

Empowering Agile Teams to Achieve a Continuously Learning Organization

An ABC Guide to Leveraging Adaptive, Bite-Sized, Composable Practices



The essence of agile is being adaptive.

Here's a few reminders of what that really means:

Adapt: "Make (something) suitable for a new use or purpose; modify"¹.

Adaptive behavior: behavior used to adjust to another type of behavior or situation.²

To be agile as teams, we need to adjust our approach to meet our immediate challenges and needs. To be agile as an organization, we need to learn collectively and evolve our approach over time to support our evolving mission, so that we continue to excel in an ever-changing environment.

We would not call a TV set "adaptive" if, in order to adjust the volume, we had to throw it away and replace it with a model with a different volume setting. So why are we prepared to accept process frameworks that leave us in a similar predicament every time we want to improve our product development performance as an organization?

To be clear what kind of difference being genuinely "adaptive" actually makes, try envisaging an ideal future for your organization - say in 10 year's time. Which of these options would you wish for:

Case A

We still have our old waterfall process, which is probably theoretically how at least some things still need to be done around here. But no one has gone near the process website in years. First, teams started "going agile", but by the time we'd hired in a bunch of people who said they knew what that meant, and how to make it work for us, the world moved on, and now "Nimble" is the new "big thing". So now we need to find some new people who know what that really means ...

Case B

We've just finished rolling out our new "STRONG™" process framework as part of our "Nimble Transformation". There has been a lot of pain along the way – lots of people are still wedded to our old Agile Process, and can't see why we had to rip it out and change everything yet again.

Case C

We continue to evolve and adapt our approach – now we have a lot of new "Nimble" practices being used, alongside some of our more mature lean and agile ones, and others that have been with us even longer, like Use Cases and Data Models, that continue to prove their worth in specific areas. The important thing for us as an organization is that our library of practice options continues to grow and change organically as we grow, and as the challenges we face continue to evolve.

Please be clear, we are not really selling my imaginary "Nimble" approach as the next "big idea", nor the entirely fictional "STRONG™" process framework as "the next big thing" in process frameworks.

The real point is that we all know that:

- Different practices are good for different things;
- No one process framework will provide for the needs of all our different projects and teams;
- New practices emerge over time to meet new challenges and circumstances;
- Some practices have been useful for a long time, and some will likely remain so;
- Industry fads and fashions cause still-good practices to be thrown out or forgotten about;
- Today's processes are not the be-all-and-end-all, and will be superseded by others in future;
- Ripping and replacing "the one big, all-encompassing process" is painful and wasteful;
- If this is our strategy, history suggests we won't have suffered this pain for the last time.

This series of articles provides a step-by-step guide to a better way to achieve the goal of agile teams within a learning organization through the use of adaptive, bite-sized, composable practices.

¹ <https://en.oxforddictionaries.com/definition/adapt>

² https://en.wikipedia.org/wiki/Adaptive_behavior



1. What is a Bite-Sized Practice?

A bite-sized practice is anything that helps a team or an organization to do something well or better.

Here's an example – an Agile Retrospective Essentials practice that helps a team to “Make incremental improvements to their way of working through regular, repeated retrospectives”.

Now you may say, “why redefine agile retrospectives, when they already exist as Sprint Retrospectives as part of Scrum?”

But the whole point here is that agile retrospectives are useful in many contexts, and therefore should not be thought of as an inextricable part of a whole called Scrum.


This kind of thinking is not uncommon, but it is very damaging. It means that organizations assume that doing retrospectives is part and parcel of adopting a bigger process framework called Scrum. And this in turn means ripping out and replacing all other practices that operate in the same space. They assume doing retrospectives must also mean planning in timeboxes (and calling these Sprints). So, before you know it, all teams are being pressured into doing all these things, whether or not it is right for them. Agile now starts to be equated with Scrum in everyone's minds. People therefore assume that those wishing to explore other options, such as Kanban-style, “continuous flow” models, must be advocating ripping and replacing of Scrum and all the practices that it contains, including the abandonment of team retrospectives. **NONE OF WHICH IS EVEN REMOTELY TRUE.**

The whole idea of bite-sized practices is that they avoid this whole false and dangerous line of thinking, and instead actually enable what agile should really be all about, which is continuous improvement through the adoption and refinement of practices, based on reflection on existing and emerging challenges, in search of ever-improving customer responsiveness and value delivery.

So, let's have a look at what a bite-sized and independently adoptable Agile Retrospectives Essentials practice actually looks like.

In this case, it happens to be just about the smallest, “lightest bite” of a bite-sized practice that we could imagine, containing just three constituent elements, or “cards”:



1. A “Mad, Glad, Sad” Pattern that gives us advice on (one way) to go about running a retrospective.



**Mad, Sad, Glad**

A popular approach to team brainstorming to identify potential improvements.
Team members write on sticky notes what has made them:


- *Mad* – frustrations
- *Sad* – disappointments
- *Glad* – things that went well

Part of its power is that it taps into people's emotions, and results in an unfettered flow of ideas that the team can then analyze, prioritize and action.

One Approach To:  Hold a Retrospective
Ref:  Mad, Sad, Glad


 2017.05



2. The activity we do as a team to achieve the results (note that nothing ever actually gets done unless there is an activity of some kind to do it!).

**Hold a Retrospective**

The whole team meets regularly to reflect on its way of working. Improvements are identified and prioritized, and actions agreed. At the next retrospective, the results are evaluated.



☐ Improvement

 **Support the Team**


 Leadership  Management

☐ Way of Working: Working Well (contributes to)


☐ Improvement: Action Agreed or beyond

 2017.05



3. What it is that we are looking to achieve or progress by performing this activity (known as an Alpha).

**Improvement**

A possible adaptation to improve a Team's Way of Working.







Relates to: ☐ Way of Working



 2017.05

Agile Retrospective Essentials

Make incremental improvements to the way of working through regular, repeated retrospectives.

 Mad, Sad, Glad  Hold a Retrospective

 Improvement  Resources

 2017.05



Note that this way of defining a practice is:

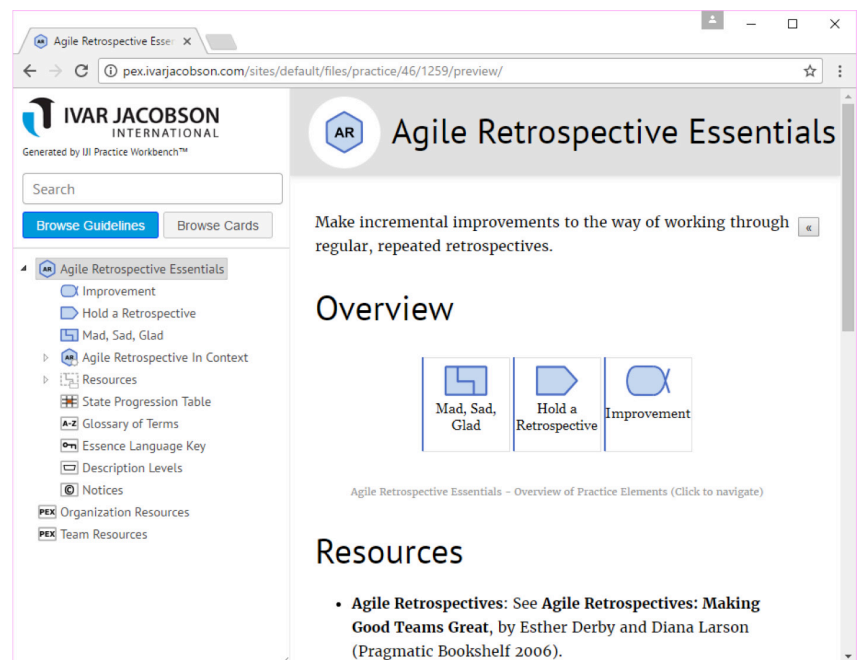
Agile – because it encourages us at all times to keep things super-simple and lightweight
Lean - because it focuses us on the outcomes that we are looking to achieve.

But, most importantly of all, it gives teams and organizations a picking list of practices that they can adopt to address or to improve specific aspects of their delivery capability, without having to change anything else at the same time if they don't want to.

Another key to the success of any working practice is its visibility and accessibility. If it is even slightly taxing to find or follow an agreed working practice, it will not be used, followed or understood.

We have seen above how the practice can be made available as a small number of “poker cards” for the team to use to guide them, and communicate to others how they are working. As we will see later, these cards can also be used in various games to help agree, sanity-check and adjust the team's way of working.

But we also really need the same practices to be available as browsable HTML, so anyone can check them out at any time.

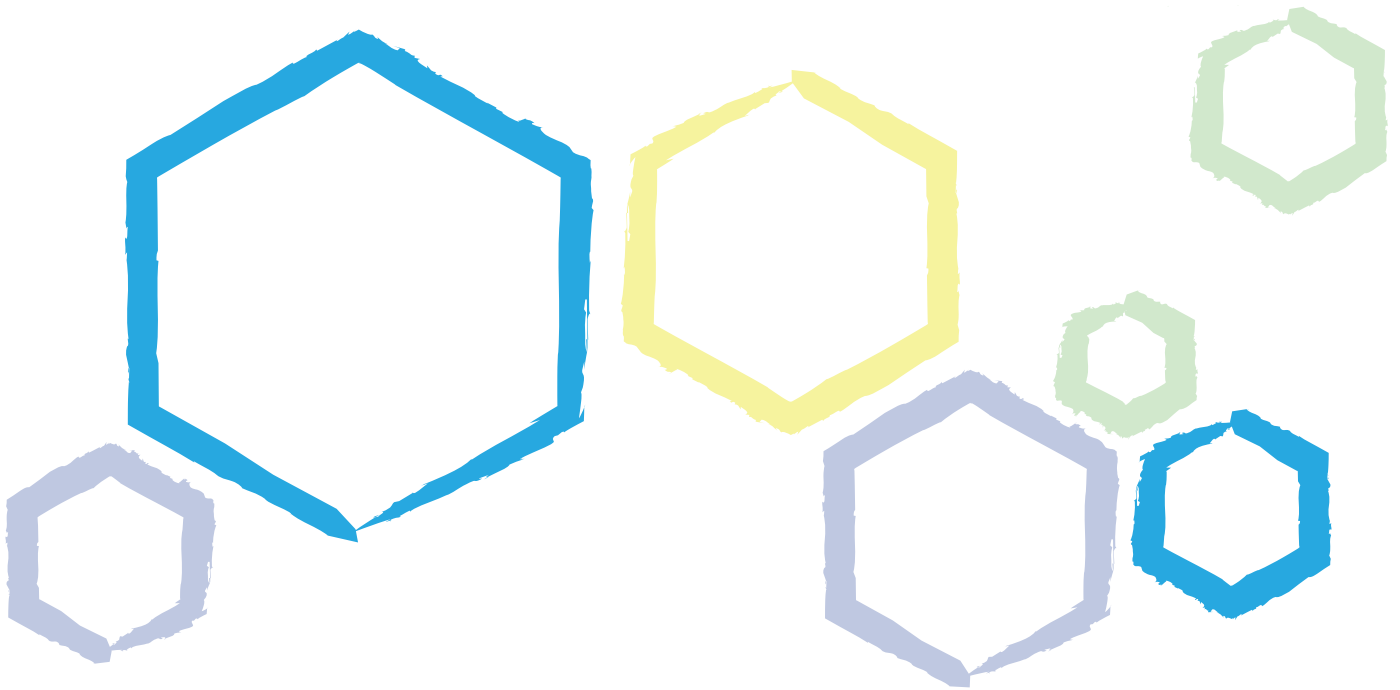


And why not also as browsable “eCards” on everyone's smartphones, for instant reference anytime, any place?

The most important thing though is that, even if “Scrum” gives way to “Kanban”, or to “SAFe”, which in turn gives way to the “next big thing”, as long as retrospectives are useful, they can still be used. And we don't have to rip and replace, or change or “relearn” how we are supposed to do them, until and unless we (as a team or as an organization) actually want to.

In the next instalment we will look at how these bite-sized practices can be bolted together to form cohesive methods and ways of working that can enable teams to cover off everything they need to do successfully to tackle even the most complex, risky and large-scale endeavors.

Part 2: What is a Composable Method?



Empowering Agile Teams within a Learning Organization

An ABC Guide to Leveraging Adaptive,
Bite-Sized, Composable Practices

1. What is a Composable Method?

In the first article we introduced the concept of a “bite-sized practice”, and we looked at some of the advantages of having practices in this “bite-sized” form, as opposed to being irretrievably embedded in a big, all-encompassing process framework.

We defined a “bite-sized” practice as “as anything that helps a team to do something well or better”, and we looked at the example of a super-compact “Agile Retrospective Essentials” practice that was succinctly described on a handful of “poker cards”, and also accessible as “eCards”, that anyone and everyone can browse on their smartphones, and also as browsable HTML guidelines.

The unfortunate truth, however, is that teams need to be able to do more than just one thing well if they are to excel at solving complex problems by using complex technologies to develop complex products, such as large-scale software development projects or programs.

The last thing we want is for our practices to add to the complexity, but we do need them to help us to tame complexity, and to help us to rise to the myriad challenges that we will face.

A composable method, very simply, is the set of practices that team members have chosen to use to help them be successful, bolted together to form a seamless, cohesive whole, that encapsulates “how the team works” – also known as the team’s “Way-of-Working” (for obvious reasons).

As with any such composable approach, for it to work smoothly, we need some enabling technology working for us “under the covers”, in the shape of:

- A “composable framework” to enable the practices to work together when bolted together
- Sufficient standards to ensure that the resulting way-of-working forms a single, seamless whole, and not an ill-fitting bag of spare parts.

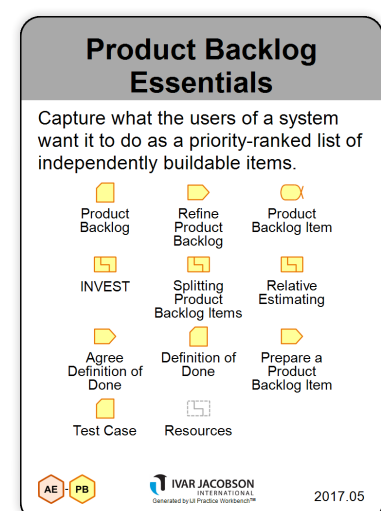
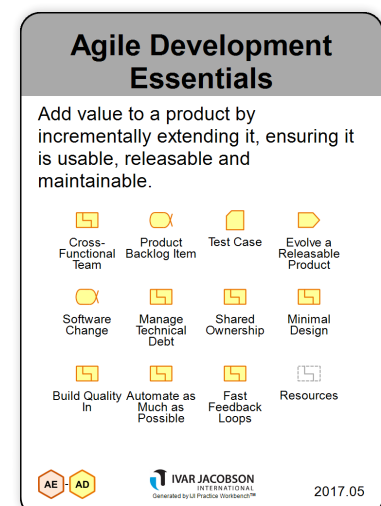
The practices and tools that we will be exploring in this series of articles all use the OMG Essence standard, which provides both of these things – through its common standard language for describing practices and methods, and its common framework, that it calls a “kernel”.

Let’s take a simple example. Let’s say we decide we as a team want to use a (Scrum-like) backlog-driven approach to prioritizing and sequencing our work, combined with a set of (XP-like) technical development practices to help us evolve a releasable-quality software product, as shown in the above.

Each of these bite-sized practices could be used separately, or in combination with other different practices, but let’s say that this is our team’s chosen combination, at least for the time-being.

But how will this combination of practices work together when we try to use them together?

To understand this, we can look at what each practice does (and doesn’t) help us to achieve, and then we can also look at what the combination of the two can collectively help us to achieve.





The diagram below is an example of a progression matrix that can be automatically generated from the information contained on the poker cards within each practice.

This is actually a subset of the matrix for the Product Backlog Essentials practice. It tells us that the activities within this practice will help the team to:

- Progress the Requirements to a “Coherent” state
- Progress Product Backlog Items to be “Ready for Development”
- Capture Test Ideas, and Script and Automate the Test Cases.
- But also it tells us that this practice does NOT, for example, help the team to progress:
- The Requirements to “Addressed”
- Product Backlog Items to “Done”.

If we want explicit guidance on how to do this, we will need to look to adopt another practice and use it in combination with this one.

By looking at the same matrix for the Agile Development Essentials we can easily identify that this is such a practice. And by choosing to compose these two practices together, we can now get a seamlessly integrated way of working that enables the team to progress more things further.






This is a subset of the same matrix view, but this time as generated from the “Backlog-Driven Development” method (as the team has chosen to call it) that was created by choosing to compose these two practices together. We can see that these practices, when used in combination as a composed method, enable us to:






- Seamlessly progress Product Backlog Items from being “Identified”, through being “Ready for Development”, all the way to being “Done”
- Also progress the Software System being developed through to being Demonstrable, Usable, Ready, Operational and even Retired.

But, we can also see that we have no explicit guidance currently as to how we move the Software System through to its initial Architecture Selected state.

Now, it could be that we as a team are not responsible for doing this, or it could be that it is such a trivial thing to achieve in our case that we do not need any explicit guidance to help us. Or it could be that we want to look for a practice to help us with this.

And, no surprises, there is an Agile Architecture Essentials practice available for the team to adopt if required, to compose in with these other practices to form expanded-scope way-of-working that helps them as a team to progress the software architecture in a controlled and agile way.

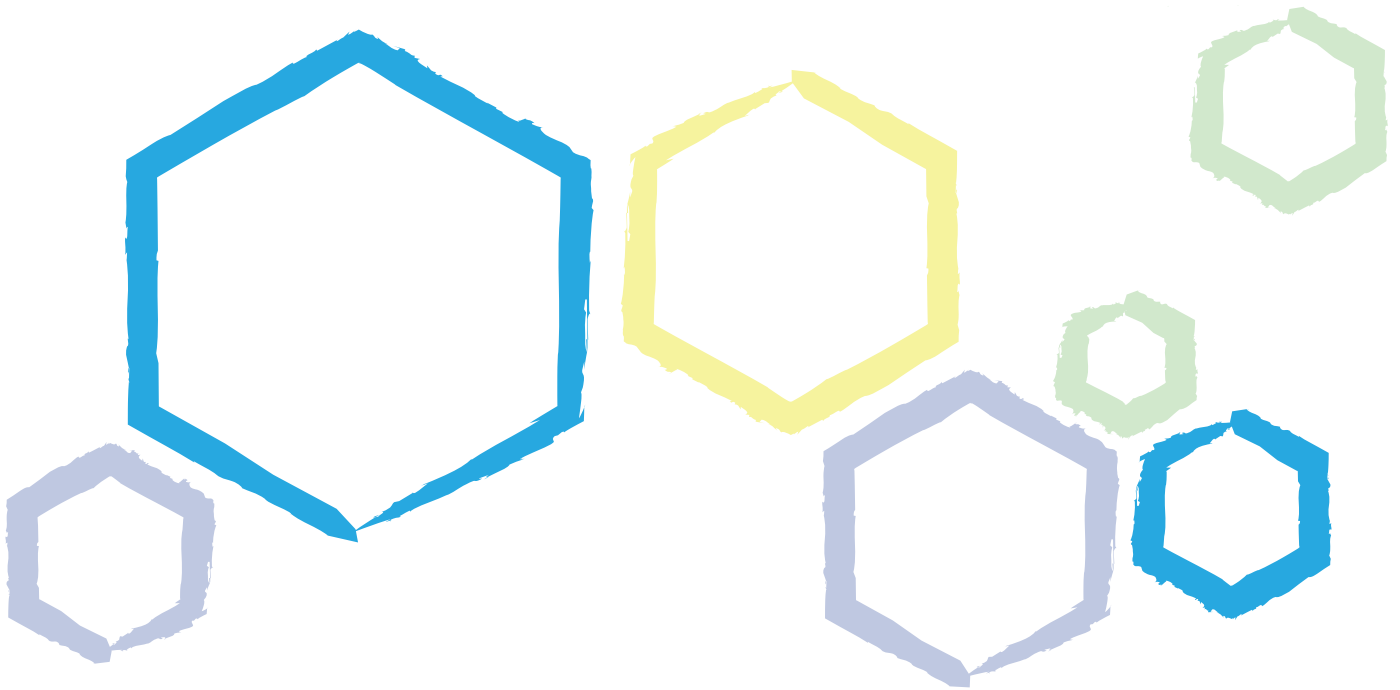
	 Refine Product Backlog	 Prepare a Product Backlog Item
 Requirements		
Conceived		
Bounded		
Coherent		
Acceptable		
Addressed		
Fulfilled		
 Test Case		
Test Ideas Captured		
Scripted		
Automated		
 Product Backlog		
Identified		
Ready for Development		
Done		

	 Refine Product Backlog	 Prepare a Product Backlog Item	 Evolve a Releasable Product
 Product Backlog Item			
Identified			
Ready for Development			
Done			
 Software System			
Architecture Selected			
Demonstrable			
Usable			
Ready			
Operational			
Retired			



In summary, we have seen how a practiced-based approach gives us an ideal combination of flexibility and control: the flexibility to choose the practices that we think we need, and control because we can easily judge whether the practices we have selected do actually give us exactly what we need, and everything that we need. But we can also see that this level of tailoring is quite course grained – we either choose to use a practice exactly as we find it, or we choose not to. In the next article we will look at how we can fine tune the tailoring, by adapting the practices we have selected to use for an even better fit.

Part 3: Adopting and Adapting a Bite-Sized Practice



Empowering Agile Teams within a Learning Organization

An ABC Guide to Leveraging Adaptive,
Bite-Sized, Composable Practices



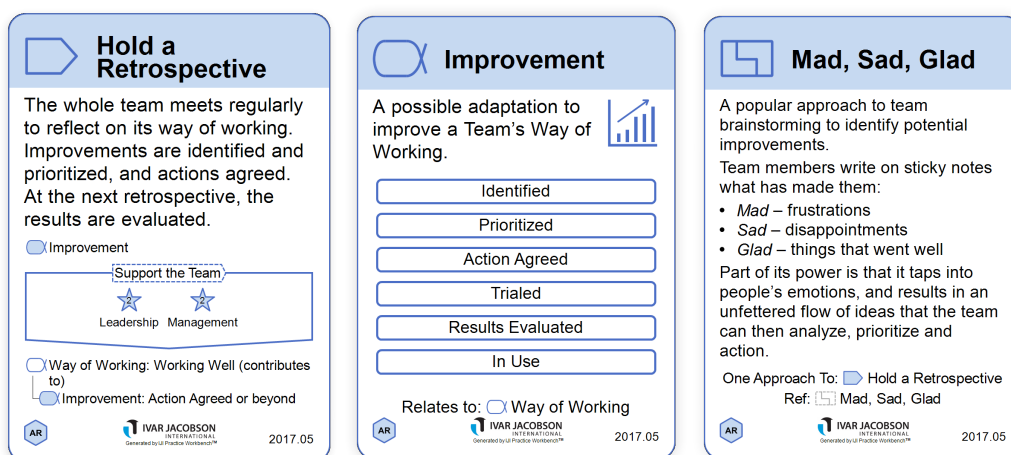
When we are working as an agile team, we want to rapidly agree a great way of working. It has to be right for us and our customers, and indeed all our stakeholders. And we want it to be crystal clear and visible to all. But we don't want to spend too long on doing this – we would much rather spend our time adding value to a product by developing great software.

One strategy for agreeing our approach is to “start from a completely clean sheet of paper” each time, and argue out how we want to work as a team starting from first principles (or just based on past prejudices), as if we were the first agile project ever to emerge blinking into the sunlight. But this has several downsides:

- It takes time too much time – often involving long-running disputes and battles of attrition
- It confuses and dumfounds our stakeholders – they are thinking “surely they can't have to make up how to do this software development thing on the fly every single time they do it?”

Part of the joy of bite-sized practices is that they can be used to rapidly agree a tailor-made way-of-working. The team simply looks at an available practice and decides whether or not to adopt it, and whether or not to adapt it.

So, let's say we decide to grab the Agile Retrospective Essentials practice we saw in the first article and start to use this as a team:



Perhaps at first all goes well, but maybe after a couple of retrospectives we start to surface some concerns that maybe we are not doing such a great job of retrospecting after all, for example with brainstorming sticky-notes coming from the team such as:

- The same things seem to come up again and again in our retrospectives
- We don't actually follow through and fix the problems we surface and debate
- Are retrospectives becoming just a communal moaning session?

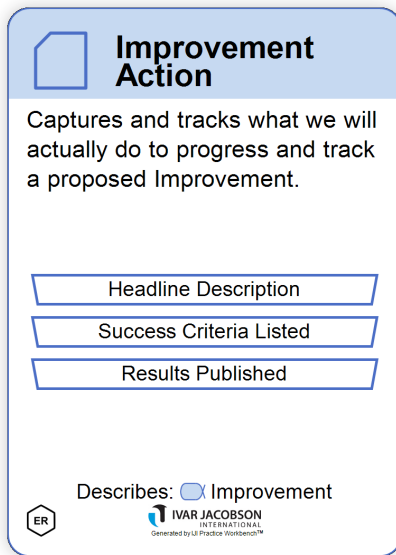
Maybe analyzing these challenges, we come up with the following change ideas as a team:

- Capture agreed improvement actions and make them visible on our board
- Plan the implementation and evaluation of improvement actions into our Sprints
- Start each retrospective by reviewing the state of previously actioned Improvements.



Rather than just add these to our list of Improvements, and then do nothing with them(!), perhaps we decide to agree what we will actually do about them by sketching out some new proposed working practices to try, starting now!

First of all, we can ask ourselves how are we going to make agreed improvements visible. Maybe we decide that these should be blue sticky notes that we stick to the left of our Kanban board. We can quickly capture and communicate this idea as a new Work Product card:



Things to note include:

- Using the Practice Workbench tool it is easy to capture a neat and concise description of exactly what we decide we need
- As part of this we are “reusing” and “extending” what we do already – in this case the concept of an Improvement is already part of the Agile Retrospective Essentials practice that we adopted – and so we show that the new Improvement Action Work Product is simply a way of making visible this existing practice element
- We are encouraged to think about what levels of detail we want to capture – in this case we have decided that we want to list the success criteria up front and also make sure we have some visibility of results.

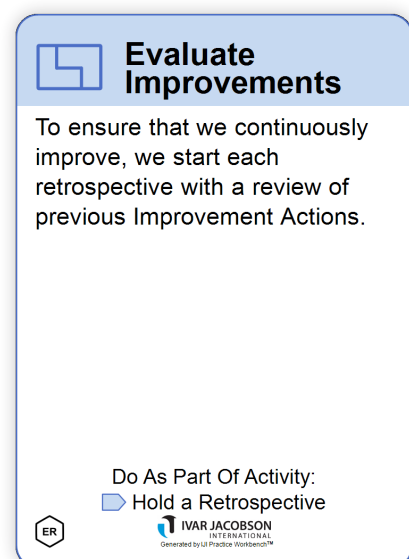
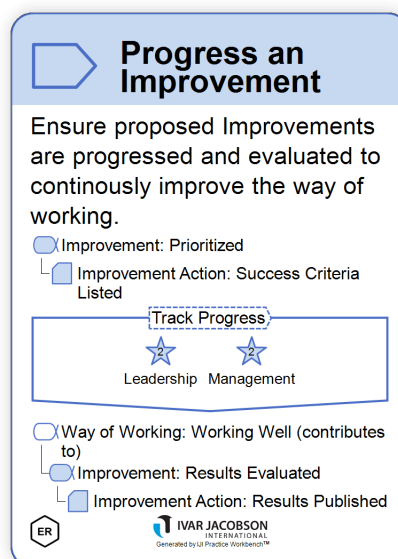
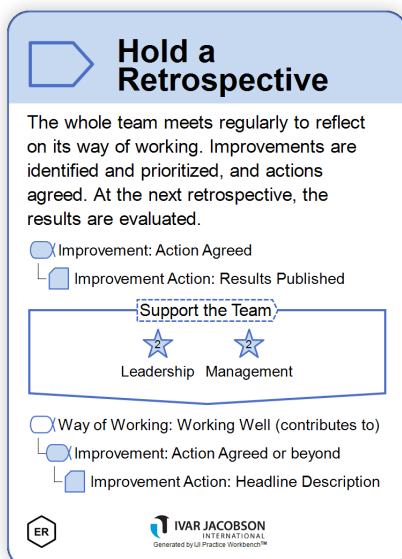
Next we need to decide how and when we will capture these details. Maybe we decide that we will capture the headline and success criteria as part of the Retrospective Meeting itself. So now we can again “Reuse and extend this” and annotate this new aspect of the activity.

We might also capture some additional things we agree about how we will do this as a pattern.

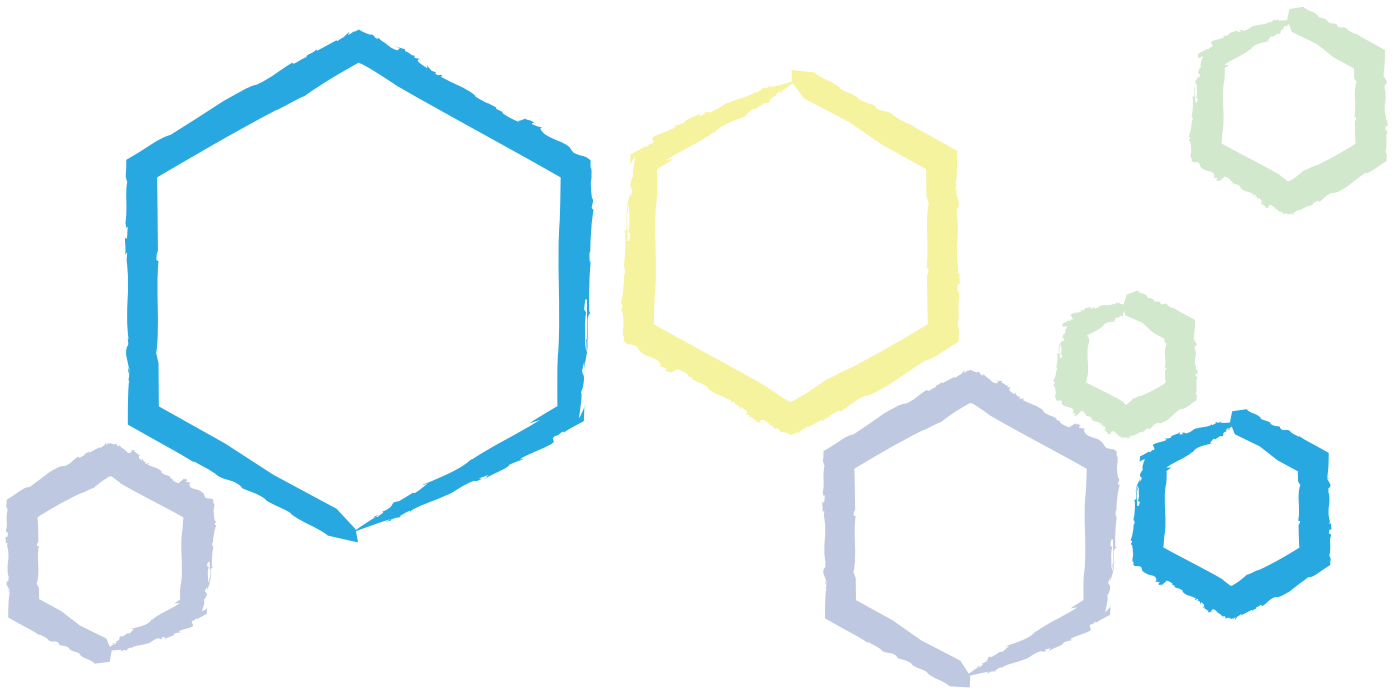
And finally we may decide we need to explicitly recognize the activity needed to progress and evaluate the improvement so that this can be planned in amongst our other work as a team.

And so we are done defining our enhanced practice. The best thing of all is that we can now make this visible for all to see and available for other teams to select and adopt.

The result: empowered agile teams, innovating and improving and making their way of working visible, and a learning organization, where teams can share good practices for others to try.



Part 4: Building a New Bite-Sized Practice



Empowering Agile Teams within a Learning Organization

An ABC Guide to Leveraging Adaptive,
Bite-Sized, Composable Practices



In previous parts we looked at how Essence can help us to adopt and adapt bite-sized agile practices to improve our team way of working. We will now look at the case where we need to create a new practice to fill a gap in our existing practices.

As part of an agile team, we pride ourselves on being responsive and adaptive. We are using a set of popular, tried and trusted agile practices to help us be agile. A key part of this is working in short Sprints, and at the end of every Sprint demonstrating a new working version of the product to the stakeholders.

But feedback has started to emerge in recent retrospectives from various sources, most of it from outside of the team and raised in the retrospectives by the Product Owner, that we as a team simply don't seem to be listening to and learning from the product users and other stakeholders.

At first we are a little distressed, maybe even momentarily inclined to be defensive, because this strikes at the heart of our belief that we are a good agile team.

But, on reflection, we realize that actually we do not do such a great job of capturing and responding to the feedback that we claim to be actively seeking.

So, we ask ourselves: What we are going to do about it?

Because this question rightly involves questions about "doing", it implies new ceremonies that the team should explicitly recognize, and focus on doing well. It feels like we need to adopt a new "bite-sized" practice. But maybe in this case there does not seem to be any pre-defined practice that can help us with this. So we can simply create our own new "Responding to Feedback" practice.

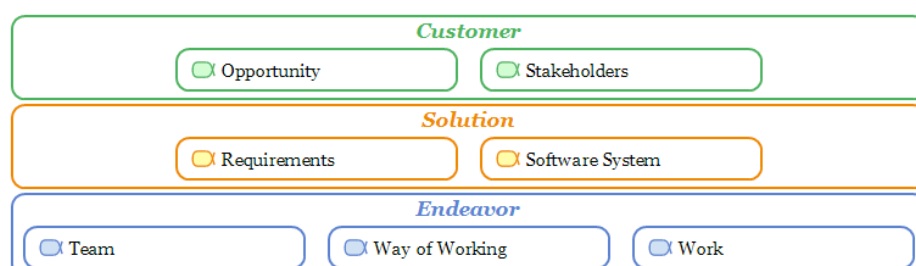
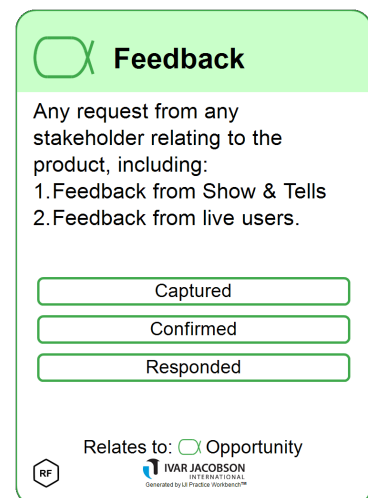
We have already seen that a bite-sized practice typically involves some concrete statements about:

- What are we actually going to do? (Activities) – because nothing gets done unless we do it!
- What is the purpose of the activities – what do they achieve or progress (Alphas)?
- Do we need to capture anything to make this visible and manageable (Work Products)?
- How should we go about doing, progressing, producing these things as a team (Patterns)?

In this case, because this is all about Feedback, maybe we decide first of all to get a little bit clearer on what we mean by this, and what it would mean to respond to it. So we can create a new Alpha card (an Alpha being anything of importance that we feel we need to progress, track, trace, manage etc.)

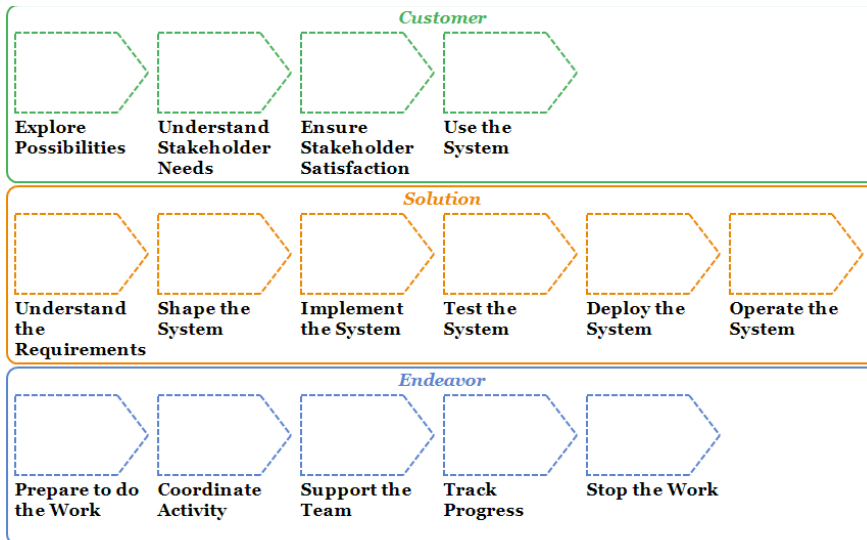
Remember we are using the OMG Essence standard to help us clearly articulate our thoughts.

Part of the Essence standard is a number of predefined "kernel alphas" that show us the key things that pretty much all software development projects are concerned with. Often any new Alphas we decide to create will relate to these, and deciding which one helps us to agree what kind of thing we are dealing with here. In this case, we decided that the Feedback we are talking about here is feedback from our customers, users and stakeholders, about what they want to do or achieve with the product. This tells us that it relates to the Opportunity alpha, as opposed to feedback about the Requirements for the Software System we are building (which we capture as User Stories) or about our Way of Working (which we capture as Improvements).





The Essence standard color-codes these different areas of concern and the things of concern within them, which provides a useful way to ensure that we are all clear what kind of thing we are talking about at any particular time - here we are clearly operating in the Customer space, and the green color-coding helps to make this clear at all times.



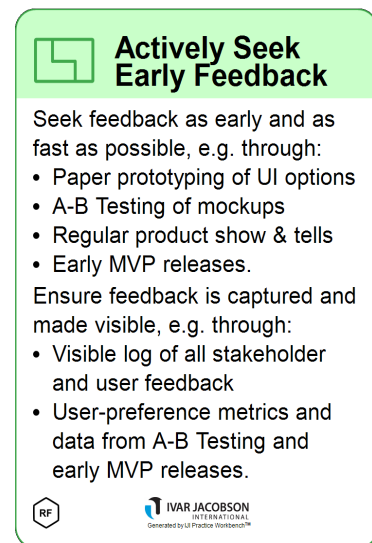
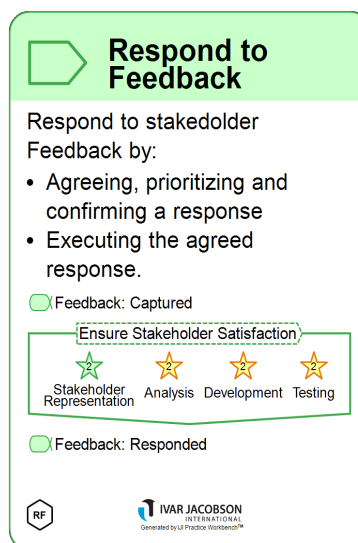
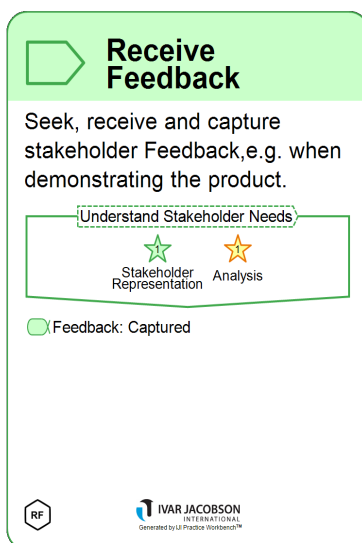
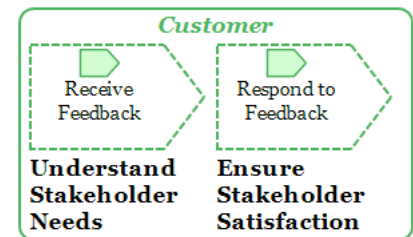
Now we know what we are dealing with and how we need to be able to progress it over time, we can start to think about what kind of activities we need to do to ensure that we successfully manage its progression.

Again, the Essence Kernel can help us think this through by providing a view of all the possible activity spaces that software development typically involves.

On this basis, perhaps the team decides that there are a couple of different things we need to do in different spaces at different times:

1. To ensure we receive and capture the feedback in the first place
2. To ensure that we respond to it appropriately.

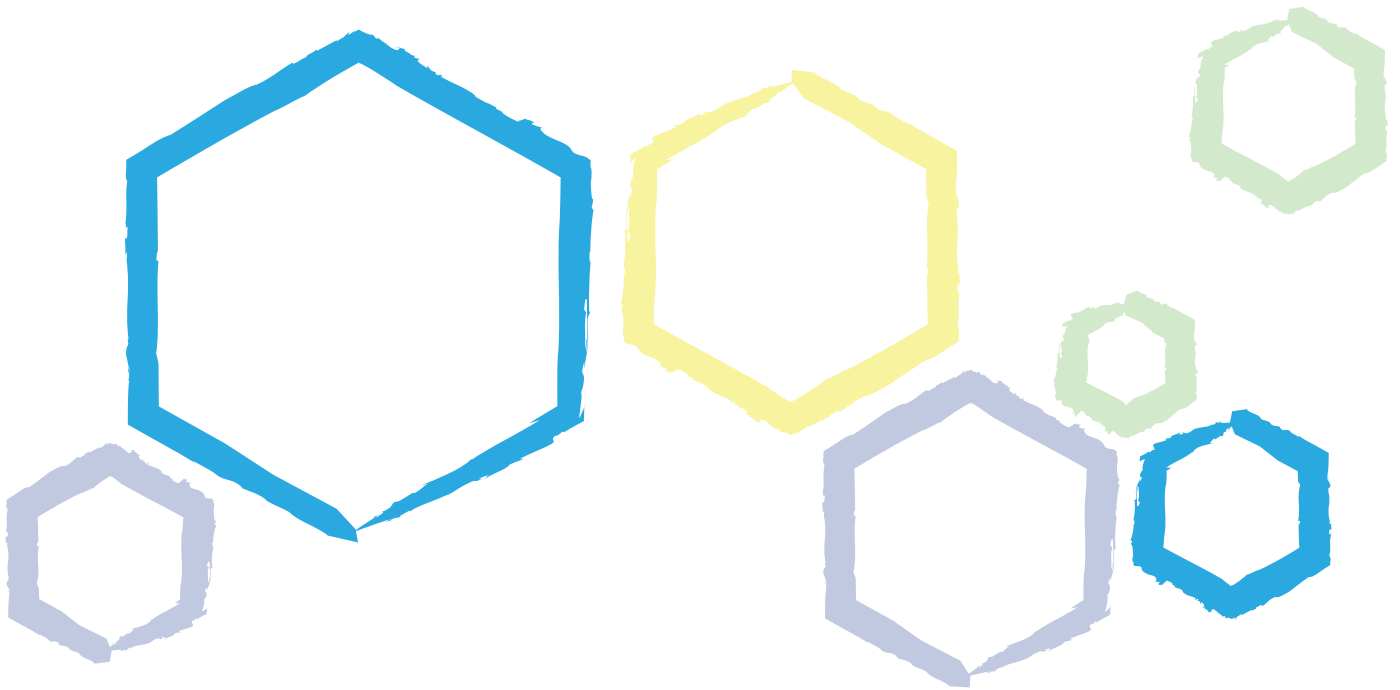
And finally we need to better understand the activities by defining exactly how they progress the Alphas and the Work Products, perhaps as follows:



And, once again, if this proves to be a worthy and valuable practice, we can share it with other teams to ensure that as an organization we continue to learn and improve our practices and products.

Up Next: 5. Composing a New Composable Method

Part 5: Composing a New Composable Method



Empowering Agile Teams within a Learning Organization

An ABC Guide to Leveraging Adaptive,
Bite-Sized, Composable Practices



Previously we have looked at how Essence bite-sized agile practices can help us adapt and innovate our way of working as an agile team. Here we will look at how we can combine these bite-sized practices into a cohesive team way-of-working and ensure that it covers all the key bases.

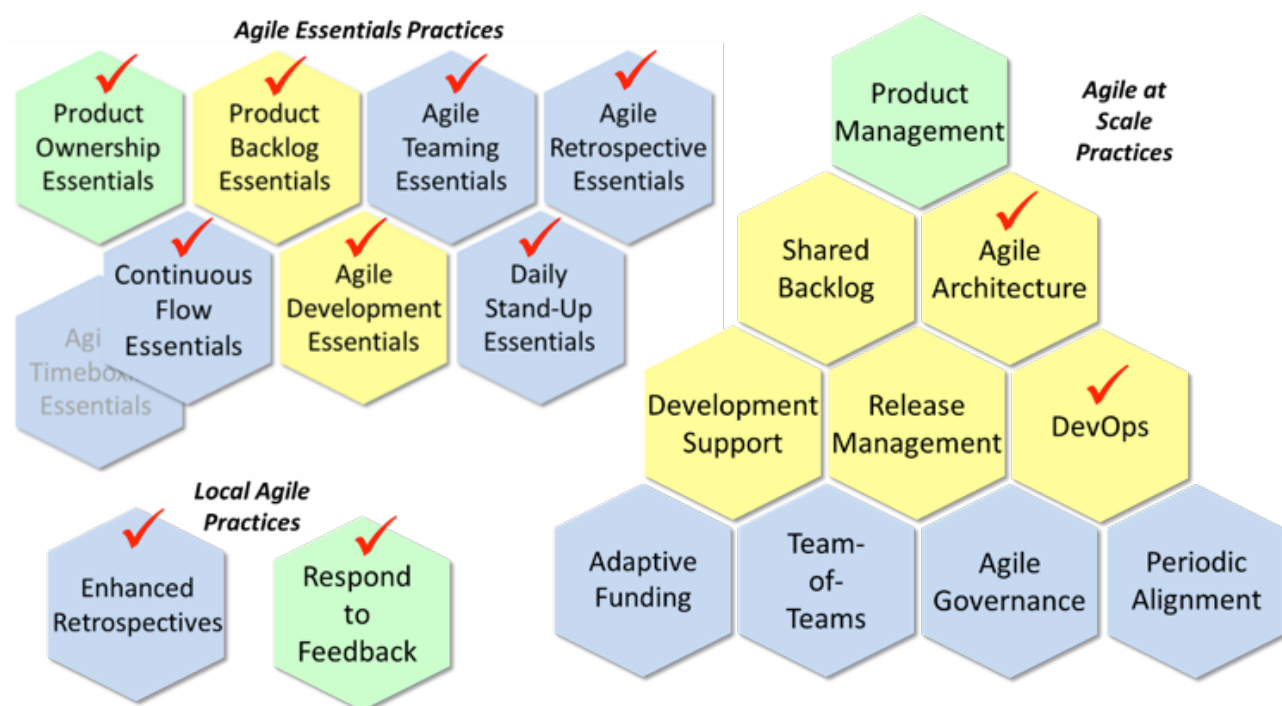
Let's say we are a newly formed agile team. Our mission is:

- Rapidly and significantly enhance an existing product
 - The product is high value, but with a small, close-knit and internal user-base
 - We want to maintain a continuous flow of new value to the users
 - Investments in continuous delivery technologies mean that new releases can be made frequently, smoothly, safely and cheaply
- User experience is key, and we need to leverage user feedback to maximize usability
- We also need to advance the underlying architecture over time to ensure that this can continue to support value extensions and enhancements sustainably into the future.

Based on experience and guidance from our experienced Agile Facilitator (or Agile Coach or Scrum Master depending on your sensibilities and preferred terminology), we decide that:

- We need to leverage the best agile team practices available to us
- Because of the nature of the work as a flow of "rapid small enhancements", we want to use a kanban-based "continuous flow" work model, not a "Scrum-like", timeboxed model
- We need to focus on seeking out and responding to feedback
- We also need an agile approach to enhancing the underlying architecture.

Based on this, we make the following practice selection choices from the practices available to us:¹



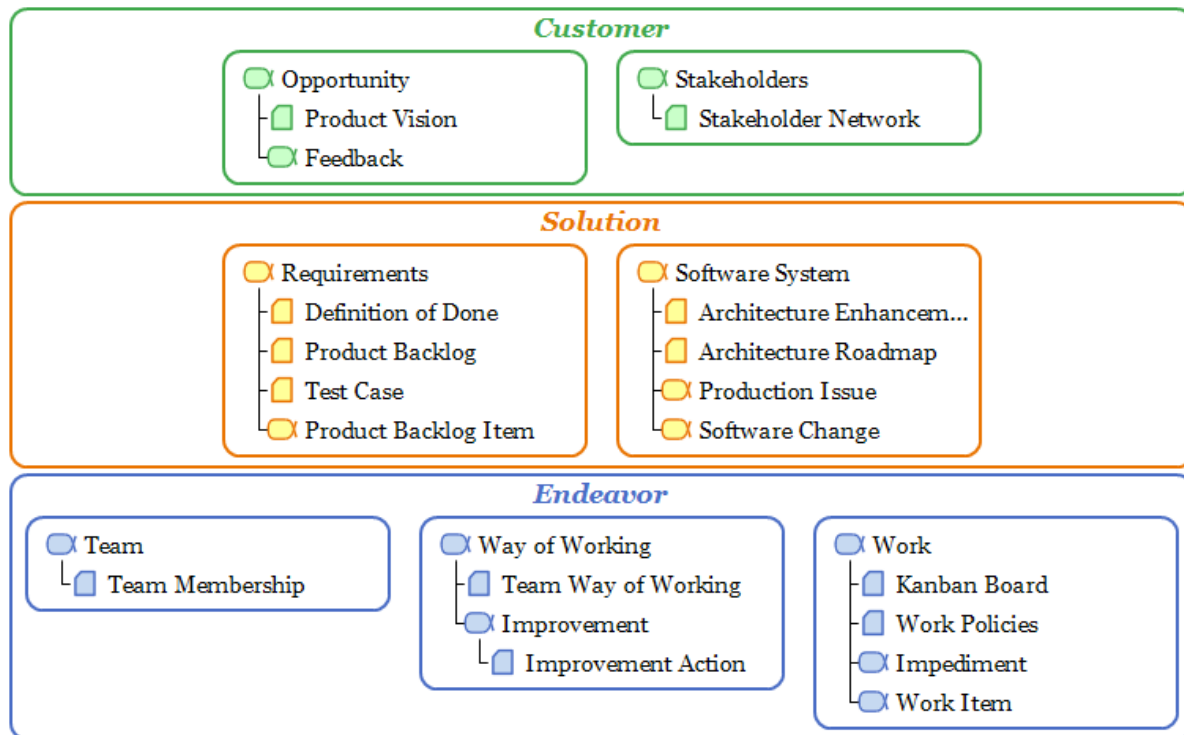
Composing our chosen practices with IJI Practice Workbench is a simple and fully automated process, which provides the team with a way of working which describes exactly what they are committed to doing, and nothing that is not relevant or that they are not committed to doing.

¹ The Agile Essentials and Agile at Scale practices are all available to browse in the IJI Practice Library [<https://practicelibrary.ivarjacobson.com>]. The Local Agile Practices were built as examples for previous articles in this series using IJI Practice Workbench [<https://www.ivarjacobson.com/esswork-practice-workbench>]



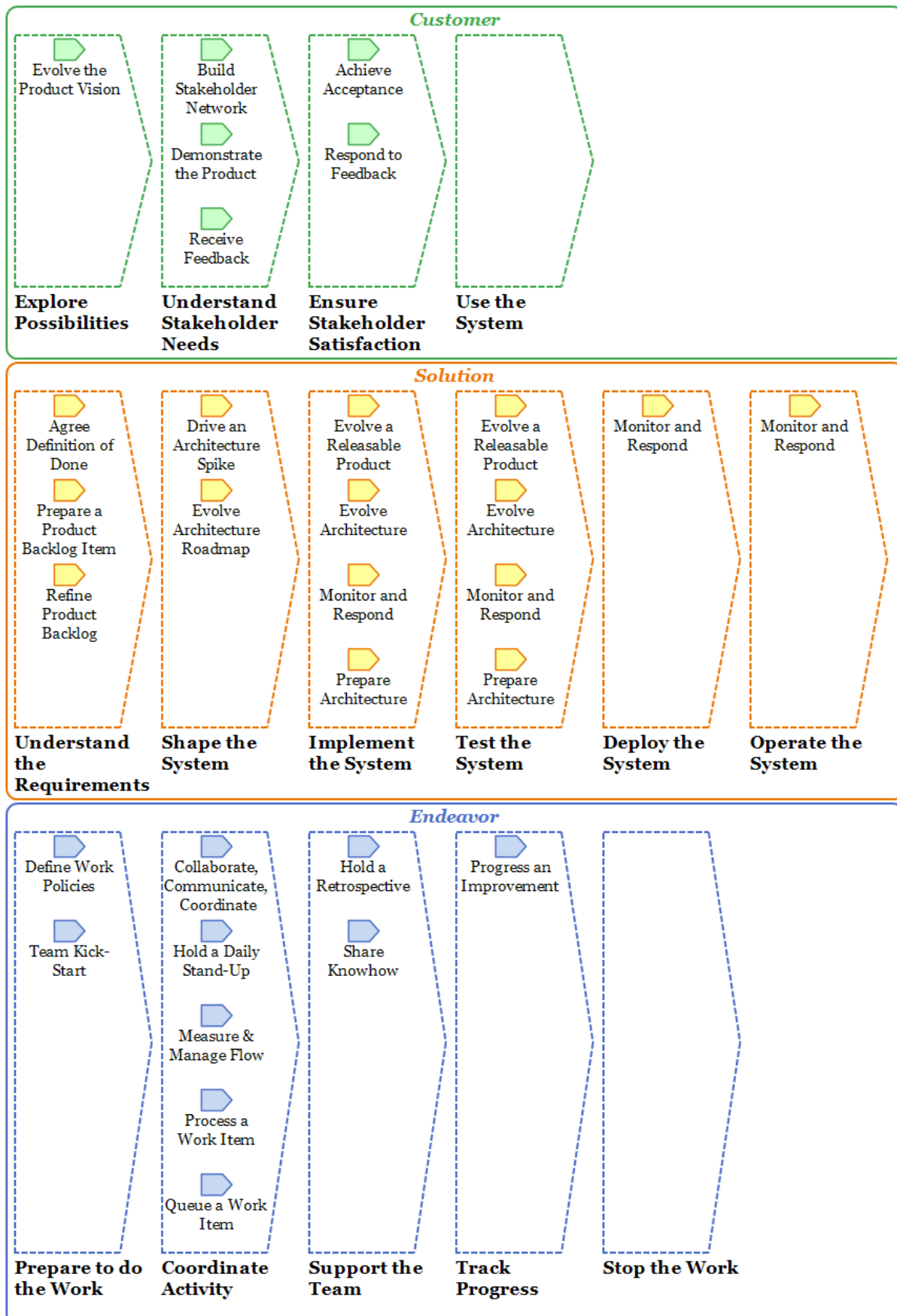
Once composed, this method can be automatically generated and made available to the team and all its stakeholders as a browsable HTML site, or as a printable pack of practice cards, or as a deck of HTML “eCards” that is browsable on everyone’s smartphones. So now there is no reasonable excuse for anyone not to know exactly how the team is working.

The browsable HTML, among other things, gives overviews of the contents and coverage of the overall way of working, including coverage of areas of concern and alphas, as shown below:





It also shows the coverage of Activity Spaces as shown below:



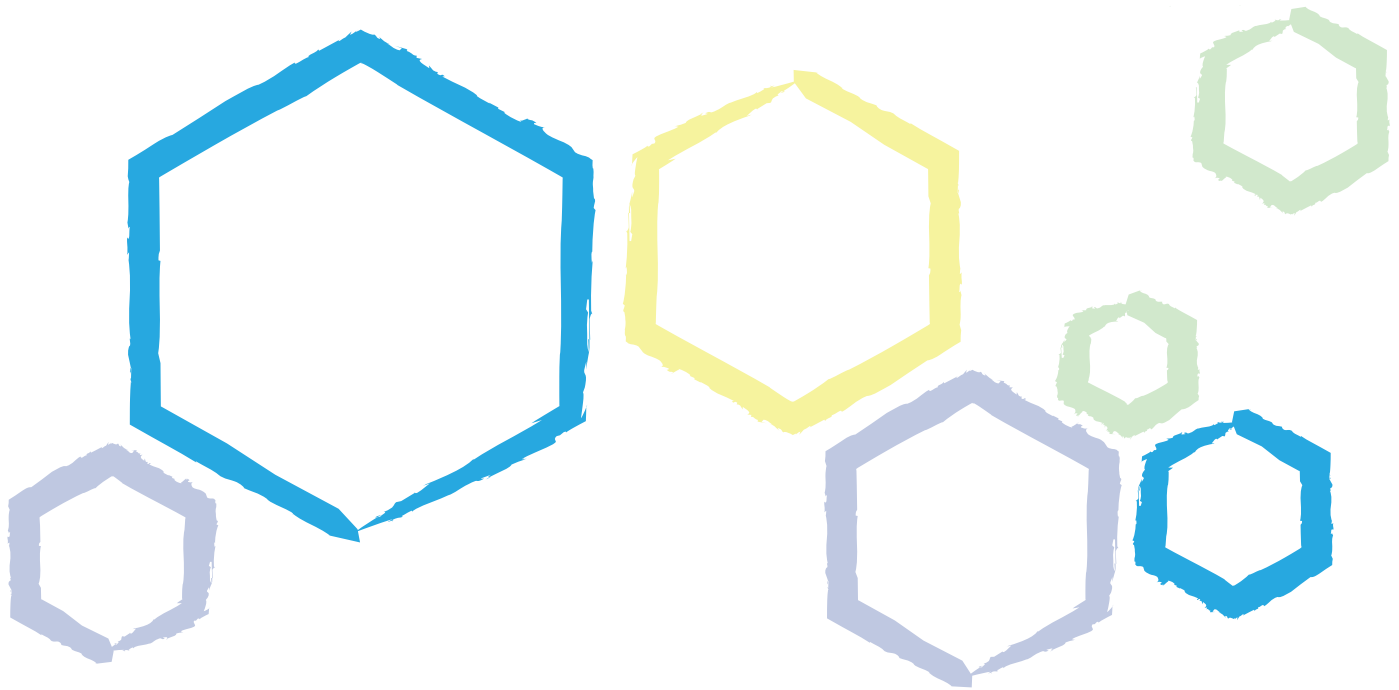


Note that while this gives good coverage for us as a team, it does not cover all possible spaces. Not every team needs to cover every base. The key to meeting process governance and compliance needs is that it is clear and visible what spaces the team's way of working does cover, and how.

The bottom line advantages are:

- Teams rapidly agree a tailor-made way of working that is exactly right for them
- The way of working contains no spurious elements that are not actually relevant
- Everyone is clear on exactly how the team is working
- For process governance and compliance, the coverage of the way of working is clear and visible.

UP NEXT: *Communicating, Collaborating and Coordinating*



Empowering Agile Teams within a Learning Organization

An ABC Guide to Leveraging Adaptive,
Bite-Sized, Composable Practices



In Part 5 of the series, we looked at how any team can rapidly create a way of working that is exactly right for them by reusing bite-sized practices and composing them into a method that represents their team way of working. In Part 6, we will look at how the team can leverage their agreed way of working to better communicate, collaborate and coordinate as a team and with their stakeholders.

Lean and agile principles and values rightly emphasize visibility in all things. This includes the work that the team is focusing on achieving to generate new value, and their working policies and practices. Having a clear, visible contract with customers and stakeholders of the team is key to ensuring that expectations are truly shared, and that those outside of the team know how to best engage and interact with the team, all of which is key to building a trust-based relationship.

The idea that teams need to tailor a way-of-working and clearly communicate the tailored way of working is far from new. Traditional all-encompassing process frameworks pretty much always came with instructions attached to “tailor to fit”, and even included mechanisms for doing this tailoring.

Traditionally this involved producing a document that described how the available process options have been adopted, applied and tailored. In the Rational Unified Process, for example, this was known as a “Development Case”. In other methods, it is variously called a “Development Plan”, “Development Approach Document” or similar.

Often the production and approval of this documented way of working was associated with an initial phase gate that controlled the “permission to start the project proper” (e.g. an “Inception”, “Feasibility” or similar phase gate) – after all it would be remiss of us to start working without pre-thinking and declaring how we are going to work, wouldn’t it?

The problems with this approach, though, are many-fold:

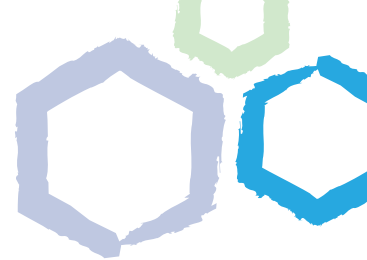
1. It takes time to write the document, and delays us in starting to build value into the product
2. Cost-of-delay theory teaches us that time is literally money (every day we delay starting is a day we delay the accrual of value as set out in the business case, which is money lost that we can never get back)
3. While the instructions with the process says we should “tailor it down”, actually it looks like it is there for a reason, so it is difficult to know what we can safely leave out
4. Because the document has a template, we write words in every section, which often ends up with the author restating the process framework in their own words
5. This is worse than a waste of time effort and money, if anyone looks at the document, it just causes confusion – which words are the right words, the ones in the document or the ones in the original process framework?
6. “No plan ever survives engagement with the enemy”, and our approach will change and be different from the one documented, but do we ever go back and change the document? Not usually – after all, we got through the phase gate and no one ever rechecks over time that we still have a documented view of the process we are following
7. The document is written by an individual, often a Project Manager, often before the team that will do the work is even recruited
8. The team certainly don’t buy into the documented process (and often don’t even read it)
9. So the document is not actually worth the cost of the paper it is written on, let alone the cost-of-delay that it has cost us as an organization.

Neither does agile solve this problem particularly. We are often offered the choice of either “doing Scrum” (or DAD, or LESS or SAFe), which is unlikely to fit exactly, and should be adapted by the team over time anyway, or doing “Scrum-not-quite” (or SAFe-not-quite, or ...), in which case how do we communicate the ways in which it is “not quite”, and how and why it is different? The problem remains essentially the same.

With the agile, bite-sized practice approach we can avoid all these challenges and get visibility of our approach as a bi-product of rapidly agreeing the approach as a team.

As we would hope and expect with an agile approach, we can choose whether we want to use simple, physical tools to do this (in this case, practice “poker cards”), or more sophisticated electronic tools. The former work well when we start up, particularly if we are co-located. The latter we may move to as we get going, or scale up, or if we are distributed. Let’s look at each in turn.

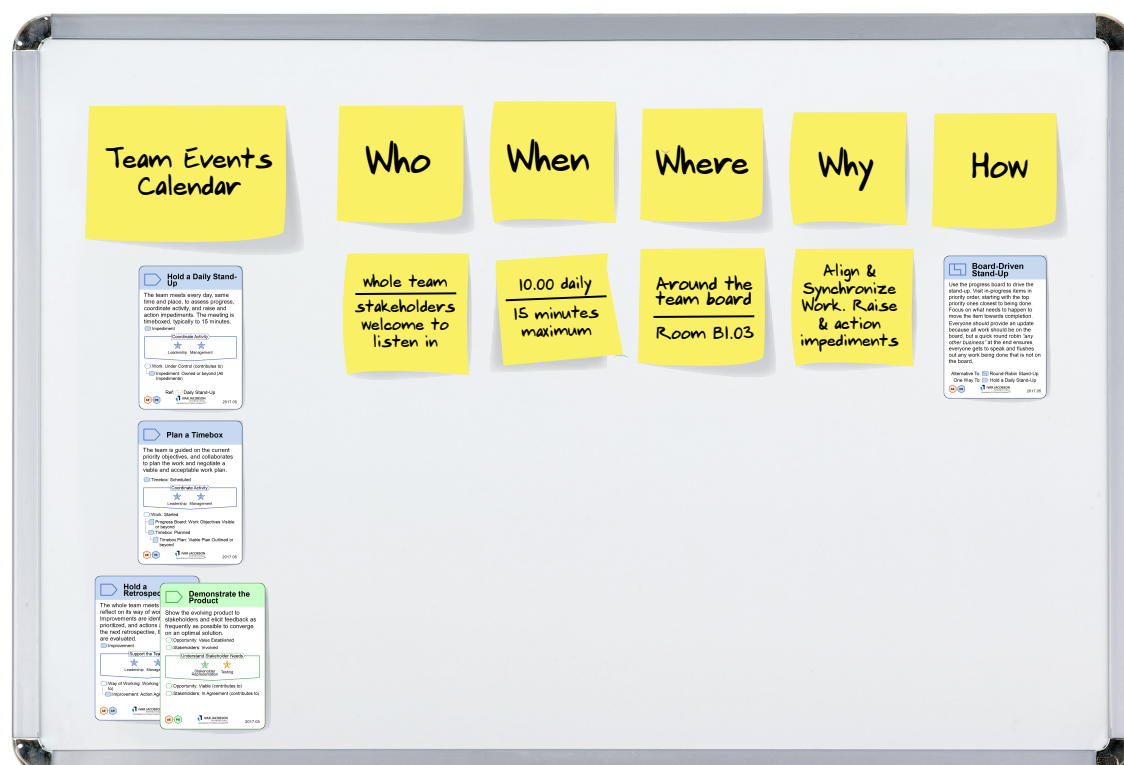
With the practice cards, there are some simple “games” we can play to agree what practices we want to adopt and how we want to implement them.



First we can select and prioritize the candidate practices that we could adopt. The diagram below shows an example of how we can do this by deciding for each practice, how well we do it now versus how critical we see it as being for us to excel as a team in this practice space.



Secondly, for each practice element card, in turn we can stick it up on a board, and add a few words to describe how we are doing this, for example to form a “team events calendar” as shown below:





Thus, simply by quickly working through this to get clear ourselves as a team how we want to work, we have something visible in our team room that documents how we work.

If we wish to work electronically, IJI Practice Exchange provides similar capabilities. We can first create a page to act as a “shop window” for our team, and select the practices we are using:

The screenshot shows the 'Practice Exchange' interface for the 'Yellow Team'. The header includes the Ivar Jacobson International logo and navigation tabs: Home, Methods, Practices, Assets, Forums, Resources, and Teams. A search bar and links for 'My account' and 'Log out' are in the top right. The team's profile section includes 'About Our Team' (a small team developing an architecture proof-of-concept), 'About Our Way of Working' (using agile practices within a Kanban-style continuous flow model), 'Team Members' (Liza Analyst, Fred Developer, Stefan Tester), and 'Team Method' (Agile Essentials with Continuous Flow). A description of the Agile Essentials practices is provided, along with a link to 'Practice resources'. On the right sidebar, there are sections for 'Admin Tools' (Create Forum topic, Create Practice, Create Resource, Create Team), 'Links' (Forum, Team resources), 'Latest Team Forum Updates' (Extending our Retrospective practice by Liza Analyst on 2016-11-16 12:07), and 'Team Resources'.

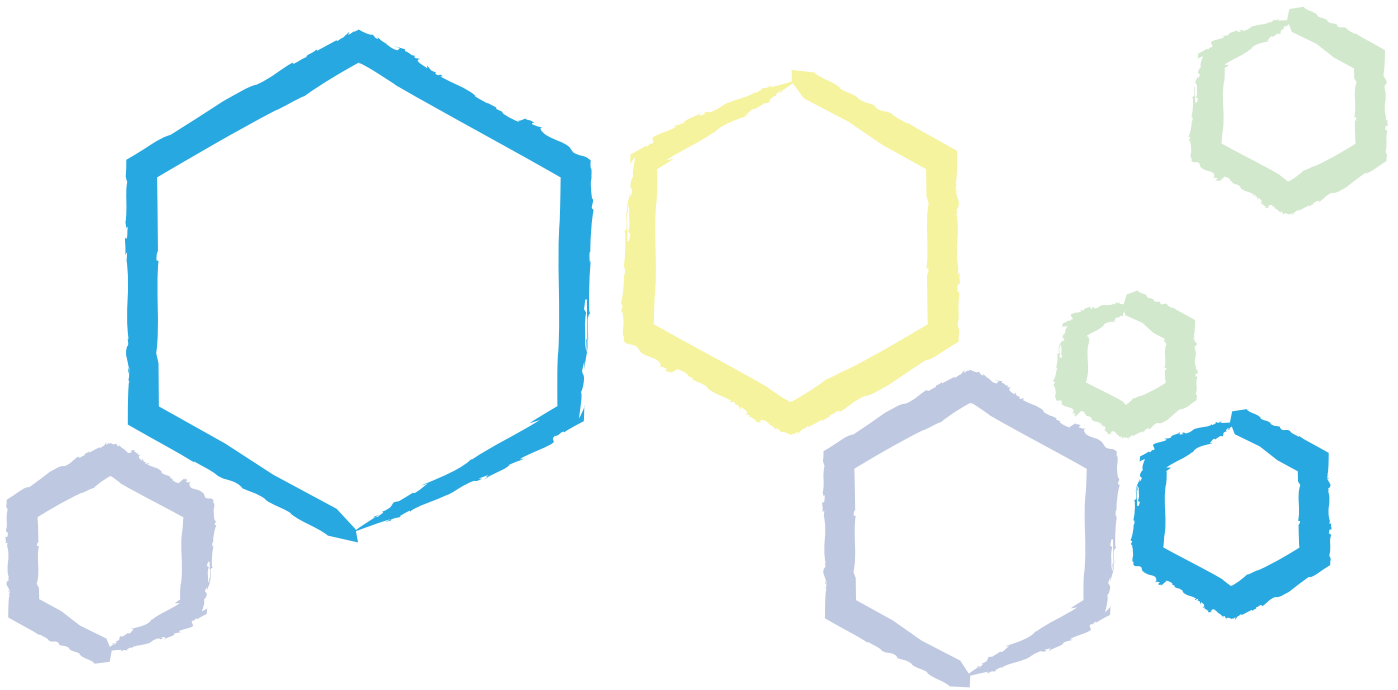
And we can also annotate these with a brief description of how we are working as a team, just as we can do with the physical cards and whiteboards as a co-located team:

This screenshot shows a 'Team Practices' card for 'Agile Architecture Essentials'. It includes a brief description: 'Guide the evolution of a solution approach that adapts to changing needs and challenges.' and a link to 'Practice resources'. To the right, under 'Team Approaches', there is a link to 'Architecture Roadmap updated'.

And what we have as a result is a way of working that we agreed and are happy with as a team, that we were able to agree in double-quick time, and that is visible across the organization, including all the updates we make as we go ... as we adjust our plans, as we “engage the enemy” ...

And what the organization has is a list of all the agile teams that have done this, all in one place, with the ability to drill in and see how each one is working as a team. This in turn enables cross-team learning and cross-pollination of good practices across teams, either on an informal basis, or in a more proactive and sophisticated ways, for example via agile Communities of Practice within the organization.

UP NEXT: *Building a Library of Methods and Practices*



Empowering Agile Teams within a Learning Organization

An ABC Guide to Leveraging Adaptive,
Bite-Sized, Composable Practices



In the previous article, we looked at how teams can leverage their agreed way of working to better communicate, collaborate and coordinate as a team and with their stakeholders. In this edition, we look at how organizations can ensure that there is a good set of practices available for teams to use and mechanisms in place for cross-team learning, and evolving and extending their practice knowledgebase.

There is no shortage of words of wisdom out there about how to do software engineering. In fact, there is a veritable clamor of voices espousing different approaches. Often these are formulated as all-encompassing process frameworks (Scrum, SAFe, DAD, LESS, Nexus, ...). But there are also many popular practices that were originally shared as more informal supplementary advice, that over time have become accepted “standard” practices because of their usefulness and value - such as “Card, Conversation, Conformation”¹ or “INVEST”².

Increasingly, it could be argued, the need is not for more and more words, but for a way for teams and organizations to find, extract and make effective use of the nuggets of wisdom that are out there somewhere, often embedded within some larger method framework.

We have seen how Essentialized practices form the basic “unit of adoption” for a team’s way of working, but how can we access, acquire, share and reuse these practices in a controlled and effective way?

An example of an Essentialized practice library is the IJI Practice Library³. This is a large and growing set of 25 practices, organized into three pre-composed reference methods as follows:

- Agile Essentials (7 practices)
- Agile at Scale (10 practices)
- Essential Unified Process (8 practices)

The library is a free-to-browse resource, thus enabling organizations, teams and individuals to browse and select practices for adoption.

But a key part of the challenge is for organizations to choose which practices are and aren’t valuable and appropriate for them, and for teams within these organizations to do the same. What is needed is a repository where organizations can collect and manage the practices and enable teams to access them.

IJI Practice Exchange is an example of a tool that enables this. Organizations can set up an instance of a Practice Exchange for the organization and load into it the practices and methods that they want to make available.


1 See <http://ronjeffries.com/xprog/articles/expcardconversationconfirmation>

2 This acronym for the quality criteria of a Product Backlog item was originally proposed by Bill Wake: see <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks>.

3 <https://practicelibrary.ivarjacobson.com/start>.



Below is a picture of the home page of IJI's internal practice exchange, with the available methods and practices clearly visible:

**IVAR JACOBSON**
INTERNATIONAL

Practice Exchange

[My account](#) [Log out](#)

[Home](#) [Methods](#) [Practices](#) [Assets](#) [Forums](#) [Resources](#) [Teams](#)

Welcome to the IJI Practice Exchange, supporting the [practice hub in agile change initiatives](#). On the Practice Exchange we share assets and collaborate around essential practices and how they are used to roll-out ways of working in agile software development organizations and their teams. Currently available practice packs for this purpose include:

Practices

Product Ownership: PO
Product Backlog: PB
Agile Teaming: AT
Agile Retrospective: AR
Continuous Flow: CF
Agile Timeboxing: TB
Agile Development: AD
Daily Stand-Up: DS
User Story: US

Methods

Agile Essentials: AE
Agile Essentials w. User Stories: AE+US

Additional Reference Methods

Agile Essentials w. Continuous Flow: AE+CF
Agile at Scale w. Agile Essentials: A@S+AE

Agile Essentials practice pack

Practices

Adaptive Funding: AF
Agile Governance: AG
Shared Backlog: SB
Release Management: RM
Product Management: PM
Periodic Alignment: PA
Team-of-Teams: TT
Agile Architecture: AA
DevOps: DO
Development Support: DV

Methods

Agile at Scale Essentials: A@S

Agile at Scale Essentials practice pack

Practices

Product: Unified Process Lifecycle
Team: Architecture Component
Iterative: Component
Use Case 2.0: Test Execution
Process: Essential Unified Process

Methods

EssUP: Essential Unified Process

EssUP practice pack

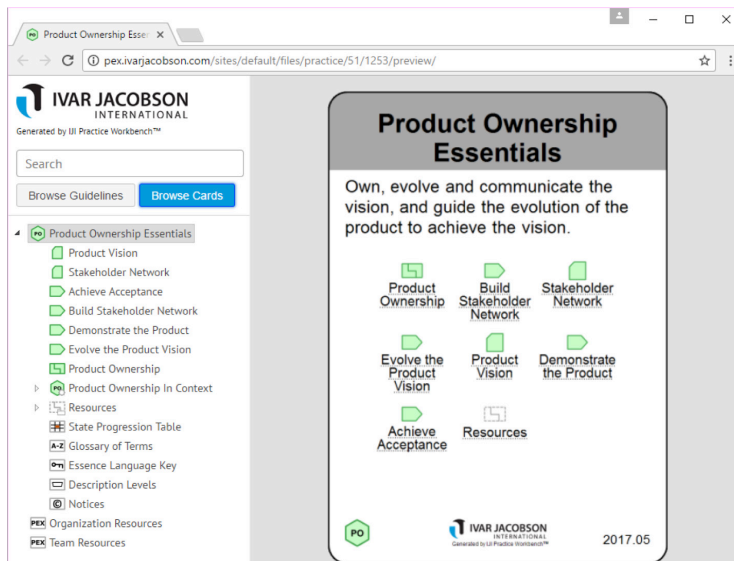
Admin Tools

- [Create Forum topic](#)
- [Create Practice](#)
- [Create Resource](#)
- [Create Team](#)

We saw in Part 6 how teams can now use this Practice Exchange facility to set themselves up with a cohesive and visible Way of Working, which they can rapidly construct by selecting from the practices that are available on the organization's Practice Exchange.



All of these practices can be browsed as HTML Guidelines and/or “electronic cards”, as shown below:



Organizations can also add their own “resources” – such as additional guidelines, templates, examples etc.

Each method, practice and team in the Practice Exchange also has an associated discussion Forum, so that practitioners and stakeholders across the organization can communicate about the application of practices, and collaborate to evolve and improve them, for example as part of the work of the specialist Communities of Practice within the organization. These Communities of Practice are often responsible for evolving new practices, often based on capturing and propagating team-level practice improvements, extensions and innovations, as described in Articles 3 and 4 of this series.

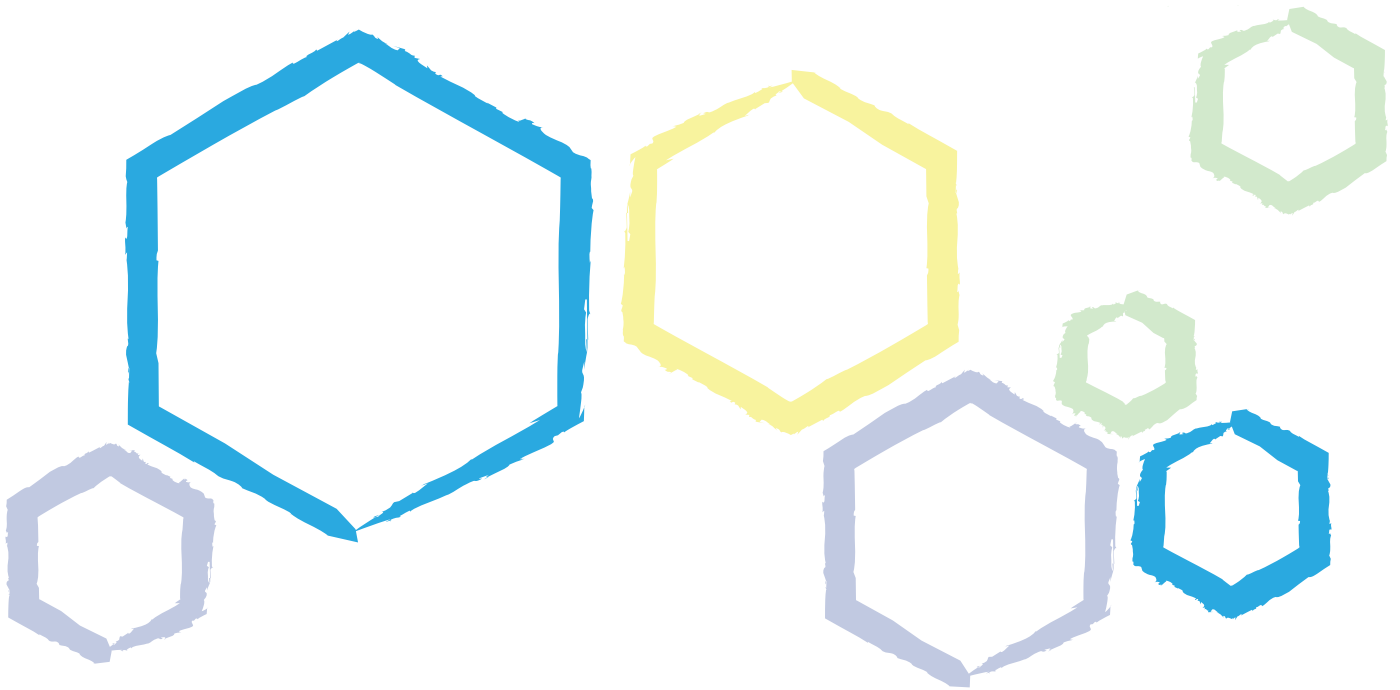
The bottom-line result is that organizations can now grow and maintain a knowledge-base of best practice that evolves over time to meet the current and future needs of all the different types of programs, projects and teams within the organization.

As we will see in the next article in this series, this represents a paradigm shift for the industry, which means:

- No more inefficient and ineffective “one-size-fits-all” single, monolithic processes
- No more need for “rip and replace” of processes as industry approaches change over time
- The knowledge and expertise of teams, organizations and the industry continuously builds and evolves over time.

UP NEXT: In the next article we will explore this vision and its far-reaching implications for the future of the software engineering industry.

Part 8: Today's Mission and Tomorrow's Vision



Empowering Agile Teams within a Learning Organization

An ABC Guide to Leveraging Adaptive,
Bite-Sized, Composable Practices



In the previous article in the series we looked at how organizations can move towards becoming truly learning organizations through Teams and Communities of Practice innovating and propagating professional practices and learning. In this article we summarize the Essence practice-based approach to software engineering that we have been exploring, examining the ways in which it represents a radical transformation opportunity for the industry, and assessing how far we have come in realizing the vision, versus how far we still have to go.

There is a powerful philosophy prevailing in the industry currently, as neatly captured in a quotation that is currently “trending” in industry presentations and papers, which is:

“Simple, clear purpose and principles give rise to complex and intelligent behavior. Complex rules and regulations give rise to simple and stupid behavior.” Dee Hock, Founder and Former CEO of VISA International

“Follow the organization’s standard methodology” might seem like a simple and clear edict, but a moment’s reflection shows us that actually it sits firmly in the latter category of leadership that is liable to give rise to simple and stupid behavior, that is if the organization’s methodology is:

- All-encompassing – “one size fits all”
- Detailed – runs to hundreds of pages
- A relatively tangled mixture both of “what to do” and of “how to do it”.

The problems with such methodologies include:

- They take too long to learn
- They are too costly to maintain and embed
- They are constraining and restrictive – requiring costly periodic “rip-and-replace” changes
- Some of their content might fit some problems some of the time – but all of it certainly doesn’t fit all problems all the time
- Which means no project or team is ever actually “doing the right thing” if they are following it – and who can tell what they are doing or whether it is “the right thing” when they are not
- Being told how to do your job by someone else is disempowering, frustrating and demotivating for intelligent, creative and dedicated software development professionals.

So much for how not to do it. The harder question though is what, in positive terms, is the appropriate approach to enabling and empowering professional software engineering organizations and teams?

Imagine a development organization where there was genuinely just one single, simple principle:

- All teams are responsible for their development approach, including ensuring it is fit-for-purpose, and visible and acceptable to their stakeholders.

This is exactly the kind of “empowerment with maximum freedom and appropriate responsibility” that will drive intelligent behavior, including:

- Everyone on the team takes collective responsibility for success – no more blaming a process / way of working imposed upon the team from without
- The team understands that ultimately their responsibilities are to their stakeholders – customers, sponsors, and that empowerment includes communication and engagement with the appropriate stakeholder to ensure that the adopted way of working meets everyone’s needs.



While most agile teams would in theory accept these as the right principles, in practice there has been insufficient support in terms of materials and tools to help them rapidly agree a fit-for-purpose approach, make it visible, gain buy-in, and evolve it over time based on learning and feedback.

Hopefully through this series of articles it has become clear that we now as an industry have the basic tools we need to support teams in doing this, including:

- The Essence standard that enables proven “de facto” standard bite-sized practices to be shared for reuse – these are the basic “tools” in the mental “tool-box” of any experienced agile development professional, expressed in a clear, condensed and reusable format – typically a handful of “poker cards” that summarizes the essential aspects of each practice
- Libraries of these practices for teams to browse and select – e.g. IJI Practice Library
- Collaborative portals where the selected practices are made visible, embellished and annotated as a clear and visible expressions of each team’s way-of-working– e.g. IJI Practice Exchange
- Tools to enable practices to be adapted, extended, created and published as browsable HTML and/or accessible physical “Poker cards”, eg: IJI Practice Workbench.

The vision that IJI is working hard to realize is that in future this will be the way in which all industry intelligence, learning and knowledge is evolved and shared.

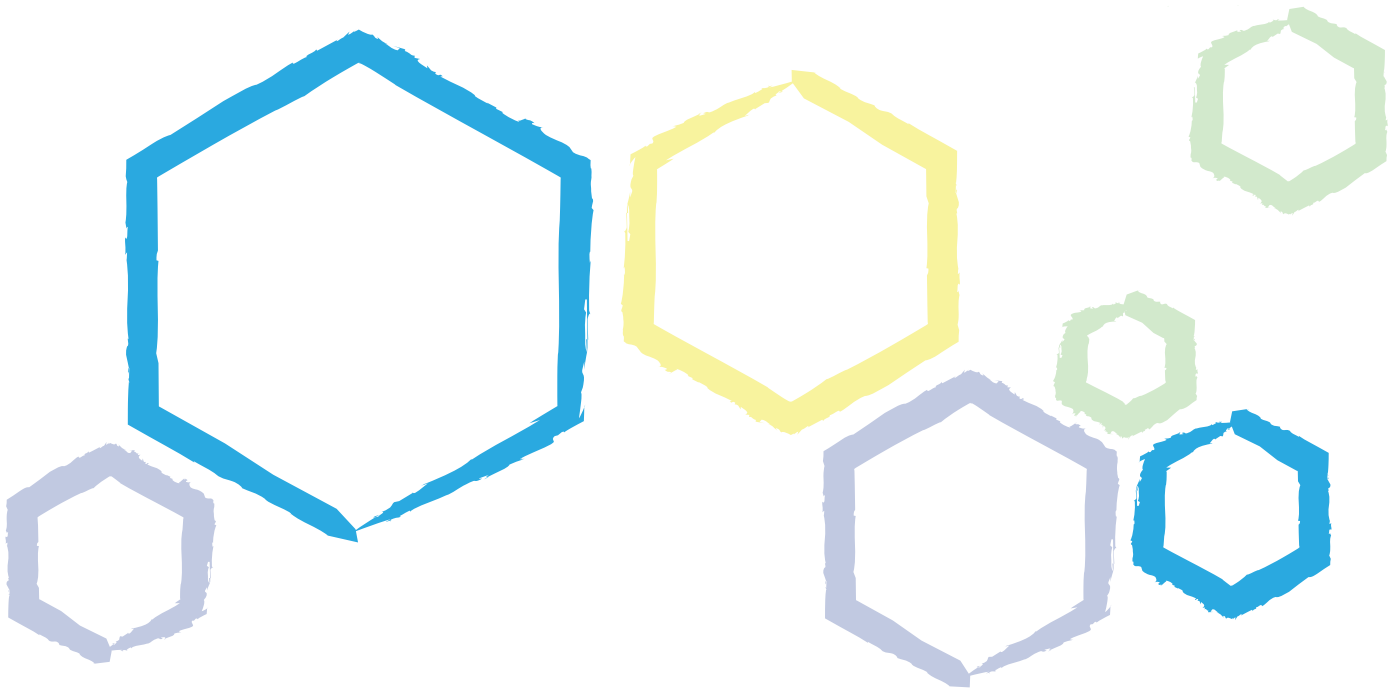
How close are we today? I think it is fair to say we are at the beginning of the journey, not the end. The standard way in which industry knowledge is packaged today is still within all-encompassing, “hermetically sealed” process frameworks (such as Scrum, SAFe, DAD, Nexus, ...):

- Each one is expressed in its own different way (“language” / “meta-model”)
- There is much overlap and commonality, but this is impossible to extract
- There are random variations in terminology – new words being regularly invented for similar or identical concepts
- While each brings some new wisdom to the world, extracting that new wisdom and using it in combination with other wisdom from other sources is simply not possible.

To realize the vision, IJI is working in partnership with a number of organizations, including through the SEMAT community, to support the work to “Essentialize” process frameworks, which means:

- Expressing the content using standard Essence constructs and symbols
- Structuring the content so that the common ground is clearly separated from what is unique, different or new
- Structuring the content into bite-size, reusable practices
- Making the practices available as succinct, accessible and useful “poker cards” and electronic equivalents, for teams to select, adapt and compose to form a cohesive and transparent way of working.

UP NEXT: *In the next and final article in this series we will look at how you can get started immediately – using Essence and Essential practices to bring immediate value and benefits to teams, and how you can build from here to transform your organizational agility, build world-class development capabilities and teams, and future-proof your ability to adapt to meet new challenges and leverage new innovations.*



Empowering Agile Teams within a Learning Organization

An ABC Guide to Leveraging Adaptive,
Bite-Sized, Composable Practices



In the previous article, we summarized the Essence practice-based approach to software engineering that we have been exploring, examining the ways in which it represents a key transformation opportunity for the industry, and assessing how far we have come in realizing this vision, versus how far we still have to go. In this article, we look at how you can start to benefit from tomorrow's development paradigm today.

In the last article, we painted a compelling future vision for the industry. But you don't have to wait to get value from Essentialized practices. These adaptive, bite-sized, composable practices and supporting tools are available for use by teams now, to help them improve the effectiveness and transparency of their ways of working and for organizations to embed effective but highly adaptive development practices.

This article outlines some simple steps that are the most common ways in which teams and organizations start to get immediate value from adaptive, bite-size practices, and build up to more transformational change that delivers massively increased organizational agility.

These steps are not the only path to value, and they don't have to be followed in this exact order, but they are set out in a way that shows some examples how you can start with small steps that lead to big value and transformational change.

Explore available practices

The first step is simply to explore some adaptive, bite-sized, composable Essence practices, including team-level agile practices, agile-at-scale practices, as well as iterative, incremental, architecture, requirements and other practices, all available for browsing by signing up for the IJI Practice Library at <https://practicelibrary.ivarjacobson.com>.

Acquire potentially useful practice cards

Whichever practices could help your team or teams better focus, communicate or execute, simply download and start to work with the practice. Practice "poker cards" can be accessed and printed off from each browsable practice in the IJI Practice Library, or professionally printed packs ordered from:

<https://www.ivarjacobson.com/publications/tools/agile-essentials-cards>.

Use the cards to explore as a team what practices can add value

There are plenty of card games that you can start playing straight away with your team to help you explore how you can optimize your team's agile way of working. Start to play the games and learn how to apply them at an interactive workshop, either for new teams to rapidly agree and embed an effective way of working (<https://www.ivarjacobson.com/publications/brochure/agile-essentials-team-quick-start>) or for existing teams to inspect and improve their ways of working (<https://www.ivarjacobson.com/publications/brochure/agile-essentials-healthcheck>).

Smoothly scale from team-level to scaled agility

This is achieved by playing agile team games as described in article 6 of this series, using practice cards from the Agile at Scale practice cards (<https://www.ivarjacobson.com/publications/tools/agile-scale-essentials-cards>), and enabled through workshops as described above.

Use the cards to make the team's way of working visible

This is achieved by playing agile team games as described in article 6 of this series, and enabled through the workshops described above.



Make the practices available for reuse electronically across the organization

This is achieved using the IJI Practice Exchange. Please contact us for more information:

<https://www.ivarjacobson.com/contact>

Ensure all teams make their ways of working visible

This is enabled through simple point and click operations within the IJI Practice Exchange.

Enable teams to capture and share practice innovations

This is enabled by the IJI Practice Exchange, through the discussion Threads on Forums and the uploading of supplementary resources such as templates, examples, hints and tips and other experience-based guidance.

Identify capability gaps and build a change backlog of improvements

This can be kick-started by using various health and progress checking and gap analysis games, using both the team and scaled agile practice games, and the alpha state cards and games:

<https://www.ivarjacobson.com/alphastatecards>.

IJI consultants are experienced in supporting organizations in scaling up to directed organizational agile transformation and have a proven approach to driving sustainable change benefits:

<https://www.ivarjacobson.com/sustainable-agile-transformation>.

Organize Communities of Practice around capability building and practice innovation

Communities of Practice can be established as part of a sustainable change initiative, and collaboration can be focused around the development and enhancement and communication of adaptive, bite-sized practices as enabled with the IJI Practice Exchange.

Join the community that is shaping future of software engineering

SEMAT is a global community of people, some organizations and universities that supports the Essence initiative to create a common ground for software engineering. You can get involved at

<http://semat.org/> or join the conversation on LinkedIn: <https://www.linkedin.com/groups>



About Ivar Jacobson International

IJI is a global services company providing high quality consulting, coaching and training solutions for customers seeking the benefits of enterprise-scale agile software development.

We are passionate about improving the performance of software development teams, and maximizing the delivery of business value through technology.

Whether you are looking to transform a single project or program or your entire organization with lean and agile practices, we have solutions to suit your needs.

www.ivarjacobson.com

Sweden
+46 8 515 10 174
info-se@ivarjacobson.com

United Kingdom
+44 (0)207 953 9784
info-uk@ivarjacobson.com

Asia
+8610 82486030
info-asia@ivarjacobson.com

Americas
+1-703-434-3344
info-usa@ivarjacobson.com

