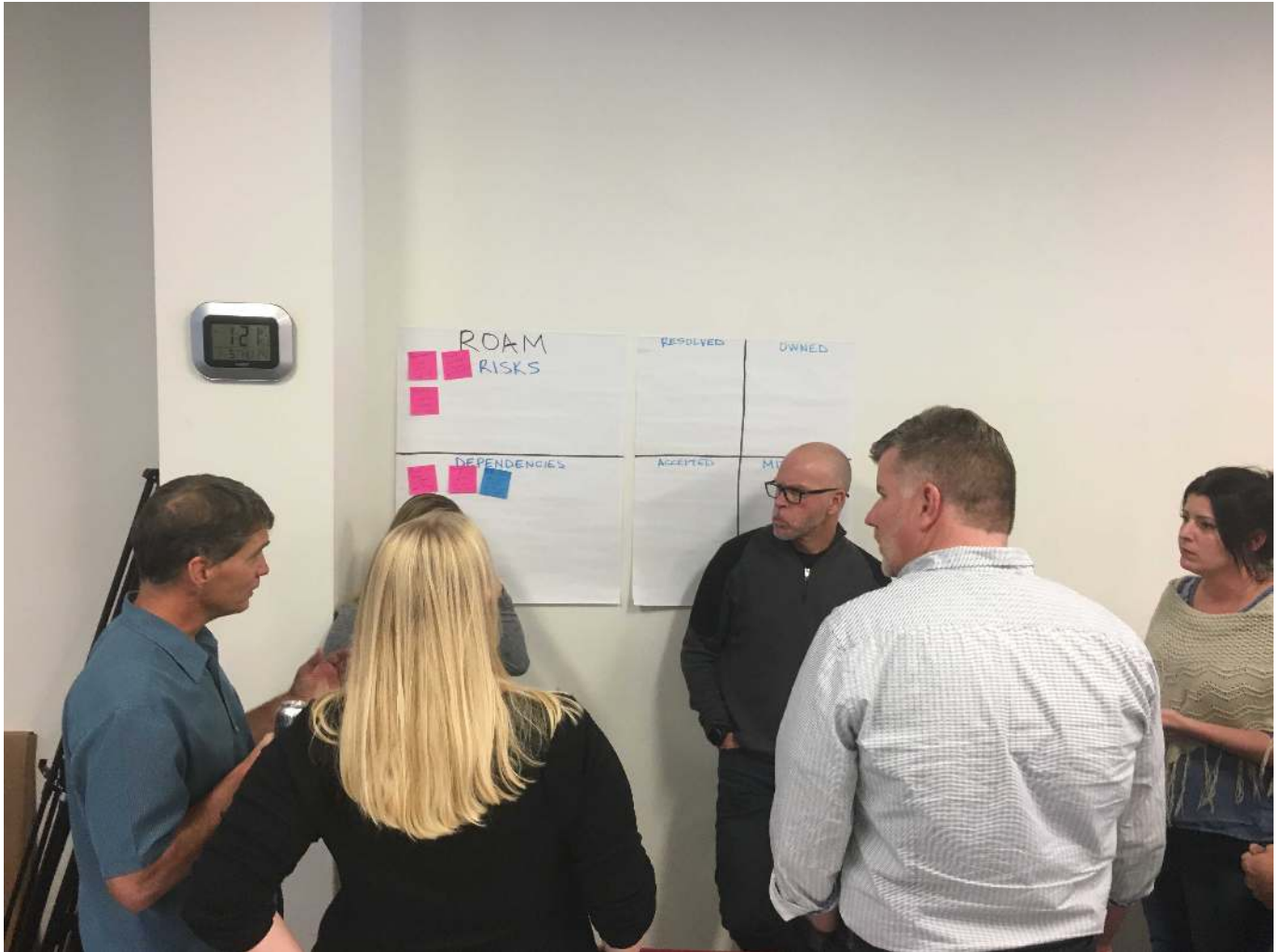
Share this:



Like this:



Be the first to like this.



By [Yuval Yeret](#) Posted *January 11, 2017* In [Agile Leadership](#), [Scaling Agile](#)

Last week I helped facilitate Program Increment (PI) Planning for an Agile Release Train (ART) practicing the Scaled Agile Framework (SAFe). One impediment for this ART was that although the leadership team ROAMed risks in PI Planning as well as continued to manage the flow of risks/issues using a ROAMing Kanban Board throughout PI execution, there wasn't enough clarity and alignment around what exactly would Owning a risk look like and what are the expected deliverables/objectives.



One experiment we tried this time around was to ask the ART Leadership Team to plan their quarter/PI very similarly to the other teams on the ART. Their source of “Features” was the list of risks/issues identified beforehand (during the last PI as well as during the Inspect and Adapt / Retrospective workshop) as well as those emerging throughout the PI Planning event. They took their features and broke them into specific

deliverables and then came up with PI objectives related to those areas. They presented them to their stakeholders – the entire ART – heard their feedback and adjusted course.

Because the ART leadership team also has a role of walking the room, visiting teams on their breakouts, asking questions, sensing what is going on, being available to answer questions, we decided to split their time on each breakout between doing their own planning and going around the room.

This worked pretty well. We feel we are better set to enable the ART to run better based on the work done by the leadership team. And teams feel better that their leaders are there for them. This was actually very visible when we took a [pulse check about the level of “Lean/Agile Leadership”](#) towards the end of the PI planning. We got very high scores in this area.

Now it’s the time for execution. The leadership team is planning to run iterations like other teams and showcase their progress on risks/issues in system demos. We shall see how well that works...

Share this:



Like this:

Loading...



Invitation based SAFe /

Applying Lean-Agile Principles to SAFe's Implementation Approach

by Yuval Yeret, CTO, AgileSparks

[Introduction](#)

[The Vision - Implementing SAFe using Invitations](#)

[Invitation-based SAFe Overview/Roadmap](#)

[Using the SAFe Implementation Workshop to invite the Enterprise to consider SAFe](#)

[Spreading SAFe through an invitation to Leaders](#)

[Using the SAFe Implementation Workshop to launch a Program/Value Stream](#)

[Invitation Based ART Launch](#)

[Summary - Evolving SAFe's Implementation Approach with SAFe's Lean-Agile Principles](#)

[About the Author](#)

Introduction

Implementing any kind of organizational change, such as adopting SAFe, is hard and raises several key concerns:

- How do we convince people to adopt the new ways of working?
- How do we get the organization moving in the new direction?
- How do we make decisions about how to implement SAFe in the enterprise?

SAFe recommends decentralizing control, while providing vision and gaining alignment. It is also about respecting people and culture and maintaining effective flow. In this guidance article we will discuss ways we can “Walk that talk” in the way we run a SAFe implementation.

The default approach for implementing organizational change is the "mandate" or "push" approach. This may appear to be the fast and easy way, where a central group of change agents decide when people will "board" the Agile Release Train (ART), as well as how the train should operate.

It may seem easy because the change is mandated and there is little or no discussion about whether the change should occur. It also appears to reduce the risk of a shallow SAFe adoption that doesn't even cover the [essentials](#), due to bad implementation decisions, by people who have limited or no experience. The problem with this classic approach is mainly that people don't like to be changed. They like to be involved in making the decision to change as well as designing the change.

The Vision - Implementing SAFe using Invitations

Martin Fowler, one of the Agile Manifesto signatories, wrote an article in 2006 called “[The Agile Imposition](#).” In the article, Fowler says, “Imposing an agile process from the outside, strips the team of the self-determination, which is at the heart of agile thinking.” But what if we can't wait for people to self-determine that they should go agile? Should the organization wait? Even if it kills the business?

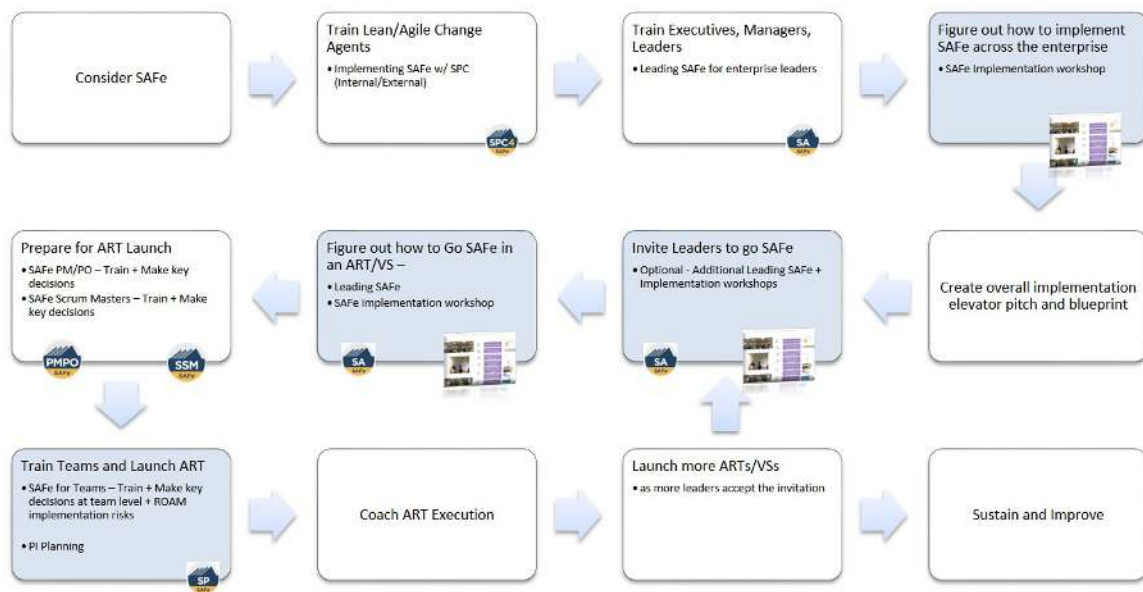
SAFe's 9th principle - "[Decentralize Decision Making](#)" provides some guidance here. The decision whether to go agile and what approach to take is *infrequent, long lasting, and provides a significant economy of scale*. So it is a classic strategic decision to centralize.

But once that central decision to go SAFe has been made, The Agile Manifesto says, "*Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*" And "*The best architectures, requirements, and designs emerge from self-organizing teams.*" If we apply these two principles to SAFe implementation this would mean the best plans for implementing SAFe will emerge from self-organizing teams (or teams of teams) of the people adopting SAFe. Implementing SAFe using a leaner and more agile approach will send a message about the strength of management's commitment to the [Lean-Agile mindset](#) described in SAFe. Can you think of a better way to signal "respect for people and culture"?

In PI planning Business Owners and Product Management present the business context, the vision. The planning context and structure of PI Planning is a "container" in which the Agile Release Train self-organizes to figure out how and how much they can do to further the vision.

Similarly, Invitation-based SAFe implementation needs to set the context and provide the right structure for the group to figure out how and how much they of SAFe they can achieve in their first implementation PI.

Invitation-based SAFe Overview/Roadmap



Using the SAFe Implementation Workshop to invite the Enterprise to consider SAFe

*“Leadership is charged with making these types of [strategic] decisions, **supported by the input of those impacted by the decisions.**”*
(<http://www.scaledagileframework.com/decentralize-decision-making/>)

One approach that my company, [AgileSparks](#), has discovered works well when teaching Leading SAFe is to accompany pure training-mode with a “[SAFe Implementation Workshop](#).” In this workshop, a group of leaders discuss:

- The reasons for considering SAFe
- How good of a fit SAFe seems to be
- Identifying value streams and designing ARTs
- Guidelines for determining how roles in SAFe are chosen or assigned to people
- Implementation risks and organizational impediments

8:00-9:00	Why SAFe?	Review the needs and goals for enterprise agility through exploration of pains/opportunities in the current way of doing things. Establish a sense of urgency
9:00-10:30	Identify value streams and design ARTs	Identify one or more development value streams and create initial design of Agile Release Trains
10:30-11:30	How do our people map to the SAFe roles?	Establish key SAFe roles (e.g. RTE, Product Managers, Business Owners, PPM, Product Owners, Scrum Masters, Agile teams)
11:30-1:00	Define the initial scope	Define initial features, capabilities, or epics to begin implementing the strategic themes of the portfolio. Use MVP, Lean-Startup thinking
1:00-4:00	How will SAFe work here?	Facilitate an open space, lean coffee, or world café to establish what it means to apply SAFe in your context
	ROAM Risks + Confidence Vote	Identify and ROAM risks and issues. Hold confidence vote on your defined SAFe implementation
4:00-5:00	Identify Next Steps	Determine next steps (e.g. set the date for the first ART launch, form guiding coalition, explore problem-solving activities, etc.)

Figure 2. Implementation workshop sample agenda

Through this interactive collaborative process members of this group will become the initial “guiding coalition,” a term popularized by Kotter. They consider options, make decisions, commit to work together towards the shared vision.

As you can see in the “SAFe Invitations Implementation Approach Roadmap” above, this workshop format is useful when considering SAFe with a group of leaders across an enterprise/division as well as later on when preparing to launch a specific Value Stream or Agile Release Train. Another way to look at it is as a different variant of how to run the SAFe Value Stream workshop that is frequently used following up an Implementing SAFe/Leading SAFe class to help identify Value Streams/Agile Release Trains for actual implementation of SAFe in the organization.

Spreading SAFe through an invitation to Leaders

The outcome of the enterprise-level implementation workshop is an invitation to potential value streams and/or ARTs to consider what SAFe would mean in their context and figure out when/how to start their SAFe journey.

In most cases the potential ART/VS leaders (think VP level leaders) participated in the initial Leading SAFe + Implementation workshop and are now ready to consider bringing SAFe to their group.

In larger enterprises there might be a need for further Leading SAFe+Implementation Workshops to expose more potential ART/VS leaders. In the “Invitations” spirit you can invite leaders to participate in such a class or bring SAFe into their organization, not force/mandate them. The first leaders to accept the invitation are ideal “prospects” (innovators or early adopters) for starting the SAFe journey and should be where SPCs should initially spend most of their time.

Using the SAFe Implementation Workshop to launch a Program/Value Stream

Once a leader decides that the timing is right to consider SAFe in their area, they should again repeat the same pattern - Leading SAFe combined with a SAFe Implementation Workshop to figure out how to go SAFe. This time the audience is Lean/Agile leaders for the ART/VS as well as ART/VS roles such as RTE, Product Management, System Architect.

Typically the Product Owners and Scrum Masters don't participate in this workshop - in many cases it is in this workshop where they leadership team figures out what is the mapping between the PO/SM/PM roles and the roles and people in the group.

As the POs, SMs, PMs, are identified they get trained in PM/PO and SAFe Scrum Master workshops. When using the “SAFe Invitations” implementation approach these workshops should include vignettes from the implementation workshop such as starting with a pains/why session and gauging confidence level and ROAMing implementation risks towards the end of the training.

This will help SMs/POs/PMs connect to the vision and feeling more involved in designing the implementation approach. This will rally them to join the “guiding coalition” of the group.

Invitation Based ART Launch

Once leaders of a certain area are on board and have identified an Agile Release Train/Value Stream to focus on and the ART stakeholders/roles have been trained and brought on board as well, it is time get the team-of-teams rolling.

The combination of training everybody using “SAFe for Teams” at the same time with planning the initial PI and getting a real feel for how SAFe will look like works better than just sticking to theory, training exercises, and games.

Bringing an invitation approach into the ART Launch means decentralizing some decisions around how to operate SAFe to the people on the ART themselves. Aspects like program board structure, Definition of Done (DoD) policies, ready policies, engineering practices, agile testing strategies and some other aspects are great candidates for having breakout and integrate discussions as part of the SAFe for Teams training. Additionally, you may want to allow teams make other local decisions about how they use SAFe, as long as they're aligned with the SAFe principles and it does not cause problems for the other teams on the train, or the ART as a whole.

Another interesting practice that invites people on the ART to participate in figuring out implementation details is team self-selection. In this practice the ART leaders provide guidelines/constraints and let the people on the ART figure out what at the actual teams should look like. [8] [9] [10]

Caution: Be careful when allowing customization at this point. There's a two-fold risk, either removing too much from the SAFe model, as well as adding too much overhead with additional process. There's tremendous value to trying out the SAFe framework, more or less “as is,” or with careful configuration/customization with the help of a seasoned SAFe program consultant. I have experienced it first-hand, and only then continuing to remove/add/change practices. For a view on the essentials that shouldn't be changed, please see the [SAFe Essentials](#) guidance article.

As a blueprint for how the ART will work starts to emerge, it's time to gauge people's level of confidence for how the implementation of SAFe will work and surface risks. You can use a "fist of five" confidence vote, similar to what we do in PI planning to gather this feedback, as well as proactively inviting people to share their concerns.



Figure 5. Confidence Vote

Follow up with questions like "Based upon what we just learned so far, are there any significant concerns that would prevent us from starting to use SAFe?" The responses from the teams can be used to seed topics for a brief problem-solving workshop or "open space" session, where people can raise their concerns, and then join or lead a breakout session to identify solutions.

Another approach is to ROAM each risk/issue like we do in [PI Planning](#). The use of the facilitation techniques, like ROAMing risks, confidence vote and open space all demonstrate a "[servant leadership](#)" style. As leaders, we are not just telling people what to do, we are involving them in figuring out the "how" and serving them by owning resolution of key systemic risks to the change. This same technique can be applied during PI Planning confidence vote and "ROAMing" of the risks.

Summary - Evolving SAFe's Implementation Approach with SAFe's Lean-Agile Principles

In essence, the approach described above uses Lean-Agile practices and principles to drive the adoption of SAFe. Using SAFe to adopt SAFe. We're asking leaders to set the vision/direction for implementing SAFe and inviting their people to come onboard and participate in designing this change that will have so much positive impact on how work will be performed in the future.

Decentralizing control and engaging as many people as possible in figuring out how to use SAFe tends to improve the quality of SAFe implementation because of “Wisdom of the crowds” and the higher motivation people have when they’re invited to be involved. This applies to leaders at various levels using the [SAFe Implementation Workshop](#) that complements Leading SAFe training and to teams and ART stakeholders using vignettes like pains/vision mapping, implementation confidence votes and risk ROAMing in each and every SAFe training used to prepare and launch SAFe in ARTs/VSSs.

About the Author

Yuval Yeret is the CTO and an active enterprise-level Lean-Agile consultant at AgileSparks. He is also the head of AgileSparks, USA located in Boston, MA.

Yuval is an SPC4 & SPCT candidate, as well as experienced in "Invitation approaches". He is a frequent [blogger](#) and conference speaker about all things enterprise Lean and Agile.

[AgileSparks](#) is a Scaled Agile Inc. Gold Partner.

Learn More

<http://www.agilesparks.com/safe-implementation-strategy-leadership-focusing-workshop/>

[1] <https://www.linkedin.com/pulse/openspace-agility-right-you-daniel-sloan>

[2] John Kotter. Leading Change, Harvard Business Press, Dec 30, 2013

[3] <http://openspaceagility.com/big-picture/>

[4] <http://yuvalyeret.com>

[5] <http://www.infoq.com/news/2014/10/kickstart-agile-kanban>

[6] <http://www.infoq.com/interviews/lkfr14-yeret-kanban-agile>

[7] <https://management30.com/practice/delegation-board/>

[8]

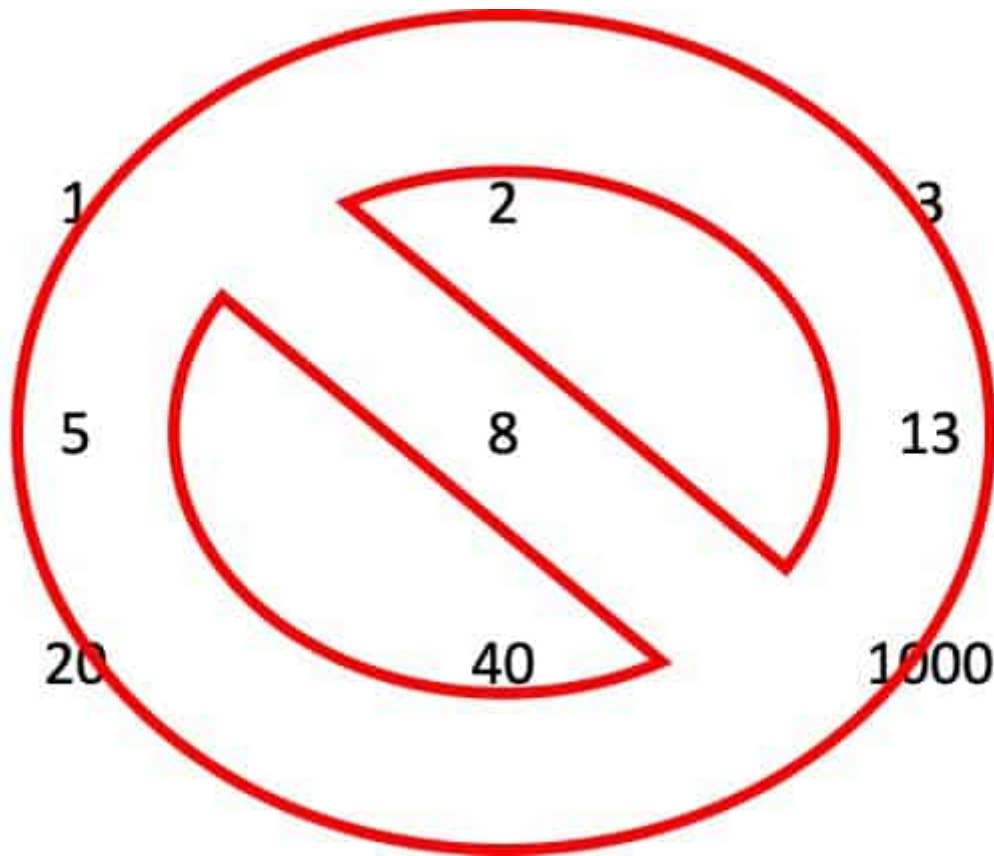
<https://www.linkedin.com/pulse/large-scale-self-selection-australia-post-interview-andy-sandy-mamoli>

[9]

<https://www.amazon.com/Creating-Great-Teams-Self-Selection-People-ebook/dp/B019EKWG6M/>

[10]

<https://www.scrumalliance.org/community/articles/2013/2013-april/how-to-form-teams-in-large-scale-scrum-a-story-of>



By [Yuval Yeret](#) Posted [October 25, 2018](#) In [Scaled Agile Framework](#)

SAFe™ (The Scaled Agile Framework) uses Story Points throughout the various levels as its estimation currency. This is covered in the “[Story](#)” article on the [SAFe site](#). This is a pretty standard practice in organizations scaling agile these days. If you dive a bit deeper into how this is done in SAFe you will see that actually the story points used in SAFe are quite similar to “Ideal Developer Day” as this helps the teams align to a common baseline and support a rational economic ROI discussion at the level of Features/Capabilities that require effort from more than one team or haven’t even been mapped to a specific team yet.

An alternative to using Story Points at the team level that is interesting to look at especially as Kanban is becoming a first-class citizen of the SAFe world is to use [NoEstimates](#).

In essence this means not trying to estimate the size of the work just slice work into a size that we can bite on and turn around quickly. To get a quick grasp of it think of replacing your classic fibonacci planning poker cards with a set of cards saying 1, TFB (Too Frighteningly Big), NFC (No Faintest Clue). This approach is rising in popularity among Kanban/ScrumBan practitioners – We’ve been looking at the “Iteration Planning” process for years trying to address some of its wastes/tensions.

This “No Estimates” inspired approach to story estimation in PI Planning has a couple of benefits:

- It is faster – which comes in handy during **PI Planning**.
- It forces teams to slice into smaller stories which is better.
- It is naturally easier to baseline/align multiple teams around the same definition. (BTW a 1 here would be something like a 3 or 5 in the classic SAFe velocity calculation – a story that takes the team about 3 days to develop and 3 days to test)



Iteration Planning becomes even easier – Just understand your velocity and how many stories you can fit. Minimal time is spent estimating.

There are a couple of challenges though. Forcing the team to split their whole Feature into small 1-size stories including those that are only going to be pulled in later iterations in the PI might be a waste of time.

Another challenge is how to make rational economic decisions at the Feature/Capability and even Epic level without estimates. #NoEstimates die-hards say it doesn't make sense to estimate even at this level, not just the stories level. I'm not convinced. What I typically do in cases where teams stop estimating story sizes is just use story counts as the currency at the higher levels. So instead of saying "This seems to be a 20 Story Points Feature" we would say "This Feature seems to be something around 20 stories" meaning we ESTIMATE it will map to about 20 stories when we eventually slice it. We DON'T slice it to stories in order to estimate. We reach that estimation using a classic relative estimate approach like Planning Poker / [Team Estimation Game](#). This is actually something that helps with the first challenge as well. Knowing that we're dealing with a Feature that we think has about 20 stories and that we identified 7 stories for the first iteration and 4 stories for the second, we might say something like there are around 9 stories more for the third iteration or even better – let's look at the remaining chunk and compare that to our feature estimation scale and see how many stories we think there are there.

Of course, if we think there are dragons (a.k.a dependencies) hidden in this remaining chunk we should make the effort to slice it into smaller stories and work out the dependencies/scheduling with the other teams on the train.

I've done this with clients in the trenches. Based on this experience, I consider this alternative approach to estimation a legitimate alternative to Story Points estimation at the SAFe Team Level.

This blog post was originally posted on [Yuval's personal blog](#) back in 2016.

Share this:



Like this:

Like

Be the first to like this.



By [Yuval Yeret](#) Posted [November 12, 2018](#) In [Scaled Agile](#), [Scaled Agile Framework](#), [Scaling Agile](#), [Scrum](#)

SAFe includes Scrum – so how come many Scrum practitioners and thought leaders consider it unsafe?

The Scaled Agile Framework (SAFe™) is one of the most popular approaches to applying agile at scale out there. SAFe's perspective is that "Nothing beats an Agile Team" and it doesn't try to reinvent the wheel or even innovate too much when it comes to the Team level. It takes advantage of established frameworks and techniques that work well – Scrum being the first and foremost of those.

Where it starts to get interesting (unsurprisingly) is when the patterns and practices for scaling are introduced – in SAFe's Program Level. SAFe's premise is that in the real world one team typically isn't enough and several teams need to work in concert to build larger systems/solutions/products. In SAFe's Program Level, a key piece is the Agile Release Train which is considered a team of Agile teams.

When it comes to the Program level SAFe doesn't try to reinvent either – but here it uses Scrum/Kanban as a starting point and innovates in order to deal with some specific challenges of larger programs. SAFe also deals with Larger Solutions and the Portfolio, but let's leave those out of the scope of today's discussion.

The above intent, together with the fact that SAFe uses the Scrum Guide as its reference to what Scrum is, are encouraging signs. So, again, why do so many Scrum practitioners, trainers, and thought leaders consider it unsafe and a Scrum-but? I hear a lot of questions and claims. Let's try to recreate some of these discussions and along the way make some recommendations?



Looks like SAFe's Scrum Master is a coordinator and focal point for the team – not just a servant leader and coach accountable to enacting Scrum.

Yep. SAFe's Scrum Master is more of an Agile Team Lead. This is much easier to implement in the real world but also means it will be much harder for the team to self-organize because they have a team lead that isn't just focused on helping them improve via Scrum but is also their focal point for Scrum of Scrums, during PI Planning, etc.

The way I look at it – this is indeed a compromise that SAFe practitioners should be aware of. And part of the journey of implementing SAFe should be to maybe start with this Scrum Master stance but evolve towards more of the classic, professional Scrum Master stance over time. To use the leadership styles model we discuss in the Leading SAFe class – the starting point is more of an orchestrating and technical expert kind of leadership stance and the goal should be to evolve towards a more serving the team and the process style over time.

The main concern I have is not that SAFe's Scrum Master is different than how Scrum defines it. It's that this difference isn't made transparent – which doesn't give practitioners the opportunity to inspect, think about it, and maybe adapt. Maybe your organization is actually better off with an Agile Team Lead than a Scrum Master. But you won't have a chance to think about it and decide if you think you already have Scrum Masters.

SAFe's Product Owner is a proxy, not a real Product Owner. They are more similar to a team member focusing on the stories than a person accountable to optimizing value delivered by a Product.

SAFe's approach to product ownership is that scale is achieved by splitting the product ownership role between Product Management, which is more like the classic Scrum Product Owner, and the Product Owner, which is indeed more like a proxy or technical product owner working more closely with teams. One of the main reasons SAFe takes this path is that it's hard for one Product Owner to deal with too many teams while balancing both outbound and inbound activities.

Large Scale Scrum and Nexus prefer to have one Product Owner for the entire product with one Product Backlog. In real life, these Product Owners are typically accountable to the value delivered by these multiple teams and rely upon a lot of assistance from the Development Teams in order to deal with the challenge of scale.

If we ignore the differences in lingo, This is quite similar to SAFe's approach. But we can't ignore the differences in lingo. We DO want to see Product Owners as individuals owning products and being accountable to optimizing value.

What I've seen in the trenches ranges from SAFe Product Owners that really own a product within the bigger solution, own a set of features or even a specific feature that is currently being developed, all the way to technical product owners that aren't real product owners. There's definitely room for more discussion of this continuum and the impact of it in the SAFe PO/PM body of knowledge (Similarly to the discussion of the Feature/Component team continuum).

Scrum is a simple framework that deals with complex domains. SAFe seems more of a methodology to Scrum practitioners as it has many more details and seems to try to solve all challenges in a prescribed way. SAFe's creators seem to enjoy the fact that it is complicated since it provides an excuse for more and more training possibilities.

Well, the reality is that SAFe serves the mainstream market of practitioners that struggle to get from Scrum's simple framework to an approach that works in their reality. Scrum simply doesn't provide enough answers to some of the tough scaling challenges, and not everybody has the time or skills to come up with an approach that works in their context. This market needs some more guidance.

Looking at SAFe as it grows over time I see a constant struggle between the desire and drive to simplify and focus on the essentials to the desire to give more answers. Like any other product, it is tough to figure out which features/aspects to build, get rid of, simplify, and how to optimize the experience. It's hard to create the ideal scaling framework.

Beyond how SAFe is defined, there's also how people perceive it. And yes lots of practitioners prefer to see it as a Methodology rather than a framework. As someone teaching SAFe Program Consultants, I try to discuss it in my classes. (See a [recent blog post](#) I wrote about this)

SAFe's Sprints are 3 months long and are planned in detail – How is that Agile?

Well, let's unpack this. SAFe has a cadence at the Team and Program levels. The team-level cadence is called Iterations but other than that different name is almost identical to the Scrum Sprint. It is 2 weeks long, the goal is to deliver a potentially releasable increment of working software, There's Iteration Planning, Daily Standup (which is essentially Daily Scrum), Iteration Review and Retrospective.

I'm guessing the confusion kicks in when people look at the Program Increment. That's typically 8-12 weeks long and includes multiple Iterations (4-6 typically). Why don't we just have a Program Increment that is at the same length of the team-level increment? Because from an economic perspective the transaction/coordination costs for running the whole program on a 2-week cycle don't make sense. Think about having big room planning with 100 practitioners every two weeks. Kind of hard.

Why do we even need this big room planning in the first place? Now that's another question. In situations where teams do have some dependencies, when we need a longer horizon business planning and we do want to involve everyone in having discussions about what is valuable, realistic, and converge on plans, big room planning comes to the rescue. Do we have to have these discussions every two weeks? probably not.

So the approach SAFe takes is to look at each one of the program-level activities and consider both the coordination cost/overhead as well as the holding cost or cost of delay and come up with the right frequency. For example, while Program Increment Planning happens only at the Program Increment cadence, System Demo, which is similar to the Sprint Review but at the program level, happens on the iteration cadence so every two weeks typically. Why? Because the cost of delayed empiric feedback is very high and we understand we live in an uncertain environment, we assume variability and we want to preserve the option to adjust course throughout the PI.

This is similar to the Scrum Planning Onion. Teams and Agile Release Trains plan at the Daily, Iteration, and Program Increment levels. The deeper in the onion we are the more detailed we plan, but the shorter our planning horizon. So in Program Increment (PI) Planning we should plan for a longer horizon but at a much lower level of detail. Do a lot of SAFe practitioners plan the PI in too much detail, not leaving enough room for uncertainty and learning once we get to the Iteration and Daily planning levels? Oh yes. Does a lot of Scrum practitioners do the same thing for the Sprint not leaving enough room for uncertainty and learning throughout the Sprint?

Like the Sprint Goal guides the Dev Team in case they want to consider changing the Sprint Backlog, PI objectives help teams adjust course throughout the PI if it helps them achieve their objectives.

Bottom line, SAFe's Program Increment and the way you plan it can be closer to "following a plan" or an agile basis for "responding to change". I certainly see it and teach it as the latter.

SAFe focuses on predictability much more than it does on empiricism and value discovery

SAFe actually tries to balance business agility with predictability. Both of those are important to the typical enterprise-scale technology organization.

SAFe includes mechanisms such as PI Planning, Roadmaps, forecasts, PI Objectives, confidence votes to provide predictability. It includes Stretch PI Objectives, The Innovation and Planning iteration and specific recommendations on how to plan in order to maximize predictability in face of variability and uncertainty.

It also includes System Demos, Continuous Integration, Minimum Viable Products, and others in order to deal better with uncertainty.

And there's no assumption that predictability will be absolute. A program/ART that achieves 80% predictability is considered within a reasonable range. And this predictability is measured in achieving outcomes, not delivering stories or points. This supports agility of adjusting what features we deliver and how as long as we focus on achieving the outcomes the business is focused on.

SAFe allows you to defer integration and hardening to the end of the Program Increment

Not really. SAFe used to have a Hardening iteration but it's been decommissioned for years now. The Innovation and Planning iteration is the place to take a breather, do some innovation activities that work better when you can clear your head and focus on them (Which doesn't mean by the way that innovation isn't allowed or needed throughout the PI), reflect on the current PI and plan for the next PI. integration and hardening is part of the definition of Done that each team strives for within every 2-week iteration. The System Demo that happens every 2 weeks is an opportunity to review the whole integrated system increment, get transparency for where you are, and inspect and adapt the plan for the rest of the PI accordingly.

What you could do about this as a SAFe practitioner/trainer

I wish more SAFe practitioners would dive deeper into the Scrum world as a step in their life-long learning journey. A self-respecting SPC should also have enough knowledge and experience with Scrum to pass at least some of the [Scrum.org assessments](https://www.scrum.org/assessments). The same applies to people training SPCs. They should have a very strong experience with Scrum and Kanban. An advice to those planning to register to an Implementing SAFe class and become SPCs – verify your SPCT knows his/her Scrum and Kanban. Check if they have a PSM1/2/3.

On an ongoing implementation, one useful thing you could do is run a workshop reflecting on the Scrum Guide and what are some key gaps to consider addressing. I've done this in one of my larger financial tech clients and it was a pivot point in the implementation. We looked specifically at the Scrum Master and Product Owner roles, identified a lot of gaps and changed our perspective about these roles.

What you could do about this as a Professional Scrum practitioner/trainer

As an SPC Trainer (SPCT) and a Scrum.org Professional Scrum Trainer (PST) I'm committed to helping people understand and implement SAFe safely. It isn't the only scaling approach I work with, but a lot of people seek me out when they do want to implement SAFe but want to keep to the true spirit of Agile/Scrum.

I'm glad to see more and more of my colleagues in the professional Scrum.org community that are interested in working towards better understanding of Scrum Theory, Values, Events, Roles and Artifacts in the SAFe world. After all, that's what it means to be a community that shows respect, openness, and courage.

Since SAFe is so prevalent, I think this is a huge opportunity to improve the profession of software delivery.

I'm excited! I see another [bridge](#) on the horizon...

[This article was originally posted on the Scrum.org blog](#)

Share this:



Like this:

Loading...

By [Yuval Yeret](#) Posted [October 15, 2018](#) In [Scaled Agile Framework](#)

Last week this was a theme in the [Implementing SAFe](#) class [Ofer Cohen](#) and I ran in Waltham, MA.

We find it crucial when training new SAFe Program Consultants (SPCs) to emphasize that they should use SAFe as a framework not a methodology.

First, what's the difference between a framework and a methodology? I found this concise useful comparison written by Liz Keogh who I think highly of [over at Quora](#) –

A methodology is a set of principles, tools, and practices which can be used to guide processes to achieve a particular goal.

A framework is a loose but incomplete structure which leaves room for other practices and tools to be included but provides much of the process required.

... Scrum could be considered a framework, as it leaves room for teams to choose their own technical processes, development roles, etc. XP might be considered a methodology, as it provides guidelines for all the same things that Scrum does, along with relevant technical practices. ...

With this in mind, what we emphasize in the workshop is the options and choices you have when you Implement SAFe. Yes, some people look at SAFe and see a methodology that tells you how to estimate, prioritize, plan, how your kanban boards should look like and what questions to ask in each Scrum of Scrums. We prefer to see all of those as a good set of options to start with in many contexts, but not necessarily best practices that always work.

For example, we don't believe story points estimation is necessarily the best way to estimate in all cases. We believe that sometimes it's enough to [just count stories](#).

The schedule/agenda for PI Planning is great, but we definitely inspect and adapt it on every implementation depending on the context and encourage SPCs and RTEs we teach to do it as well.

We always inspect and adapt the definition of Workflow of the Program and Portfolio Kanban boards on our implementations and we talk about it in class as well.

We always mention that SAFe's approach to Weighted-Shortest-Job-First Cost-of-Delay-based prioritization is only one option and that some other interesting and useful and maybe even better ones for your context are available (and we point people to [Don Reinertsen](#) and [Joshua Arnold](#))

What is the right Agile Release Train and Value Stream design? SAFe provides ways to help you design your implementation including some principles and considerations, but it doesn't give you a hard and fast answer... This is one of my favorite sessions in the Implementing SAFe class actually.

Which elements of the SAFe Big Picture do you need? Which Spanning Palette or Large Solution elements does it make sense to use even if you're using just Essential SAFe? And does it make sense to use Large Solution or Portfolio or Full? When?

In general, What is the right way to roll out at scale? Again, SAFe gives you some options and considerations to be aware of but doesn't give you a concrete playbook.

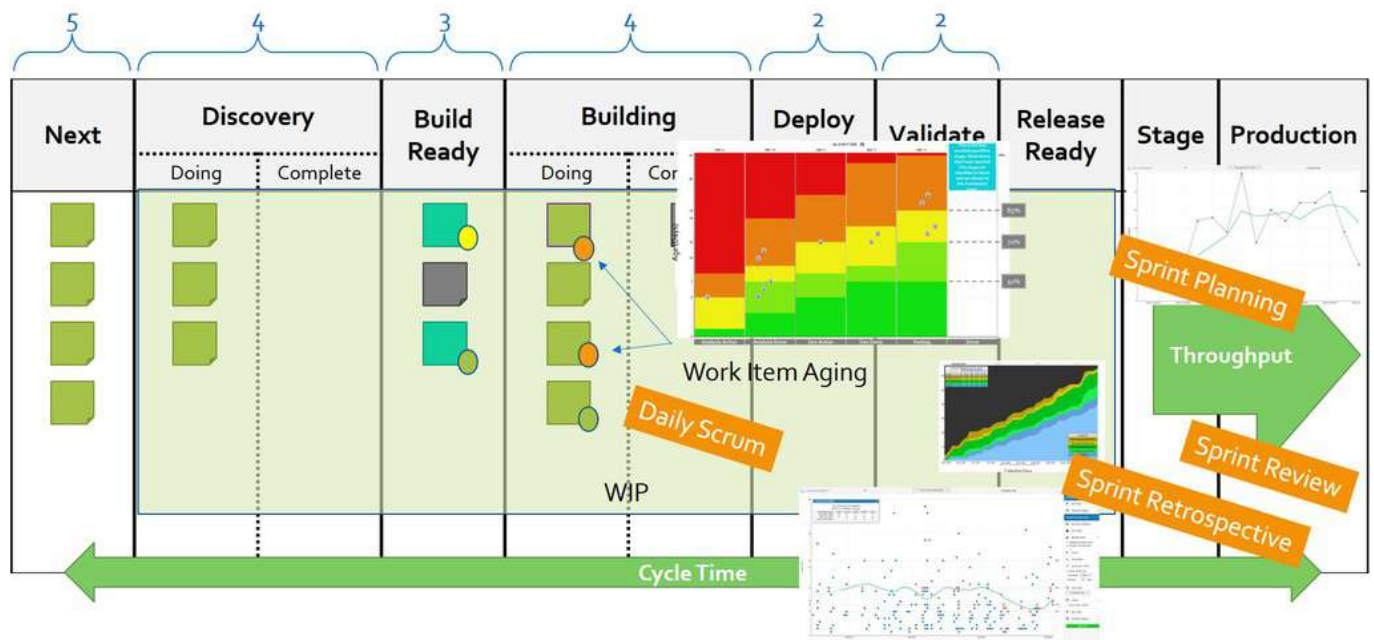
Bottom line, both when it comes to how to practice SAFe as well as how to implement it, we prefer to consider it a very useful but flexible/incomplete structure that requires well-trained and experienced practitioners to successfully apply, and that's a key design principle for our [Implementing SAFe workshops](#) where we train future SPCs.

Share this:



Like this:

Loading...



By [Yuval Yeret](#) Posted [October 26, 2018](#) In [Kanban](#), [Metrics](#), [Scaled Agile Framework](#), [Scrum with Kanban](#)

The Premise

A year ago Scrum.org, in collaboration with Daniel Vacanti and myself, published the [Kanban Guide For Scrum Teams](#), a guide that is aimed at helping Scrum Teams take advantage of Kanban/Flow principles and practices. (I wrote an earlier blog post about [understanding the guide](#))

SAFe™ has included Kanban at all levels since version 4.0. Some basic guidance about Kanban is included in most if not all SAFe curriculums. Can a SAFe practitioner learn anything from the Kanban Guide For Scrum Teams?

In this blog post I'll explore some of the flow metrics from the guide with an emphasis on those that aren't covered in SAFe.

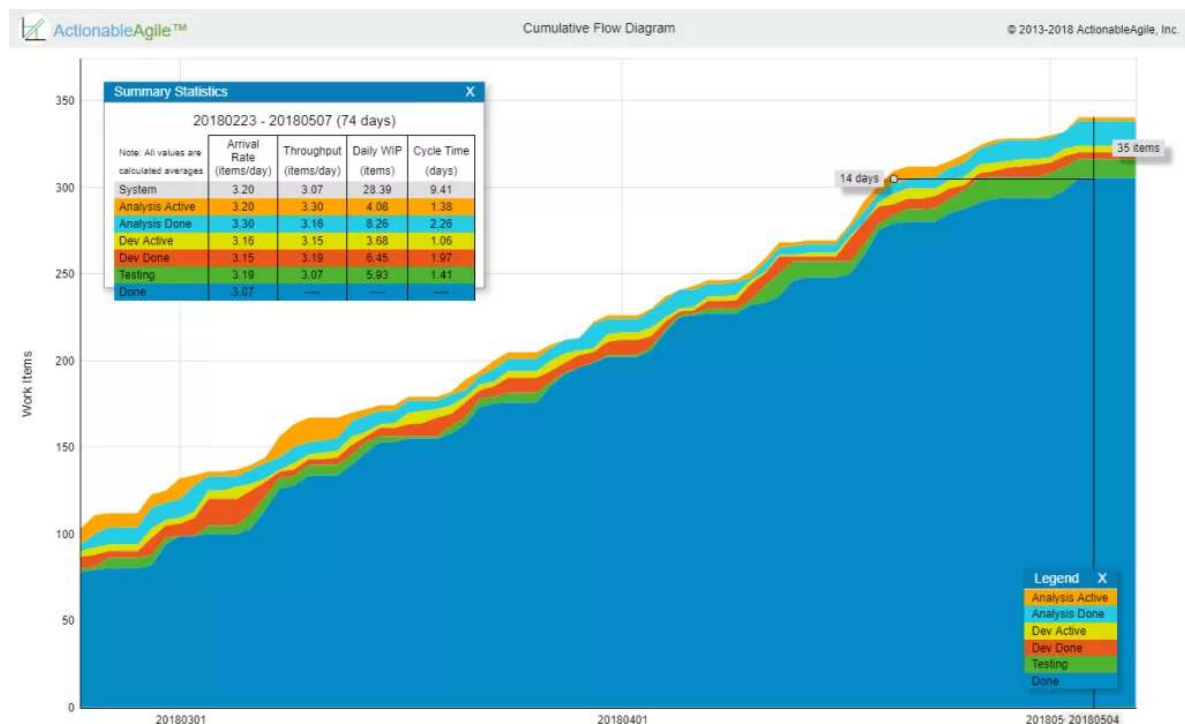
Scrum with Kanban Flow Metrics

Work in Progress (WIP)

The number of work items started but not finished

Note the difference between WIP and the WIP Limit. The WIP Limit is a policy which Agile Teams use as a “constraint” to help them shape the flow of work. The goal of the WIP Limit is to reduce the amount of actual work in process (WIP). The team can use the WIP metric to provide transparency into their progress towards reducing their WIP and improving their flow.

While teams can directly visualize the WIP levels over time (which I recommend), most people use the Cumulative Flow Diagram to visualize the WIP. Visualizing and reducing WIP levels is mentioned in SAFe but it never hurts to repeat it since it’s such an important flow metric to pay attention to.

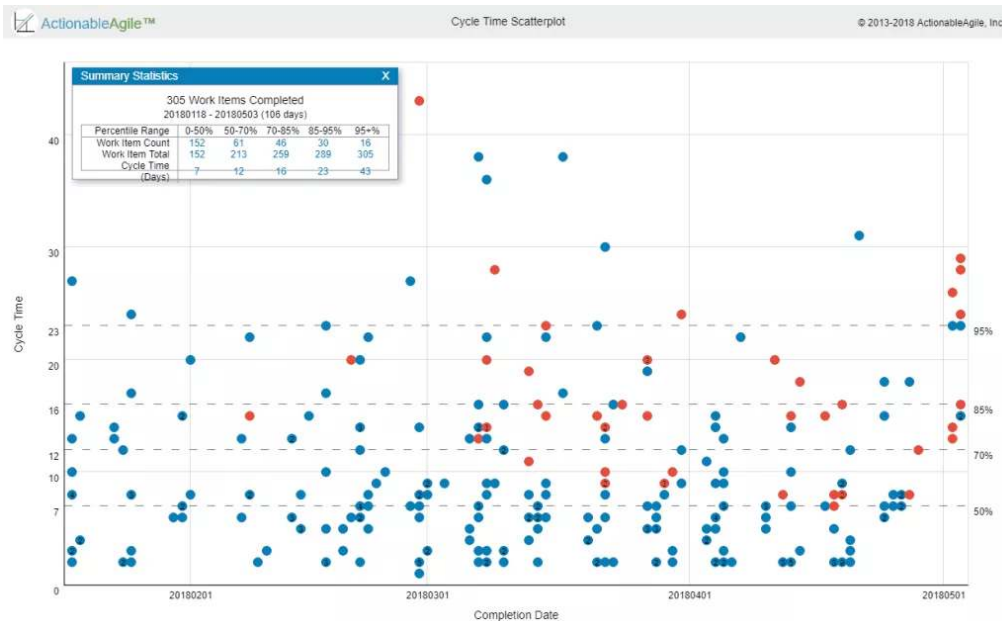


Cycle Time

The amount of elapsed time between when a work item “starts” and when a work item “finishes.”

This metric is a lagging indicator of flow. It is available only after an item is actually finished from the workflow perspective (e.g. reached a Done lane on the Kanban board). It is typically used to drive improvement work as well as to be able to establish internal/external expectations as to the team’s turnaround time on specific items.

Cycle time is mentioned briefly in SAFe but the SAFe practitioner would benefit from taking a deeper look at charts/reports used to visualize and analyze Cycle Times like the Cycle Time Scatterplot where teams can understand their Cycle Time trends, distributions, look at anomalies.

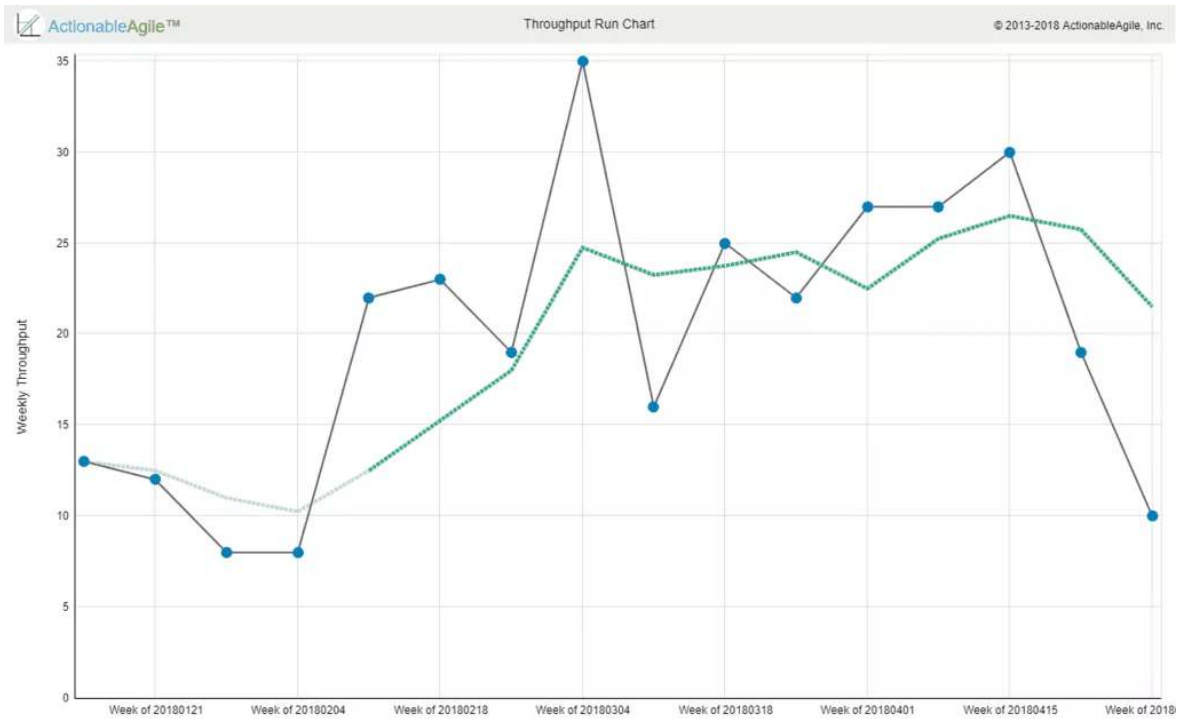


Throughput

The number of work items “finished” per unit of time.

Note the measurement of throughput is the exact count of work items, without any compensation for item size – which is a major difference between throughput and story-points based velocity. Throughput is measured at a certain step in the workflow, typically at the finish line of the workflow. Throughput can be visualized via a separate run chart or by looking at the angle of curves on a Cumulative Flow Diagram.

While it does include throughput in its list of metrics, Classic SAFe emphasizes measuring Velocity for the purpose of planning. My advice to SAFe practitioners is to start to track throughput and compare it's predictive powers to those of velocity and either combine the two approaches in order to improve their predictability (e.g. during PI Planning / Roadmapping) or choose one that works better in their context. (Read more about this in my earlier blog post about [SAFe Without Estimates](#))



Work Item Age

The amount of elapsed time between when a work item “started” and the current time.

WIP and Cycle Time are classic metrics every Kanban/SAFe practitioner is probably familiar with and throughput is somewhat similar to Velocity.

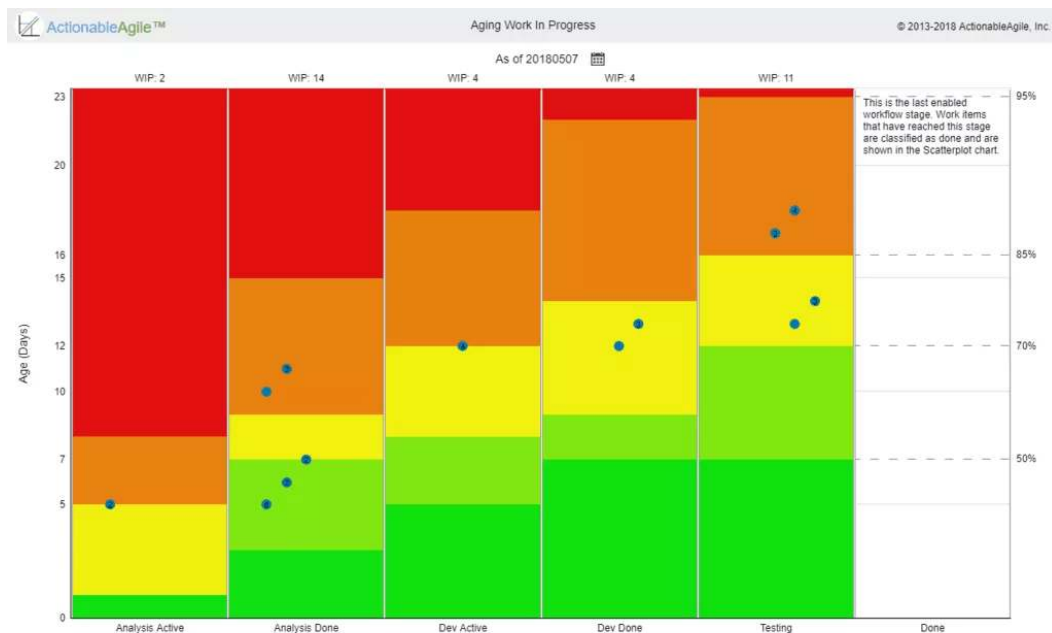
Work Item Age is the new guy on the block. Work Item Age complements Cycle Time. If Cycle Time is a lagging indicator only relevant for finished items, Work Item Age is a leading indicator only relevant for non-finished items. The basic idea is to provide transparency to which items are flowing well and which are sort of “stuck” even if not formally blocked.

I’ve been using some variant of this metric with most Kanban teams I’ve worked with. I also worked with several Kanban tool vendors to introduce some way to visualize card/item age.

Age on its own is interesting but not enough. We also want some indication of flow health. One common thing to visualize is the age in the current step in the workflow also known as “cards that didn’t move recently”.

Another way to look at it would be to look at the overall age but combine it with where the work currently is in the workflow as well as what the team expects their cycle time to be (We call that expectation Service Level Expectation (SLE) in the Kanban guide for Scrum teams and the PSK class). Combining all this information can help the team focus on the items that are at the most risk of missing the team's expectations/SLE. For example, let's say a team has an SLE of 16 days with 85% confidence. If one of the cards on their board has an age of 10 days, is that ok? is it a problem? The answer is that it depends. If that card is very close to the end of the workflow it is probably not a problem. If it is very close to the start of the workflow it is probably an indication of a problem that requires attention. The "Aging Work in Progress" chart below provides this perspective of both where active items are in the workflow, what the typical cycle times for this team are, and based on that which items are indications of flow risks (obviously orange-red means very low probability of finishing within the team's flow expectations).

To sum up – Work Item Age is the best metric to look at if you want to determine when an item that has already started is going to finish. This is in contrast to an item that hasn't started – where your best bet is your historical Cycle Times. The Service Level Expectation is just an expectation set by the team to themselves answering the question "What Cycle Time do we expect to see for an item of this type, and what is our confidence level for this?".



Note: The charts above were created using the demo version of [ActionableAgile Analytics](#) – a tool created by my co-steward of the Professional Scrum with Kanban class – Daniel Vacanti. You can access the demo yourself and play with these metrics and think about how they would help your Agile team, SAFe ART/ST or portfolio. Note that the definition of Workflow in this chart is just one example of how your workflow might look like. Each team would define their Workflow according to their context.

Using the Flow metrics in SAFe's team-level

First, we need to discuss how to even apply these metrics for the team-level. On the face of it, should be quite trivial because this is familiar ground – The Team Backlog and Iteration Backlog are pretty familiar to the Product Backlog and Sprint Backlog. And yet a good question raised by Travis Birch in a LinkedIn thread about this is “Where in the workflow of these items do we start to measure cycle time and include the item in WIP?”.

In general in the SAFe team level most work is represented by User Stories that are a breakdown of a Feature from the Program Backlog / Program Kanban. These User Stories start their life on the Team Backlog as the story is identified during feature breakdown. I would typically consider the item started once the team starts to refine the story in prep for an actual iteration. This means that story identification and high-level assignment to Iterations during PI Planning isn't considered actively working on the User Story yet. Same would apply to other items in the Team Backlog.

When should we consider items in the team level Kanban board finished and out of WIP and stop the clock on cycle time? At the moment they are Done according to the team-level definition of Done. Typically – once they finish fixing whatever problems they find in story-level testing and the story is accepted. At this point these Stories sit and wait for their friends in the feature to join them and only then does the Feature move along. but from the Story-level perspective, their workflow is already finished.

So now that we've mapped the metrics to the team-level how can these flow metrics be used to improve the Scrum events used in SAFe's team level? This is one of the key learning objectives in the [Professional Scrum with Kanban](#) class. In a follow up discussion with some of the Professional Scrum Trainers who attended one of these classes, we came up with a matrix mapping the metrics to the events. (credit [Maarten Kossen](#))

	WIP	Cycle Time	Work Item Age	Throughput
Sprint Planning			*	✓
Daily Scrum	✓		✓	
Sprint Review		✓	**	✓
Sprint Retrospective	✓	✓	x	✓

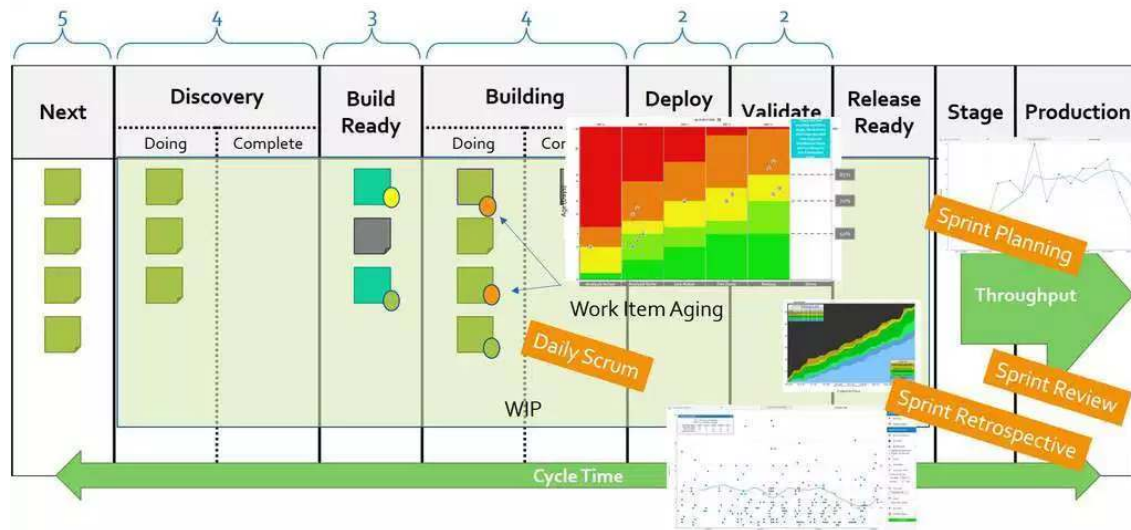
I'll explain (using SAFe's names to help map to SAFe):

Iteration Planning mainly leverages Throughput in order to create a realistic forecast for the Iteration Backlog. Work Item Age might be relevant when you have some items left over from the previous Sprint and you want to decide what to do about them.

The focus of **Daily Standup** is the ongoing flow within the Iteration so naturally what we care about is what's currently going on. Therefore, Current WIP and Work Item Age are the most important metrics in the Daily Standup.

Iteration Review includes a review with stakeholders of both the Increment as well as overall flow behavior of the team – trends in Cycle Times and Throughput are interesting. Throughput can also be used as part of release planning/road-mapping discussions, especially when combined with Monte-Carlo simulations provide some better visibility/confidence into "What can be done by when". NOTE: It is always important to emphasize that these are projections/forecasts, not commitments.

Iteration Retrospective is all about inspecting and adapting the process and the workflow. Therefore it is the place to look at WIP, Cycle Times, Throughput from a perspective of looking for areas to improve.



Applying the Metrics in SAFe's Program/Large Solution Level

So far it was an easy mapping because SAFe uses Scrum events at the team level. How about using these flow metrics in the Program level on your Agile Release Train?

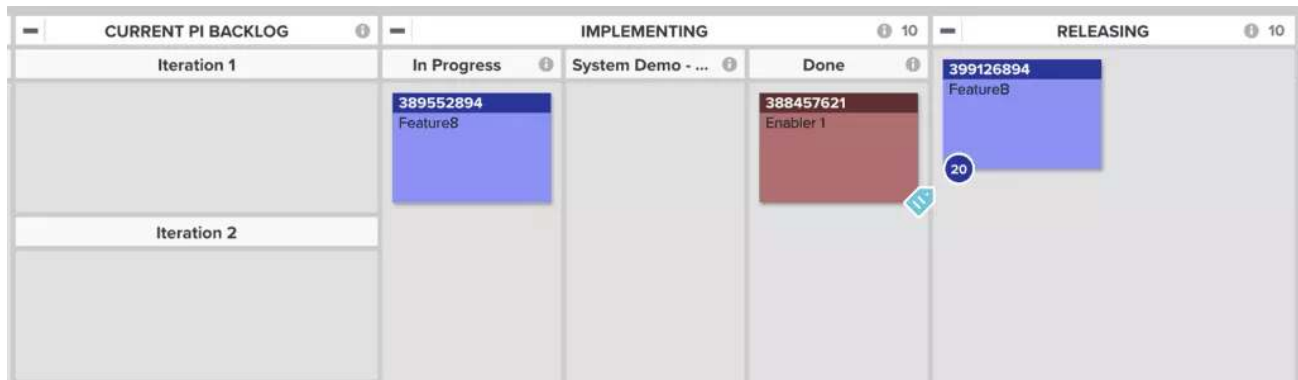
SAFe's program level kanban items are features/enablers. The wider perspective here would be to consider them work in process and measure cycle time from the moment we identify them during Exploration to the moment they've been Released, all the way through Integration and Deployment all throughout the SAFe Continuous Delivery pipeline. In many environments though this would be a very long time (especially when Continuous Delivery is just an espoused vision) and it would be interesting to understand the WIP and cycle time in each one of these major stages.

When it comes to program-level events – **Program Increment (PI) Planning** should mainly leverage Throughput in order to create a realistic forecast for each team's capacity in their iterations. Work Item Age might be relevant when you have some items left over from the previous PI and you want to decide what to do about them.

Beyond looking at throughput at the team level, it is interesting to track throughput at the Program-level. One could argue that at this level it makes more sense to take into account Feature size because Features are so varied in size. I'd argue you want to also look at throughput because as we know from WSJF we want to reduce batch size and deliver more frequently and analyzing throughput helps us see how far we are from that vision.

The focus of **Scrum of Scrums / ART Sync** is the ongoing flow across the program during the PI so naturally what we care about is what's currently going on. Therefore, Current WIP and Work Item Age are the most important metrics in the SoS/ART Sync. We should pay attention both to overall trends for story-level Age across teams, as well as Feature-level Age on the Program Kanban board and look for features that are aging dangerously.

In order to get better visibility to the flow of features, I like to split the "Implementation" stage in the Program Kanban Board to some interim stages that reflect the state of the feature. One way to do that is to add a "System Demo" or "Alive" lane that reflects whether this feature was demonstrated already – which means it's not complete yet but we've seen some of it working. You might come up with other ways to split "Implementing" but be careful of having a "Design" "Build" "Test" split because we don't want to develop the Feature using waterfall – what we want to see is stories in different stages of the development cycle while the feature is in implementation. you could have a "first story demonstrated" lane and "final stretch" lane if you want...



System Demo in SAFe focuses on the demonstration of the System Increment. An aspect in the Scrum Guide that SHOULD be added is a review with stakeholders of both the System Increment as well as overall flow behavior of the ART – trends in Cycle Times and Throughput are interesting and can be used to adjust the plan for how to deliver the PI objectives.

Inspect and Adapt is all about inspecting and adapting the process and the workflow. Therefore it is the place to look at WIP, Cycle Times, Throughput from a perspective of looking for areas to improve at both an aggregative team-level view as well as a Feature-level view.

Applying the Metrics to the Portfolio Level

The portfolio doesn't have a prescribed cadence of events, but you could see how metrics such as Cycle Times, WIP, Aging and throughput could complement your perspective and help you visualize, limit and reduce WIP as well as improve Innovation Flow at the portfolio level.

The Human Aspect Of Metrics

One of the key tenets in SAFe AND Scrum is Respect. Respect for people and culture. When introducing any sort of new practice including flow metrics we should respect people and invite them to try these metrics rather than force it upon them. One of the key points the Scrum Guide and the Kanban Guide for Scrum Teams emphasise is the fact that the team owns their definition of Workflow and what metrics they use. Team is the Dev team when it comes to the SAFe Team level, the ART when it comes to the Program level, and so on.

IF we introduce the metrics to the team this way and invite them to engage, my experience shows that their collaboration, engagement, and inclusion of the whole team in tackling issues grows as a result.

Of course if we force/mandate these metrics and use them to micro-manage the team into submission then your mileage may vary somewhere between mutiny, Italian strike ([Work-to-Rule](#)) or just lower engagement/motivation. Aiming to smooth flow also results in diversifying the knowledge and skill level in the team and reduces the dependency on specific team members and heroism.

(Thank you [Simon Powers](#) for making sure I tackle this in this post!)

There's more to Kanban than you find in SAFe's curriculum

And that's natural and expected. SAFe is a framework, it cannot include everything. It introduces many people to Kanban and then invites them to go on a lifelong learning journey. Hopefully this journey includes exploring in more depth bodies of work like the Scrum Guide and now the Kanban Guide for Scrum Teams and deeper Kanban understanding in general. SAFe practitioners especially interested in metrics should definitely check out [Daniel Vacanti's book](#) and [workshop](#).

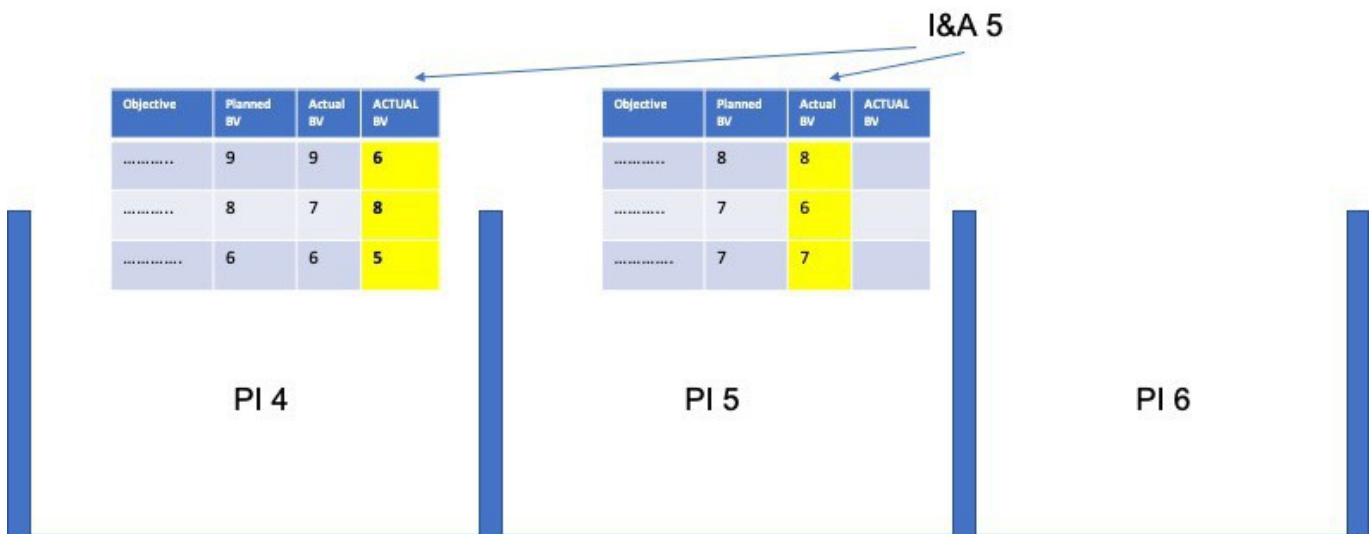
NOTE: I'm planning a special Professional Scrum with Kanban class for SAFe practitioners. This will be a formal PSK class with the exercises and discussions exploring SAFe-specific contexts at the team and program level. If you're interested, drop me a note over at the [AgileSparks PSK page](#).

Share this:



Like this:

Loading...



By [Yuval Yeret](#) Posted [October 23, 2018](#) In [Scaled Agile Framework](#)

Actual is a relative term when it comes to business value delivered by a SAFe PI Objective. We had a discussion about this a couple of weeks ago in an [Implementing SAFe](#) class and I promised a blog post about this. Here it goes.

Planned Business Value – Making sure Business Owners and the Agile Team are on the same page

Let’s start from the basics though. PI (Program Increment) Objectives are used as a “back briefing” mechanism by Agile Teams on an Agile Release Train to share their plan for the PI and validate that they are indeed focusing on the highest priorities and are planning to deliver objectives that will be valuable for the business.

Business Owners (Business Executive, Product Management, System Architect) circulate between teams during PI Planning and score each PI Objective on a scale of 1..10 based on business value they ASSUME will be delivered in case this PI Objective is accomplished in the PI.

This becomes the “Planned Business Value (BV)” for that objective.

Actual Business Value – Assessing Business Value based on Demo of a real solution

Later on, during PI Inspect & Adapt (or earlier during System Demos) the same Business Owners circulate between the teams and score each of these PI Objectives again, this time on “Actual Business Value (BV)”.

What does Actual mean here? Well, in most cases the evaluation is based on seeing a demo of a working solution and still making ASSUMPTIONS about what the actual value would be when the solution meeting this objective will be released and available to real users/customers.

Sorry, but while being much closer, that’s still not ACTUAL business value.

ACTUAL Actual Business Value – Based on released solutions and the outcomes they deliver in the real world

ACTUAL business value delivered can be evaluated only AFTER the solution is released.

On most Agile Release Trains / SAFe contexts the PI I&A is too early to make this evaluation so you could understand why we’re still making assumptions at that stage.

But if we really care about outcomes and delivering value, we shouldn’t close the books on these PI Objectives and the PI at that point. We should get back to it later on and Inspect and Adapt based on real business value delivered.

Adjusting the SAFe Inspect & Adapt to track ACTUAL actual Business Value

If you’re with me, you’re probably asking what can we do about this? What is the right timing to get back and assess the ACTUAL actual Business Value? From a cadence perspective there are two main ways to do this.

The simplest is to take advantage of the Inspect and Adapt PI System Demo to review the ACTUAL business value delivered by the objectives in the PREVIOUS Program Increment. E.g. if we’re now finishing PI 5, we’re assessing actual business value delivered by the objectives in PI 5 that will be released sometime during PI 6, as well as PI 4 objectives that hopefully got released during PI 5.

I&A 5

Objective	Planned BV	Actual BV	ACTUAL BV
...	9	9	6
...	8	7	8
...	6	6	5

PI 4

Objective	Planned BV	Actual BV	ACTUAL BV
...	8	8	
...	7	6	
...	7	7	

PI 5

PI 6

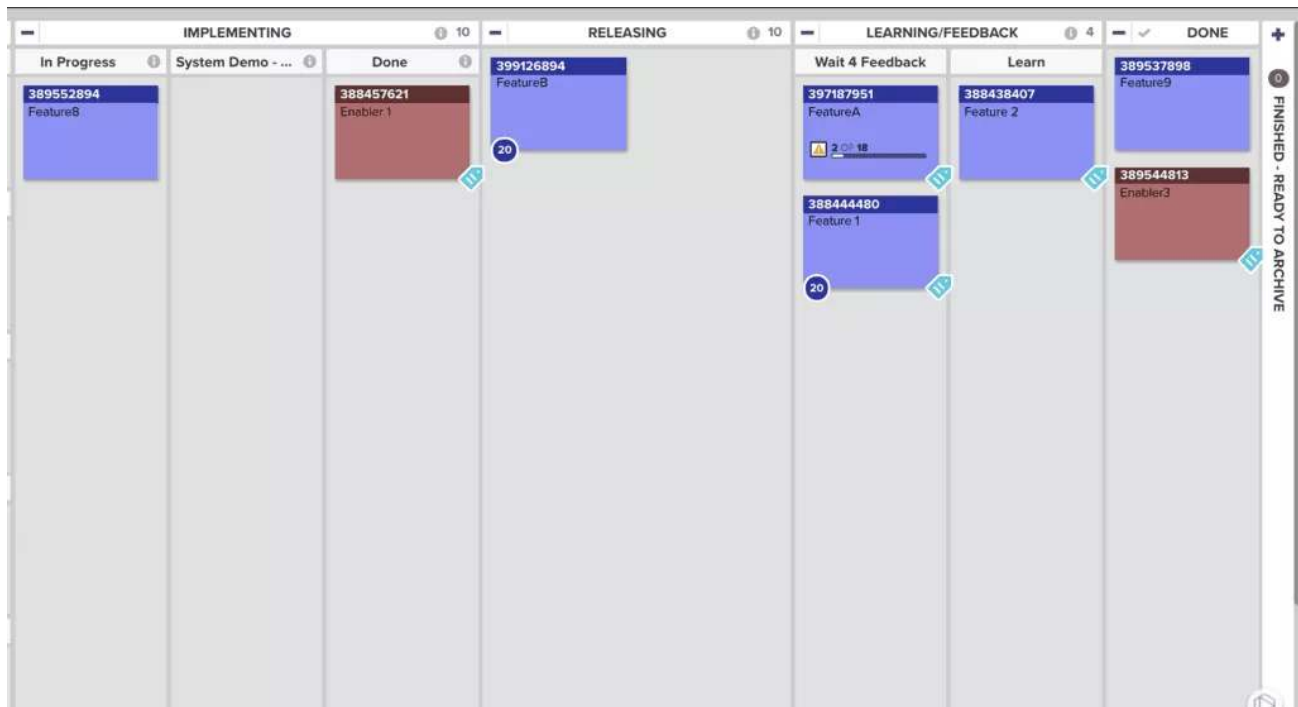
For each one of these PI 4 objectives, now should be a reasonable time to talk about things like – Have customers started to use this solution? Are they happy with it? Did it achieve the impact we had in mind for it? Did we stop incurring the cost of delay we had in mind when prioritising this? At this point, the 1..10 score should be data/evidence-driven.

If we AREN'T able to evaluate the actual business value at this point – that means there's a short-circuit in our empiric feedback loop that we should work on fixing.

If we haven't released the solution yet, then we should keep the actual score for this objective empty and get back to it in the next PI. This objective should still be "work in progress" from our perspective.

It's not DONE until we evaluated the ACTUAL Business Value

You might guess what's the next aspect of this. Mentioning Work in Progress should be an obvious clue. The Program Kanban has a role in helping us out here as well. Features on the Program-level Kanban shouldn't be considered DONE until we collected this feedback and evaluated the ACTUAL actual business value on them. They should hang out on the board – maybe in an area called "Feedback" or whatever you prefer.



I've been recommending this sort of Program-level Kanban Board structure for a long time now. Some of my enterprise-level clients have improved their Product Management practices dramatically through the accountability and follow-through that this practice encourages.

Just think about the impact on Product Management, Business Owners, Sales people asking for features, if they know they are accountable to the outcomes from these features after they're released.

Who's accountable for delivering the actual business value?

This brings us to an interesting question. Who's accountable to delivering actual business value? Who's accountable to delivering ACTUAL actual business value? Is it the Agile Team? Product Management? Business? Sales?

I've seen way too many teams frustrated when they deliver the objectives according to what they presented as the plan, and yet the actual business value score is lower than the plan because the Business Owners don't think as much value will be actually delivered. When we're moving from assumed actual to actual actual the gap can be even bigger. On one hand, in the spirit of transparency and being focused on value and outcomes rather than output, this is the right way to score the business value. It's about value delivery rather than tracking to plans. On the other hand, you can probably understand the frustration here.

The way I see it, the only way out of this is to understand that the PI Objective plan vs actual vs ACTUAL isn't an indication of the individual performance of either one of these roles. It's an indication of the performance of the whole development value stream including the upstream activities related to choosing and prioritising features and the downstream activities related to selling the solution, convincing users/customers to use it, implementing it in the field, operating/supporting it.

That, together with Lean/Agile Leadership that emphasises principles such as Assuming Variability, Objective evaluation of working delivered systems, and relentless improvement of the whole value delivery cycle, is the key to focusing on learning from these surprises whether they are systemic and repeating or rare exceptions.

A relentlessly improving organisation would inspect what's the trend when it comes to plan vs actual vs actual actual for the whole program and per specific PI Objectives and try to see what it can learn from when the value gap happens and does it happen for a specific type of objectives or in a specific area of the program/business.

It's all about Value

Value is the goal of Lean and the fast delivery of value is the goal of SAFe. If we're serious about that, We should raise our game when it comes to managing value as a first class citizen in SAFe. Business value on PI objectives is the perfect place for doing exactly that.

So, Next time you have PI Inspect & Adapt, don't just look at the PI you're finishing just now, take a look at the objectives from the previous PI as well. And on your Program-level Kanban only consider features done after evaluating actual business value delivered, ideally based on quantifiable metrics.

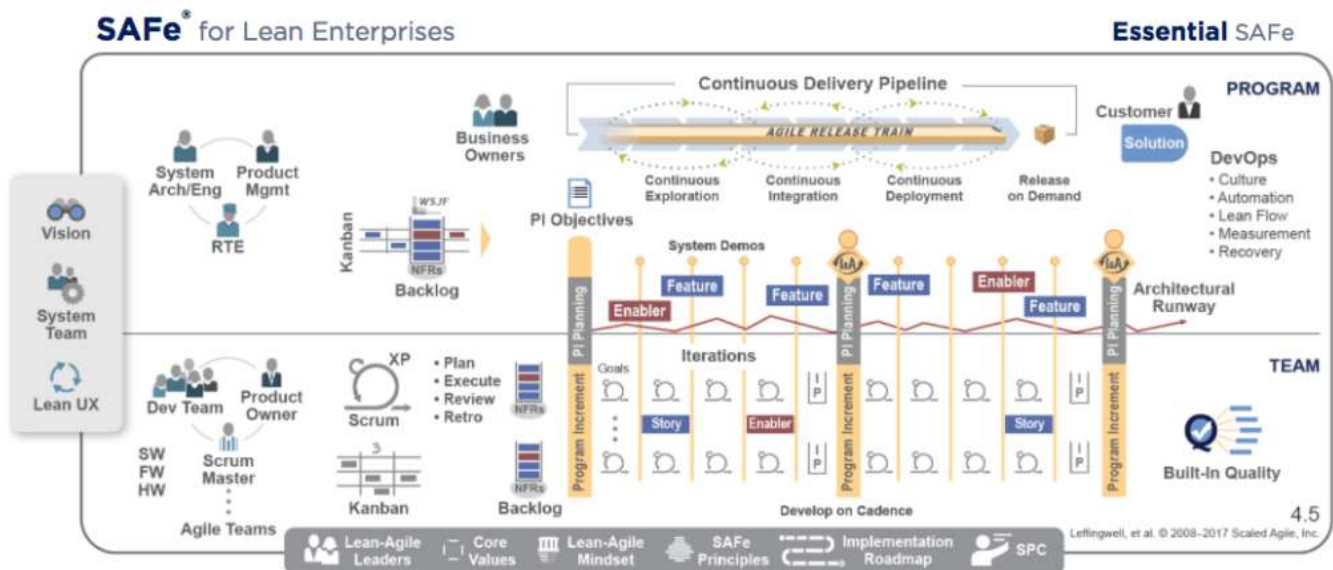
I love it when discussions at [class](#) drive me to write up some of my experiences, tips and tricks for the blog. Awesome kudos to my students, now [SPCs](#) off to implement a healthy and value-oriented SAFe in their organizations!

Share this:



Like this:

Loading...



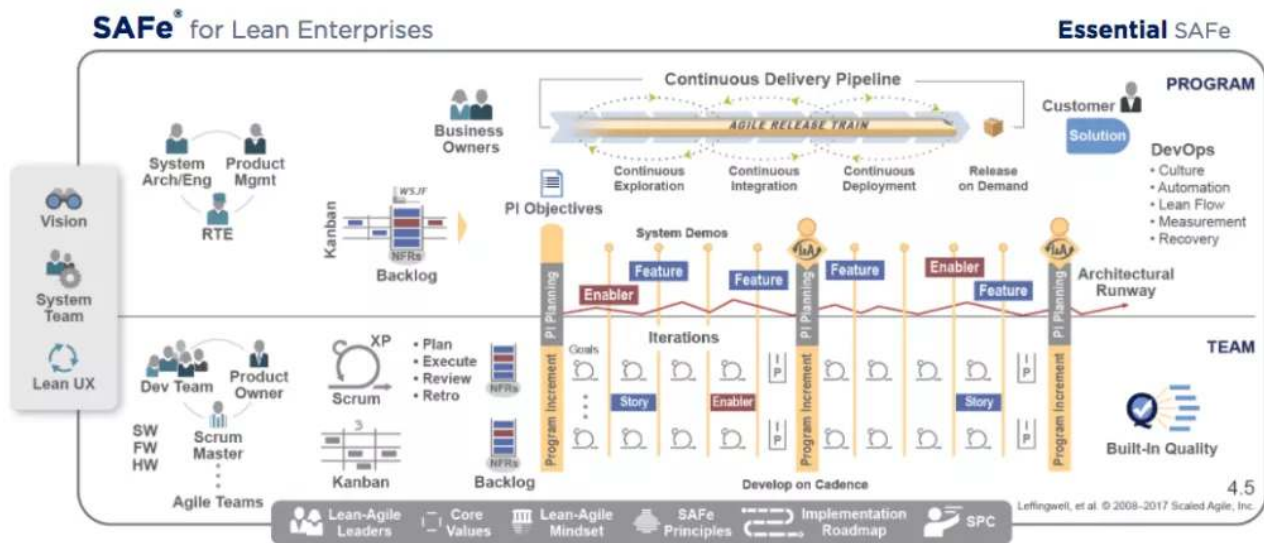
By [Yuval Yeret](#) Posted [July 27, 2017](#) In [Agile Marketing](#), [Scaling Agile](#)

Focusing on SAFe™

In the previous article in the Scaled Agile Marketing series I provided an overview of how Scaled Agile Marketing looks like. This time around I want to provide some more details on one of the approaches I mentioned for implementing Scaled Agile Marketing – the Scaled Agile Framework (SAFe™).

Why SAFe? First of all it is the most popular scaling approach these days and so many marketers will find themselves in organizations where SAFe is actually used in IT/Technology and the option of using it in Marketing as well will come up. As a result of that it is also the scaling framework I've actually had a chance to use in a marketing context with good results.

Let's make sure we cover the SAFe Essentials



SAFe has several configurations that can range from the “Essential” configuration through “Portfolio” and “Large Solution” configurations all the way to “Full SAFe” which includes all configurations together. In this article I will start with the “Essential” configuration and mention “Portfolio” towards the end. The “Essential” configuration isn’t just a version of SAFe it is also a set of **10 essential elements** without which you might be scaling agile but you aren’t really using SAFe to do it.

I actually covered some of the essential elements in the previous article:

- #1 Lean/Agile Principles
- #2 Real Agile Teams And Trains (In the “The Agile Marketing Team of Teams” section)
- #3 Cadence and Synchronization
- #4 PI Planning (covered in the “The Agile Marketing Team of Teams” section as well)
- #7 Inspect and Adapt (Learn at the System Level)
- #10 Lean Agile Leadership

So let’s look at the remaining essential elements:

- #5 – DevOps and Releasability
- #6 – System Demo
- #8 – IP Iteration
- #9 – Architectural Runway

#5 – DevOps and Releasability -> MarOps and Releasability

This is an essential that requires slight tweaking for a Marketing context: SAFe Agile Marketing organizations aim to break down silos between marketers and marketing technology (MarTech) and operations. Each Agile Marketing Train should be able to continuously run marketing experiments or deliver new marketing plays/campaigns to the live customer/buyer journey. Over time, the separation between marketing and marketing tech/ops is significantly reduced and marketing trains operate with an automated continuous delivery pipeline that includes easy instrumentation and measurement to enable continuous experimentation and validation of hypothesis.

#6 – System Demo

When we [defined Agile Marketing](#) we talked about some of the key things we value – “Validated Learning”, “Customer Discovery”, “Adaptive Campaigns” among others. One value that isn’t explicitly mentioned in the Agile Marketing manifesto but is implicitly required to achieve these is “Working Marketing” meaning objective observation of working marketing deliverables rather than lengthy comprehensive documentation/designs/plans. I used to tell agile development teams that unless they’re the Microsoft PowerPoint development team their demos shouldn’t be running PowerPoint. In a marketing context I cannot say that anymore because sometimes a PowerPoint deck IS the marketing deliverable but you get my drift.

We should frequently look at real marketing deliverables so we can discover whether they really drive the customer journey experience we are looking for as well as get a real feeling as to progress towards our goal. In a scaled context where we have multiple marketing teams working on a larger customer journey or marketing campaign, it’s crucial to frequently integrate the whole marketing story using real deliverables, get some feedback on it, and adjust course if necessary. This is the intent of the System Demo in SAFe. *Every two weeks, the full system – the integrated work of all teams on the marketing train for that iteration – is demoed to the train’s stakeholders. Stakeholders provide the feedback the train needs to stay on course and take corrective action.* In a marketing context we probably need a better name for this. Any suggestions?

#8 – IP Iteration

The Innovation and Planning iteration occurs every Program Increment (8-12 weeks typically). Since we don’t plan specific content for the IP iteration it can act as an estimating buffer for to help meet your PI objectives. In addition it and provides dedicated time for innovation, continuing education, PI planning and Inspect and Adapt events.

#9 – Architectural Runway

In product development, Architectural Runway refers to side work that needs to happen to support in order to support fast and clean implementation of high priority near-term features. In a marketing context it refers to marketing technology/infrastructure that needs to be in place to support upcoming high priority marketing plays/campaigns/activities (think for example a lead nurturing solution in case we plan to do lead nurturing in the next PI), exploration/research, and maybe some key architectural components like a page template or a slide deck template or brand guidelines that reduce the amount of effort when getting to work on actual marketing plays.

Essential SAFe works pretty well for a Marketing context, with some limitations

As you can see applying the 10 essential SAFe elements to a marketing context isn't too hard. There are some modifications but at this high level the mapping works. This doesn't mean that Agile Development is Agile Marketing. What it does mean though is that once you have a good team-level agile marketing process/structure, there's good applicable guidance for how to scale it. This is exactly what we've seen in CA Technologies when we applied SAFe to a marketing context. Our main challenge was and is changing leadership mindset especially around decentralized control and organizing around customer focus and de-emphasizing the marketing specialties/silos as well as driving a real learning/experimentation mindset and process at all levels.

When it comes to leveraging SAFe the main challenge we had is that SAFe was designed for a product development context and therefore the materials and knowledge base which are one of SAFe's biggest advantages weren't a good fit for us. We actually started to use one of SAFe's workshops to train the teams but realized midway that there's too much development language and examples so until we create either a more neutral version of SAFe or a marketing-focused variant, the best we can do is leverage the practices without relying on the great materials. It was also apparent that marketers prefer lighter-weight methodologies/frameworks and mainly didn't have the patience to learn about SAFe in depth. The essential elements were all that could fit their attention span.

This combination of incompatibility of the materials and the distaste for formal lengthy training and a large set of practices also meant that when it came to SAFe expertise it was crucial to have people around that don't just recite SAFe gospel but also have a deep understanding of the principles and are able to adapt SAFe to other contexts without killing its spirit.

This is of course true for scaling agile marketing in general, regardless of whether you're using SAFe, LeSS or any other approach. You'll be working in exploration mode trying to identify the right language, process, structure. In most cases marketing people will give you limited attention. Make sure you have somebody that can support you in that mode rather than just teach you a methodology.

There's one thing I would always add to Essential SAFe

Yes, I know, the thinking is to keep it to bare essentials, and 10 elements is better than 11, but there's one key concept and practice that is part of SAFe, is portrayed in the Essential SAFe big picture above, but isn't mentioned as a key element here. It is also one of my favorite focus areas. Enough clues for now. I'll let you try to figure that one out until the next article which will focus on this topic...

Also published on [Medium](#).

Share this:



Like this:

Loading...



About AgileSparks

AgileSparks is a global boutique agile consulting firm with offices in Israel, Boston, India and 20 full time coaches on staff.

AgileSparks' experts have proven experience in transforming over 300 companies worldwide, from small companies/startups all the way to some of the largest Fortune 500 companies using a variety of techniques and practices in the agile domain.



Our lead coaches ran “Boost”/”Tune-up” workshops and initiatives in dozens of companies. These days this is actually our most popular service, as more and more companies are on one hand already practicing some agile/scrum aspects, but on the other hand realize the real potential of breaking through the glass ceiling. Many of these companies have been practicing a form of “Scrum/Agile Theater” and are now realizing it and looking to do agile properly and professionally. We have a suite of services focused on this space - ranging from “Agile Boost Camp” through our “Agile tooling tune-up workshops for JIRA/TFS”.

We’ve taught thousands of students how to be effective agile team members, Scrum Masters, Product Owners, and other roles - both beginner’s classes all the way to advanced levels.



Most of our clients apply Agile at the enterprise scale. We are a prime choice for such organizations looking to figure out what's the right approach for their context – How to scale effectively and how to implement the framework of your choice in a way that respects people and culture. Yuval Yeret, our SPCT, is a thought leader in the area of using an Invitations/Pull-based SAFe implementation approach based on our experience in the field. Our team of experienced SPCs have all worked in multiple enterprise-scale agile implementations using SAFe, LeSS, Enterprise Kanban and other home-grown approaches.

When we use SAFe, We see it as a Framework. Successful implementation requires using the framework effectively according to your context. We're very careful of "SAFe in name only" that results from a combination of thoughtless mandates and dogmatic/insensitive "by the book" implementation. We start our engagements with a unique SAFe implementation strategy workshop with leaders of the organization to create a SAFe implementation blueprint and roadmap. This workshop is combined with a SAFe-certified Leading/Implementing workshop thereby providing both in-depth SAFe knowledge with a customized implementation strategy.



AgileSparks SAFe Services

Our approach at AgileSparks is to combine the SAFe's Implementation Roadmap with our own AgileSparks Way for managing the change, which is based on our vast global experience. Our Key SAFe services are:

- Open enrollment and in-house Leading and Implementing SAFe workshops to inspire people to start to think about SAFe and initiate the journey.
- Agile Release Train launches – From Leading SAFe combined with an Implementation Strategy workshop, through preparation activities like SAFe Scrum Master and SAFe PO/PM workshops as well as program backlog preparation, WSJF exercises, Team formation, all the way to SAFe for Teams, SAFe DevOps Practitioner, and initial PI Planning. In this phase, we combine SAFe best practices with our own AgileSparks Way building blocks.
- We coach the Agile Release Train on its journey to help stabilize the first SAFe Program Increment (PI).
- We facilitate the Agile Release Train Inspect & Adapt Workshop as the Agile Release Train approaches its first destination
- We scale the organization through Portfolio and Value Stream setup and coaching, leveraging our thought-leadership on Enterprise-level Kanban used as part of the SAFe Portfolio, Solutions, and Program.
- We scale SAFe success in the organization by applying the AgileSparks Way/Implementation Roadmap to more and more Value Streams/Trains. The main ingredient of success across the enterprise is to choose the right transformation/change management model for the context. Whether it is a bold decision to go SAFe across the board or to provide SAFe as an option and invite leaders of trains/value streams to come onboard.
- We help Release Train Engineers continue their SAFe learning journey by running the Release Train Engineer workshop.
- We extend agility to the wider company – for example by applying SAFe in a Marketing context.



We are a Scaled Agile SPCT Gold Partner with 2 SPCTs (Yuval Yeret and Vikas Kapila) and 12 SAFe Program Consultants (SPCs) on staff.



I'm honored to have Yuval actively participating in the Scaled Agile Framework community as both a contributor to the framework as well as a SAFe trainer and practitioner. I'm impressed with the depth of his knowledge and experience as well as self-confidence and gravitas. I will not hesitate recommending Yuval to any organization looking to implement Lean/Agile at scale.

Dean Leffingwell - Chief Methodologist and CEO - Scaled Agile, Inc

We are also Scrum.org Training Partners, Providing professional Scrum Master, Team, Product Owner, and Scrum with Kanban classes:

AgileSparks is a highly professional Scrum.org partner with a team of people who have a deep understanding of the Scrum theory, practices and its real-world application, along with complementary practices such as Lean, Kanban and others. We believe they are a great partner for your agile/scrum journey. They are a member of our Professional Training Network (PTN) partner community and have multiple of our Professional Scrum Trainers (PSTs) on staff. They can also reach more broadly to our community of over 195 PSTs if needed to expand their reach and coverage. All of our trainers go through extensive training and validation of their knowledge and abilities with Scrum before being approved to become a PST. A process that is overseen by Scrum co-creator and Scrum.org founder Ken Schwaber.

Dave West, CEO, Scrum.org



About The AgileSparks Implementation Approach

AgileSparks' full implementation approach consists of several stages. The proposed Assessment is part of the first step "Plan and Initiate".

1. Plan and initiate – plan and prepare for Agility.

- a. Initial Assessment that comprises interviews with leaders, key stakeholders, various players in the group, review of activities, artifacts, ceremonies.
- b. Thinking/Planning Assessment and Focus Management Workshop (usually 2 days - in this case 4 days combined with the Implementing SAFe training)
 - Map the group goals and current situation, set the goals for the new process
 - Introduction and discussion of key Lean/Agile solutions with a focus on the group's context and goals
 - Understand how Agile approaches can help the group reach its improvement goals (discuss small batches using Agile Requirements, Scrum, Kanban, Scrumban, Agile releases, Lean Startup approaches, commitments in Agile, etc.)
 - Design the high level process approach to delivery and improvement for the group and understand its implications and implementation steps
 - Establishing buy-in and vision for the process.

Key Deliverables:

- List of pains/ opportunities
- Goals
- Common language
- Management buy-in including willingness to pay the "price"
- High level plan for the journey and more concrete plan for **Kickoff** including an Implementation Program Increment plan for the first implementation increment (10-12 weeks)

2. Kickoff – achieve basic Agility

Key Deliverables:

- All initial process and building blocks deployed and can be observed in operation
- Organization is starting to learn the new way of working and gets ready to **stabilize**



3. **Stabilize** – reach competent Agility with the current process

Key Deliverables:

- Reached the agreed upon goals
- Reduced significantly the identified pains
- Current process feels reasonably ok, feeling we are after the initial storming.
- Can sustain current process/practices even without support/coaching
- Can expect limited amount of further improvement/tweaks will happen without support/coaching
- Have updated goals/pains to address in the **Improve** phase

4. **Recharge** – recharge organizational energies for further improvement

- a. "Digest" the new process and maintain stable operation

Key Deliverables:

- Recharged energies and now ready to take significant improvement steps

5. **Improve** - Competent at adapting/ improving

Key Deliverables:

- Continuous improvement capability - can expect continued stream of significant evolutionary improvements without external coaching
- Reached the agreed upon updated goals
- Reduced significantly new pains identified at end of Stabilize
- Organization is familiar with, and knows how, to use a set of key continuous improvement techniques and tools – Control Charts, Five Whys, Retrospectives, A3, etc.

6. **Closure** – Learn from the journey

Key Deliverables:

- Lessons for future change initiatives
- Identified next steps to ensure continued improvement
- Identified and implement mode of external coaching support/maintenance
- Identified opportunities to leverage agility elsewhere in the organization

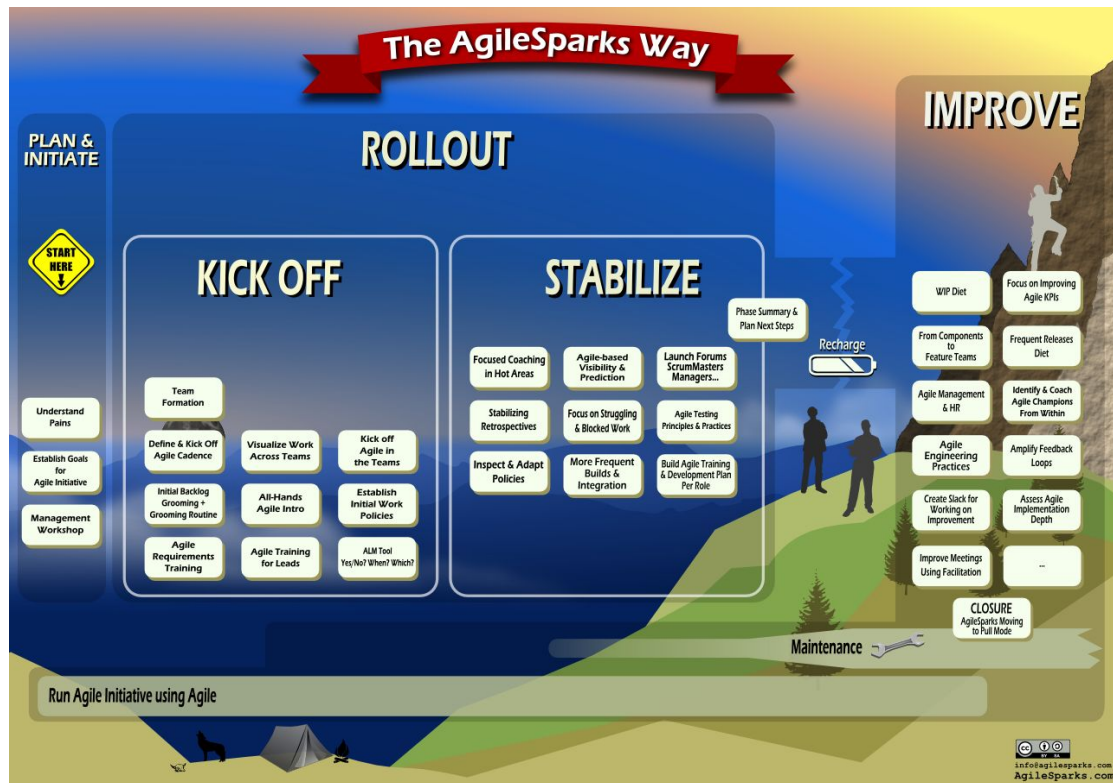
7. **Maintenance** – Keep the fire burning

Key Deliverables:

- Periodic meetings with role players, to support and keep energy levels high
- Support improvement initiatives
- "Health checks" to identify areas that need attention



The phases above are depicted and summarized in the picture below:





Appendix C - AgileSparks Reference Accounts

Enterprise Software Accounts:

- Siemens PLM Manufacturing segment - Support a multi-year lean transformation journey leveraging Kanban, Scrum, Lean Startup in multiple groups and teams in the **US, Netherlands, Germany, Israel, India, China**. The story of ONE of the sub-groups of the segment was presented in Agile Israel 2015 - <https://www.slideshare.net/AgileSparks/agile-israel-2015-enterprise-2-going-agile>
- Siemens SES / Vistagy - Accompany a legacy highly-complex line of business located in **Boston** in transforming to cross-functional scrum teams with real self-organization, storming and all, managers as coaches, small stories, MMFs/MVP thinking.
- Intel - worked with several groups (hundreds of people) in the **US, Israel, India, Germany, Singapore** to implement agility, including groups developing IOT Big Data advanced analytics. Training scrum for more than 8 years via public and private courses. Implementing SAFe and agile at scale. See story from one group: <https://www.slideshare.net/AgileIsrael/making-scaling-agile-safe-at-intel-yariv-weltschcohen-agile-israel-2014>
- Aras Software - Enterprise PLM product with offices in Andover MA and Eastern Europe.
- HP Software - Helped multiple product groups located in **Israel, Germany, India, China, Ukraine** go beyond shallow agility towards real Lean/Agile/Scrum/Lean Startup. AgileSparks is the chosen agile training/consulting vendor of the HP Software R&D center in Israel (Now Micro-focus). Our work in HP included accompanying multiple product groups in their first steps into cloud-based DevOps.
- Nice Systems - Helped multiple enterprise software groups successfully implement Agile - including a group focused on identifying Nice's Big-Data architecture.
- PerfectoMobile - Helped product development with initial move to Agile/Scrum as well as major tune-up to achieve real agility including DevOps, Lean/Kanban overall flow, Test-first. PerfectoMobile provide a "Mobile/Web/IOT Test Lab As a Service" and are world-leaders in this space including projects on Connected Cars and the most popular digital experiences.



- Nice Actimize - Helped both infrastructure/platform groups as well as solution/application groups in this financial tech enterprise software vendor achieve business agility through work with teams, product organization, using Scrum, Kanban, SAFe. Work was done in **Israel, the US and Eastern Europe**.
<https://www.youtube.com/watch?v=b0Lh5nleIkY&t=1s>
- Eagle Investment Technologies - Achieve enterprise solution delivery agility leveraging Scrum and scaled agile concepts and practices. Work done with teams in **the US, Canada, India, Ukraine, Poland, Singapore**.
- CyberArk - Helping this cyber-security vendor scale implement and improve their agility over a period of several years as they scale the organization all the way to a successful IPO. Some aspects of their story can be found here:
https://www.youtube.com/watch?v=Z6i35P1wE_c&t=3s

TELCO and network:

- Amdocs - Help multiple divisions that develop and customize customer experience enterprise software/solutions (e.g. Billing, CRM, Ordering, Network OSS) implement agile/Scrum effectively. Work is delivered in the **US, UK, India, Cyprus, Canada, Ireland, Brazil** and more. This project spanning 10,000 + people started in 2008 and continues until today as we go in/out and help different divisions move to agile, stabilize an agile process and then go back and help them tune it up / boost it to higher levels. We presented this case study at the Lean Systems and Software Conference in Atlanta in April 2010 (see <http://atlanta2010.leanssc.org/2010/02/yuval-yeret-and-erez-katzav-scaling-amdocs-pbg-from-team-scrum-to-a-multi-program-portfolio-using-lean-and-kanban/> <http://www.leanssc.org/files/201004/powerpoint/4.22%2010.45am%20Yeret&Katzav%20ScalingAmdocs.pdf>),
- Viaccess-Orca - working with the development centers in **Israel and France**. Implementing agility and a SAFe-like process build on scrum. Implementing DevOps in both centers.
- Harmonic - Working with the development centers in **Israel and the Ukraine** (~100 people). Training and consulting.
- AT&T Interwise - work with the ~600 people development center in **Israel** to implement and improve agility.
<https://www.slideshare.net/AgileSparks/agile-at-att>
- Cloudyn (acquired by Microsoft) - Implementing agility in the centers in **Israel and the Ukraine**.
- Alcatel-Lucent Cloudband - implementing agility and scrum with teams in **Israel and Belgium**, see their story:
<https://www.slideshare.net/AgileSparks/cloud-band-itai-case-study>
- Cisco (SON division) - Helped multiple product groups implement Scrum in depth and at scale. **Israel and India**.



Medical industry:

- GE - worked with the CT group in **Israel and India** to implement agility and scrum.
- Philips Medical Systems. Work with multiple groups, mainly ICAp and I4, several hundreds of people. Implemented agile, scrum and SAFe. **Israel, the Netherlands and India.**
- McKesson - trained and coached scrum for a group of ~100 people.

Consumer:

- 888 Holdings - Trained and coached scrum in multiple groups and teams (several hundreds of people) in **Israel and Romania.**
- Ebay - worked with the Structured Data group (~100+ people) to improve their agility and scrum processes.
- Sizmek -Israel. Worked with the 200+ group to move to agile. See their case study:
<https://www.slideshare.net/inbaroren/from-6-months-release-to-3-weeks>
- FiftyOne - (Documented in <https://www.slideshare.net/AgileSparks/benny-peer-fiftyone-case-study-agile11-presentation-10apr2011-with-video-version> & Beyond Agile - <https://www.amazon.com/Beyond-Agile-Tales-Continuous-Improvement/dp/0989081214>)
- (<https://www.slideshare.net/AgileSparks/achieving-predictability>)

See full list of testimonials at <https://www.agilesparks.com/testimonials/>



Boston, MA - Yuval Yeret

With over 17 years in technology management both in IT and Product Development, Yuval easily jumps into the world of our clients – whether a small startup, enterprise product development, or an IT unit. Yuval has led some of our key enterprise-level agile implementations and has been central to AgileSparks’s pragmatic best-of-breed solution design, taking the best from each approach, and not sticking to any particular dogma. Over the last couple of years, Yuval has led the inclusion of thinking/models from Kanban, Complexity Thinking, Lean Startup, and more into the AgileSparks method and has created our Kanban/Flow Lean practice and basically kickstarted Kanban for software in Israel.

Yuval is a SAFe Program Consultant Trainer ([SPCT4](#)), a Professional Scrum.org Trainer ([PST](#)) and an internationally recognized Kanban Trainer and thought leader. He’s a recipient of the [Lean Systems Society Brickell Key Award](#) for upcoming world-influencing talent and is a Lean Systems Society Founding Fellow. You can check out his thinking in the popular blog at [YuvalYeret.com](#) or in his book [“Holy Land Kanban”](#). Yuval is also a SAFe contributor - see for example his [guidance article](#) on using invitations to drive healthier SAFe implementations - now included in formal curriculum of the Implementing SAFe 4.5 class. He’s also currently involved in an effort to create a Scrum.org Professional Kanban for Scrum Teams that is designed exactly for teams requiring tune-ups and looking to improve their agile maturity by combining the best practices from both Scrum and Kanban.





SAFe™ Implementation Strategy Leadership Focusing Workshop

SAFe™ Training is a great start but not enough

For Enterprise Software-enabled organizations looking to use the Scaled Agile Framework (SAFe™) to improve their business agility, the AgileSparks SAFe™ Implementation Strategy Management/Leadership Workshop is an interactive deep-dive follow up session for Leading SAFe training aimed at creating a strategy for implementing SAFe in the organization. Going beyond the formal training, this workshop starts with the specific pains, context, goals of the organization and focuses on how to implement the learning and how to drive the change.

This workshop is based on the AgileSparks Management/Executive Workshop used in dozens of enterprises to kickoff successful agile transformations - a key part of the AgileSparks Way. It combines field-proven best practices, applied in an interactive and expertly-facilitated manner, to provide the organization's management with a context-specific road map to improvement.

When to use this workshop

The workshop is suitable for organizations that can be in different stages of their Agile journey - those new to Agile, trying to scale it or trying to go from practices or even "agile in name only" to real understanding and application of Lean/Agile at the organizational level. It is typically bundled with Leading SAFe training for the same group or follows up an earlier Leading SAFe public/inhouse training.



Key Objectives/Deliverables

- Establish Need/Goals for Enterprise Agility through exploration of pains/opportunities in current way of doing things
- Establish what SAFe means applied to the organization -
 - What will be our Portfolios? Value Streams? Trains?
 - Who will fill which roles?
 - Do we need 3-level or 4-level SAFe?
 - What will be our Epics? Capabilities? Features? Stories?
 - What should be our Release Cadence?
 - How do we fit SAFe into our current governance context? The bigger organization? Partners? Customers? (Dealing with waterfall/phase gate/CMMI/etc.)
- Review and learn from applicable case studies for SAFe / Scaled Agile.
- Explore complementary practices required to successfully implement SAFe in the specific organizational context, e.g.:
 - Technical Safety
 - Sustainable Pace
 - Lean/Agile Requirements in a world of uncertainty - Minimum Viable Products (MVPs), Minimum Marketable Features (MMFs), Requirement slicing techniques, Lean Startup thinking.
 - Combining Scrum and Kanban into ScrumBan
 - AgileSparks Way Approach to Change Management
 - Agile Testing Approaches
- Is SAFe SAFE here? Create and review a draft blueprint, explore and deal with risks until a high-confidence plan surfaces.



Workshop Approach

The workshop is highly interactive. We insist on a small intimate forum where everyone participates and affects the outcome. Innovation Games, Gamestorming and other techniques are used, and as an important side effect the participants also learn how to effectively think and collaborate to solve tough system-level problems in the future.

As the purpose of the workshop is to outline high-level improvement strategy together, the bulk of the thinking is done together in real-time, with us helping you apply relevant thinking approaches and tools. In order to have an effective workshop, we do spend some time getting acquainted with your context and expectations, using a few focused assessment interviews.

In true agile fashion, we aim to deliver value rather than follow a predefined plan/syllabus. While a backlog is prepared based on initial preparation/interviews to learn the general context and aims, we continuously seek feedback and adjust direction driving at maximum impact and return on the time of participants based on the discussions and directions that emerge.

Key agile tools such as Kanban Boards, Burnups, Estimation/Prioritization games and frequent Retrospectives are used throughout the Workshop, leading to hands on learning of how Agile works.

Actually, We recommend these techniques be adopted for the ongoing routine of Management Teams

Duration

1-2 days of interviews and preparation.

Main Workshop is typically 2-3 days, depending on the organization size and complexity, preferably held in an offsite location.

Delivery

The workshop is facilitated by one of our SPC/SPCT-certified Enterprise Agility Lead Consultants. In some contexts we recommend 2 consultants facilitate the workshop in tandem.



About AgileSparks

We are a boutique consulting firm with offices in the US (Boston), Israel and India. We help product development and IT organizations improve their outcomes, productivity, predictability, quality, employee engagement through a variety of Agile/Lean solutions like Scrum, Scaled Agile Framework, Kanban, Lean Startup.

We are a small team of experts in Agile/Lean, including internationally recognized thought leaders, book authors and accredited/certified trainers with leading bodies in the Lean/Agile. We are Scaled Agile Gold Partners.



Proven Enterprise-level Experience

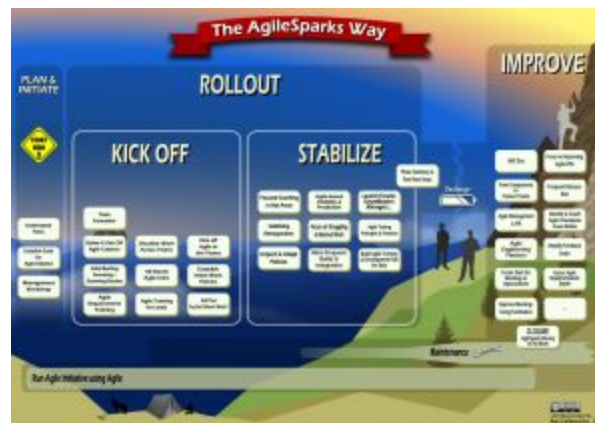
AgileSparks' experts have proven experience in Agile/Lean training, consulting and implementation in over 100 companies world-wide, ranging from startups all the way to Fortune50 companies in a variety of industry segments - ISVs (product development), enterprise IT and governmental organizations, TELCO, embedded SW and Web/SaaS products, Defense contractors and custom project development shops. Past references include companies such as: Siemens, HP Software R&D, Intel, Cisco, EMC, Informatica, Amdocs, GE, CA, Philips, Retalix-NCR, Nice, Cisco, Motorola, CyberArk, eBay. See our site for testimonials from those clients and others.





Sparking REAL Agility, not Agile Theater

In these days, everybody's going agile or is already "agile". We are seeing a lot of "Agile Theater" out there - people implementing surface-level practices without deep changes/spirit. With extensive experience in different sizes and contexts, and a team of world-class experts that continues to challenge itself to develop and improve best practices and building blocks, **AgileSparks** is uniquely positioned to help you choose wisely the way to REALLY improve, as well as be your trusted partner along the way. We have the experience and the know-how and, not less important, the passion to help your organization improve. It might be a tougher journey requiring deeper changes, but the outcome will be an actual improvement in your business agility not just the ability to say you are "doing agile". We call this **"The AgileSparks Way"**



We continuously challenge the comfort zone of our customers and, by means of this, we improve their efficiency, productivity and ability to deliver high quality products, while keeping the ability to be flexible (agile) for their customers.



Appendix - Workshop Agenda Sample (Will be customized per context)

Opening (30min)

- Intro+Purpose
- Working Agreements
- Expectations Setting/Collection

Why we're here (1-2h)

- Identify current state
- Pains session - choose what to improve

What SAFe means here (3-5h)

- What will be our Portfolios? Value Streams? Trains?
- Who will fill which roles?
- Do we need 3-level or 4-level SAFe?
- What will be our Epics? Capabilities? Features? Stories?
- What should be our Release Cadence?
- How do we fit SAFe into our current governance context? The bigger organization? Partners? Customers? (Dealing with waterfall/phase gate/CMMI/etc.)

Review and learn from comparable case studies for SAFe / Scaled Agile

- AgileSparks Case Studies
- Industry Case Studies (e.g. Spotify, Lego, other SAFe case studies)



Follow up Context-specific Education/Learning (3-5h including exercises)

- Human aspects of Lean/Agile - More techniques for Decentralized Control, Technical Safety, Sustainable pace
- Lean/Agile Requirements in a world of uncertainty - Minimum Viable Products (MVPs), Minimum Marketable Features (MMFs), Requirement slicing techniques, Lean Startup thinking.
- Combining Scrum and Kanban into ScrumBan
- Intro to Agile Testing
- AS Way / Change Approach

Is SAFe SAFE here? (2-3h)

- Create and review a draft blueprint
- Confidence Vote
- Dealing with Risks/confidence gaps

Closing (30min-1h)

- Takeaways, Epiphanies, Feedback, commitment to next steps