

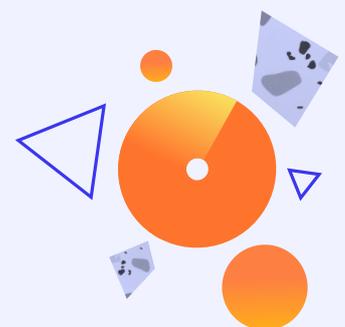
SUMMER 2022

The four facets of agility: a foundation for successful transformation

The four facets of agility: a foundation for successful transformation

Contents

Not another agile think piece	3
Why should I care about agile?	5
Failure is not an option (but it is a possibility)	11
Meet the four facets of agility	17
Part one: Complex thinking, simply put	21
Part two: Nurturing an agile mindset	24
Part three: Creating a culture for new processes to thrive	28
Part four: Take the tools off auto-pilot	39
Embrace agile with Aligned Agility	42
Unlocking the four facets of agility	46



Agile is more than a box-ticking exercise to keep up with the competition – it's a state of mind.



Not *another* agile think piece

If this is the first time you're reading about agile, which corner of the internet have you been hiding in? We ask because a lot has been written on the subject. We're talking opinion pieces, consultancy whitepapers, ream upon digital ream of tips and tricks to help your business be agile, do agile, eat, sleep, and breathe agile. Frankly, it's exhausting. And for most organizations stuck at the starting line or struggling to scale, it can all feel a tad overwhelming.

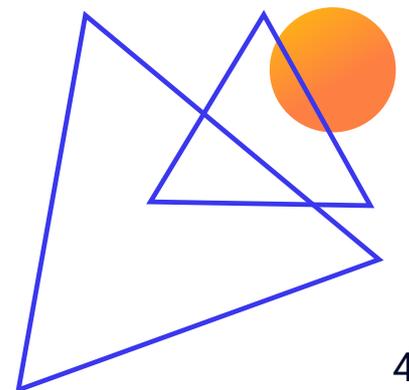
Why then, we hear you ask, is Aligned Agility wading into these murky waters to serve up its own slice of agile acuity? The answer is that we've hit upon something others have failed to impart: a holistic approach that encourages productive dialogue around the various aspects of agility. This is much more than the 'people, processes, and tools' conversation that others are having. We view your agile transformation through the lens of the four facets of agility, which can help you understand where you might need to focus more time or energy.

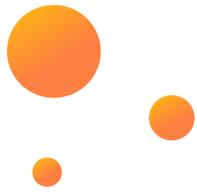
Everyone knows when it comes to agile, there's no one-size-fits-all checklist that works wonders and ensures success. Agile takes time and it's messy. But using the four facets as a foundation, you'll know exactly where the messy bits are that might need some more attention.

Passing the torch

Aimed at senior leaders and executives, this book shines a light into the darkness. It covers the 'whats' and 'why bothers' of agile and the failure-shaped elephant in the room, because things don't always go to plan. Then it breaks agile down into four facets – a synergetic strategy to navigate agile holistically, rather than pinning false hope on implementing it through a narrow lens. Finally, we explore some of our working in practice – real-world agile transformations and the lessons we've learned along the way.

Wherever you are on your digital transformation journey, we hope you will find something of real worth here to propel your people forward. From questions to help you shift your mindset as a leader to practical advice based on years of expertise, our agile consultants have laid it all on the table.





Why should I care about agile?

Let's get back to basics.

It's been two decades since the [Agile Manifesto](#) was first let loose on the world. That simple declaration, put together by 17 software development leaders in Utah's Wasatch mountains, included four central values that positioned the value of individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; and responding to change over following a plan as a better approach for developing software.

In the years since, so many layers have been piled on top that it's easy to lose sight of the core thinking behind what agile is and why it came into existence in the first place – the heart of agile.

An agile approach rests on [12 principles](#), including:

- **Focus on customer satisfaction as the highest priority through early and continuous delivery of valuable software**
- **Welcoming changing requirements even late in development; ensuring daily collaboration and face-to-face conversations**
- **Importance of self-organizing teams who reflect on their effectiveness and fine-tune as they go**

'At the gathering atop Snowbird, we left our egos at the door and were able to get to the gist of what our various software development methodologies and approaches had in common. It seems miraculous that we got agreement on the four main bullets, but then again, many of us were seasoned object modelers and software development practitioners. In my humble opinion, these four values have stood the test of time, as have most of the 12 principles, and will continue doing so for the next 20 years.'

Jon Kern, co-author of the *Agile Manifesto* and Agile Consultant

The way things were

Before agile, most organizations took a big, upfront design and 'all-in' waterfall approach to software development. They held the assumption that customers knew what they wanted and that nothing would change once a project plan was in place. Defects, unnecessary features, and delays were just some of the drawbacks, and poor planning would be considered the cause.

The manifesto might have coined the term, but agile working practices were going on for many years before that, both [in and outside software development](#) – born out of a need for a different approach.

When software teams found something that worked, they shared their methodology with others – and these ideas developed into widely used frameworks. Central to many of these frameworks were close collaboration between developers and other stakeholders across the business, and self-organizing teams. But a focus on delivering something of value to customers frequently was key. Scrum, widely used today, was one such framework.

What is agile?

The word ‘agile’ was chosen as the name of the 2001 manifesto because it represents the adaptiveness and ability to respond to change that the whole approach hangs on. But agile is not, as it is often mistakenly referred to, a methodology, framework, or system on its own. It’s a mindset, underpinned by the values and principles outlined above and supported by many practices, processes, and tools.

It uses iterations, frequent feedback, and close customer collaboration to always deliver the highest-priority items of the moment (determined by customers). An agile approach – to software development or in any business setting – embraces change and collaboration; depends on small, self-organizing, cross-functional teams; and values quality, inspection, and adaptation.

Being agile means an organization can change quickly, and can strategize and restructure as necessary, enabling it to create value, scale, and succeed in a turbulent environment.

By its very nature, agile will never be something you tick off a list by following a set of instructions. It's the means, not the end.

Most people are familiar with the practices and tools that help businesses to be agile: stand-ups, sprints, and retrospectives; and tools such as kanban boards and digital tools like Jira. And then there are the frameworks, such as Scrum, Extreme Programming, SAFe, LeSS, and [many more](#); which form the basis for a team's or organization's agile methodology – the way it implements agile principles and practices.

And while some frameworks address the concept of strategy, business value, and portfolio management, as a business leader, you'll still have to do the hard work of determining what that is, figuring out a structure for delivery, and defining customer value. Because these things are done differently than in plan-driven methods and are often foreign to people who have spent their careers working in different ways, this can be a real challenge.

Is it for me?

If you're a senior leader in R&D, IT, or software development and delivery, chances are you're already familiar with some aspects of agile, even if it's not working well for you at the moment. If that's the case, then don't give up just yet. As we mentioned, this is a long game and it takes time to iron out the kinks before new ones inevitably appear. That's part of agile's magic – it never stops doing this job.

Arguably, nowadays every company is a software company in some form or another. But even if you think your organization is as far from Silicon Valley as you could possibly get, don't switch off. As the original

thinking behind the manifesto attests, rigid processes, siloed thinking, and unmoveable hierarchies plague industries far beyond software development. The benefits of an agile mindset are not to be sniffed at, no matter your sector.

To succeed in this unstable, digital-first world, all companies need to shed archaic notions of the corporate life that was and embrace doing what's best for the customer (not following processes just for the sake of it) to deliver greater value right now. This isn't a free-for-all anarchy where documentation and planning ceases to exist. It's a movement that recognizes the rigidity and ridiculousness of over-documentation and the limitation of planning for something you can't yet even imagine.

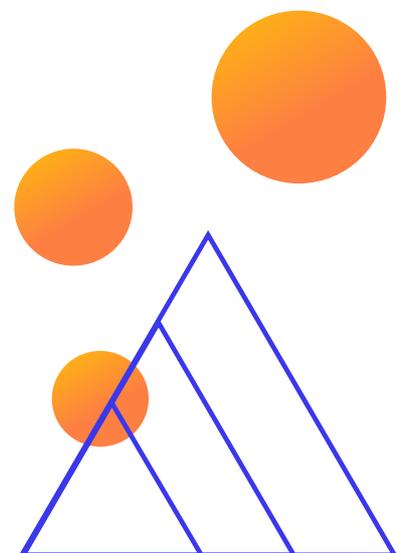
Why should I care?

Agile isn't a word on a plaque that you can hang up in your corner office. The proof of the pudding is in the eating. But agile does help solve many of the most pressing problems facing businesses. If your organization is struggling to cope with continuous change, can't keep up with the competition, and software is becoming central to what you do, then agile ignorance is not doing you or your people any favours.

It helps you to embrace uncertainty, vital for the increasingly volatile and complex situations most companies are finding themselves in, and put your customers front and center. What's more, most people love working this way. If retention or employment engagement are low, agile ways of working, which aim to provide more autonomy for people in their day-to-day work, can do wonders for morale and engagement, keeping the best people on your payroll.

It needs to be stated, however, that agile is not a process that can be force-fed – imposing rigid rules and processes on your people definitely won't lead to success (or help their morale). In fact, [imposition of agile practices is the antithesis of agility](#).

Beyond that, the advantages are plentiful. As a nimble organization, you can more easily respond to change, assessing and fulfilling customer needs more quickly than the competition, and ensuring quality is built in from the beginning. Risks are reduced, with early warning signs to get a handle on any issues before they escalate. Your ROI should increase as you continuously deliver products or services, discovering along the way what people actually want. And, perhaps most importantly, your people will be happier and more productive.





Failure is not an option (but it is a possibility)

If you're already an agile sceptic, waving your 'agile is dead' flag with pride, then it's not hard to imagine why. The digital ether is littered with articles espousing the complexity of agile, the 'good in theory, terrible in practice' experience of vocal opponents, and the high-profile failures of scaling frameworks at large organizations that sit under the agility umbrella.

Investing in agility – from mentoring workshops and coaching to new tools – can be costly, but it's an investment that will pay dividends if done right. When you're promised unicorns and rainbows, and you pile on time, resources, and big expectations, you're right to be frustrated when your transformation crashes and burns.

The key to avoiding failure is understanding why some transformations don't live up to lofty goals and recognizing the common pitfalls putting other companies on course for catastrophe.

Agile isn't a cure-all

Often agile comes up in a crisis. *Things aren't going right, so how are we going to fix them? What are other companies doing? Maybe we should be doing agile.* This common misconception that agile is a panacea for all an organization's ills means most people get off on the wrong foot to begin with. Don't assume it's a cure for every ailment and that a few flashy frameworks and conversations about self-organizing

teams will turn things around. If your organization is riddled with problems and bureaucracy, bringing agile in, hoping it can 'save the day', is a fool's errand.

Agile isn't something you do – it's something you are

Which brings us nicely on to the next point – this notion of practising agile, 'doing' agile, is a falsehood. In actual fact, agile is a mindset. The techniques, tools, and rituals your organization incorporates to support its agile thinking are secondary to the attitude both you as a leader and your people need to make it work. There's no right or wrong framework – what works for one company will be problematic for another. And what works today, based on your current people and processes, might not work a few months down the line when needs change. The agile manifesto is a philosophy, not a recipe. The 'inspect and adapt' mindset goes for the frameworks and practices as well as the products companies produce.

Agile isn't a quick win – you're in this for the long-haul

This thinking centers around another lie – that agile is an end goal, when in truth it's a means to an end as well as a continuous journey. Anyone trying to sell you a framework that feels like a flashy alternative to the static methods you're already using isn't going to help you be more agile. If someone promises you that you can install agility on a rapid timeline to save the business, they're leading you up the garden path. If you're singularly focused on what the company's going to look like 'after it's transitioned', then you need to adjust your thinking. With agile there isn't an end point but a never-ending exploration of what is, and isn't, working.

Agile isn't blindly following a framework

Don't trade in enforcing one set of rigid thinking or heavy processes for another. A truly agile company is flexible and will be able to adjust continuously. If adhering to a rigid framework is the answer to your problems, then you're not practicing agile. More importantly, if people aren't allowed to improve their own processes, you'll end up with rapid disengagement in the organization. Rather than worrying about doing things to the letter, know that your chosen framework might not have all the answers – be open to uncovering better ways.

Following the framework is never the point. The point is to figure out the best way of delivering the most value for your customers as you can. And don't lose sight of common sense. Agile is stepping into the unknown, embracing the uncertainty of what you might find, while keeping your eye on the end goals.

Agile won't fix what's broken

Face facts: there will always be problems, from internal issues and customer complaints to global pandemics that require adaptation in an instant. Being really agile won't stop problems from occurring or magic them away. What it can do is help your people identify issues before they get out of hand, shining a bright light into the darkness to help figure out the cause. Then it's up to your teams to solve them.

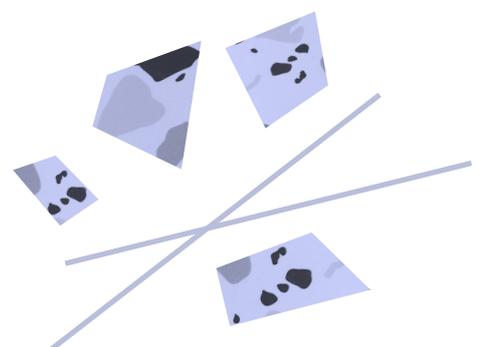
If anything, being agile will help problems to surface more frequently than before. But this is a good thing – it gives people the chance to fix issues there and then, and ensure your business survives. Working iteratively and trying to deliver value sooner will expose underlying, systemic challenges, as well as disasters waiting in the wings.

Agile isn't just for some people – it's organization-wide

While some departments will take to agile with ease, it's not enough for isolated pockets of the organization to transform if the rest of the company doesn't come along for the ride. You will only realise minimal benefits if you aren't able to get others on board. Don't put up barriers, blaming individuals or lack of skills in less technical departments – it's much more likely that legacy practices, policies, and hierarchies are holding up adoption. If you don't transform the wider company culture, efforts made by quick converts will be meaningless.

Agile isn't easy

Many organizations fail to become more agile because they fall at the first hurdle. Things are hard, or uncomfortable, or don't work as quickly as they hope they will, and so they throw in the towel. Heading into your transformation journey knowing it will challenge existing ideas and ways of working means you've won half the battle. Agile encourages trial and error, failing fast, and teasing out the nitty gritty for all to see. It's why there's a strong argument for working with coaches. Just because you're smart, it doesn't mean you couldn't do with the help. So set thoughts of smooth-sailing aside, roll up your sleeves, and prepare to get messy.



Are you agile?

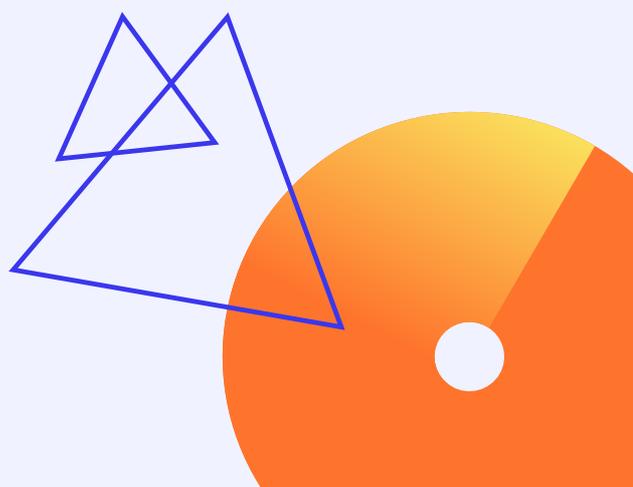
These are some common agile stumbling blocks that might be impacting your transformation. If you hear the following statements, or something similar, in your organization, it might be time to reassess your approach.

'We already do agile, don't we?'

You might have agile teams in place but, on closer inspection, your people don't have the tools or support they need to succeed. Without continuous integration and continuous deployment (CI/CD) and automation, your teams won't be able to deliver small pieces of value to customers. And if no one teaches people how to structure their work as customer outcomes, they'll keep doing what they've always done.

'Agile is just at the team level.'

Agile exists in a silo – restricted to software delivery teams with no understanding that it impacts all levels of an organization. Engagement is typically low at the team level, as people have usually been told how to work and what to build. They have much less autonomy than they used to.

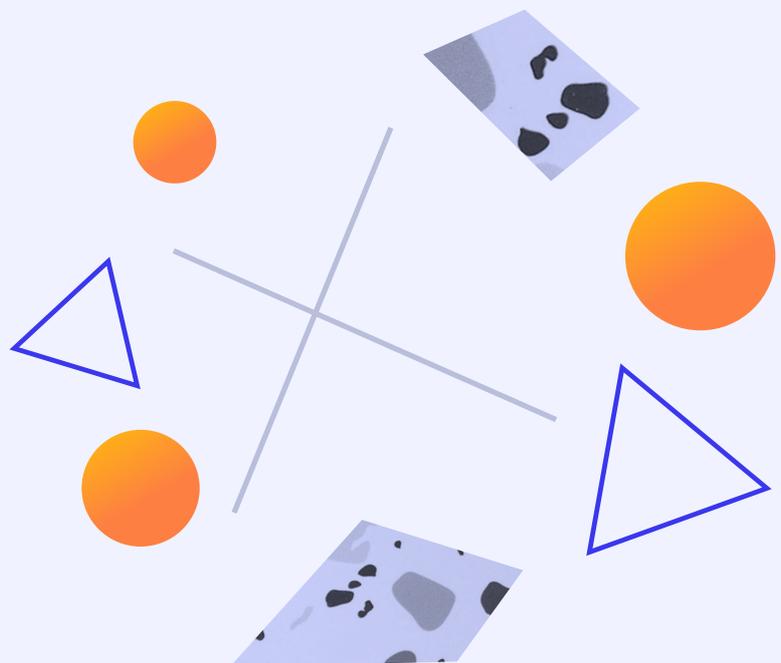


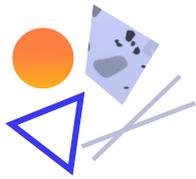
'We don't have support from leadership.'

Are you fully invested? Or are senior leaders refusing to support autonomous teams and viewing Scrum as a rote process to follow, rather than a dynamic holistic approach. Perhaps you believe it doesn't need to involve customers or feedback at all.

'We can't try new things out.'

You have some agile practices in place, but your organization's culture means teams can't try out other helpful practices such as behaviour- or test-driven development. This curtails any real change in the way they work.





Meet the four facets of agility

The ultimate goal of agile is to embrace learning or become, as Robert Kegan coined it in his book *An Everyone Culture*, a 'deliberately developmental organization'. In order for agile to truly permeate your organisation, addressing processes and tools won't be enough. You must analyze and adapt your people, culture, and your own style of leadership too.

One of the biggest challenges when it comes to helping organizations transform is lack of buy-in from the top. As a senior leader, your understanding, support, and propulsion are vital. It's not enough that a few individuals, a dedicated team, or a whole department are hardcore converts if the people making big picture decisions and, ultimately, holding the pursestrings are not onboard too.

Hopefully you already know agile is the way forward and are determined to see it embraced and proliferated across the organization, but let's take a look at some of the thinking that holds leadership back:

1. Big picture paralysis

Do you have a very narrow view of what agile transformation is? Is it based on what consultants have told you (or the bullet point from your lengthy reading list)? If so, you need to get stuck in and understand how it is not merely a process change at the team level but a re-imagining of your entire business operating system.

2. Not understanding the problem

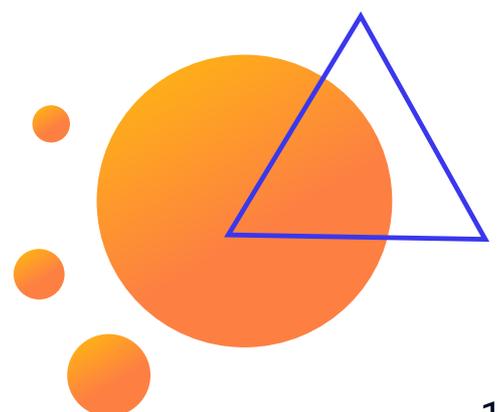
Do you see 'agile' as the goal or solution itself? It's important to gain a deep understanding of the problems 'agile' will solve for your business and create custom business goals unique to your organization.

3. Limited hands-on experience

If you're in the dark when it comes to how agile works in practice, that's probably because you have little to zero hands-on experience. By taking the time to learn how impactful it can be at all levels of an organization, or even what Scrum Masters and Agile Coaches do, you can begin to become as invested in agile as your teams.

4. The need for certainty

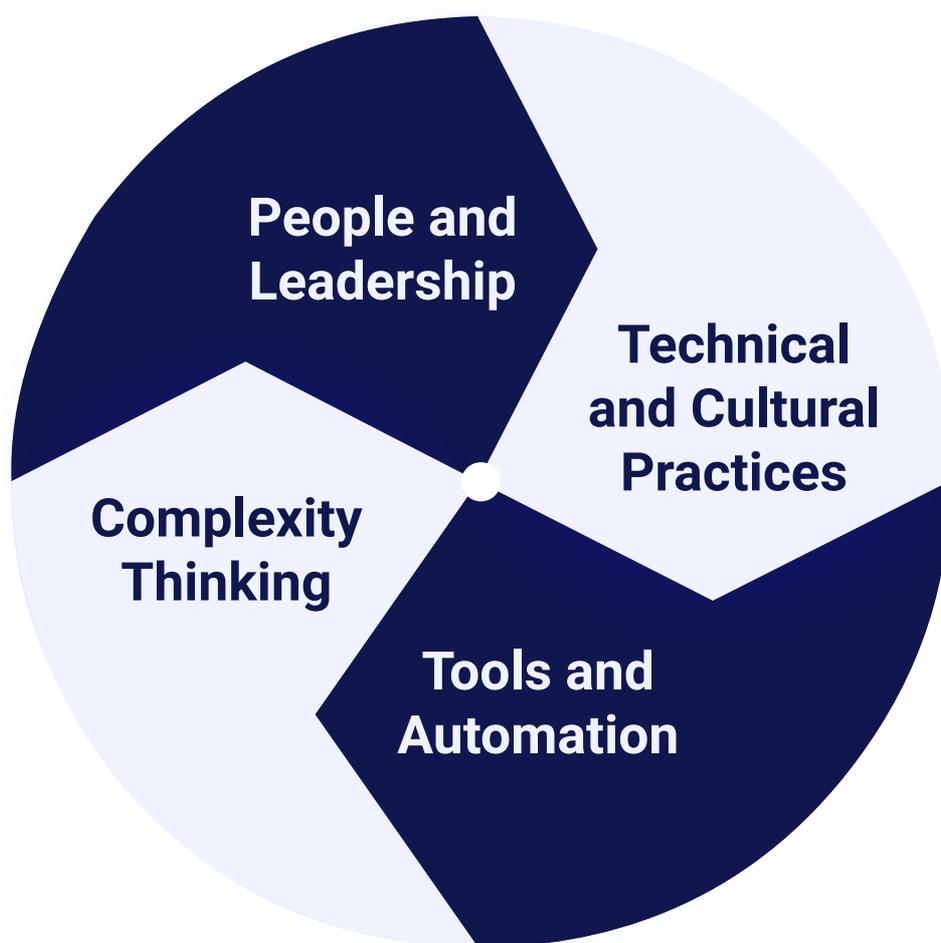
The unknown can be scary – humans have a propensity for wanting to feel sure of what comes next, to control the uncontrollable and plan ahead for everything. If this sounds like you, the trick is to relinquish this control, to welcome the unknown, decide to enjoy the journey, and to know that we can be certain of nothing except our own mindset (and some next steps).



A four-sided foundation

Even once you're a fully fledged agile convert, your acceleration could be far from smooth. Because the success of agile does not rest on one singular component, such as enthusiastic leadership. Only by recognising the necessary nature of a holistic approach can organizations hope their transformation takes flight.

To help bridge these gaps, communicate the key aspects of agile more effectively, and avoid the pitfalls that cause so many transformations to fail, consider the **Four Facets of Agility**: complexity thinking, people and leadership, technical and cultural practices, and tools and automation.



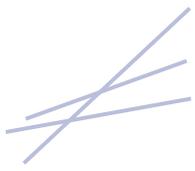
The Four Facets of Agility

This approach has helped a number of organizations to reframe their thinking and find out where they need to invest time, money, and staff. For example, maybe they've got all the tools at their disposal, but their teams need coaching to work more cohesively together. Or maybe the right practices are all in place, but leadership still wants fixed dates and plans for the next year.

Like any other, this model isn't perfect – it doesn't cover every single aspect of an agile transition, and it's not intended to exclude other patterns or practices. But the four facets have resonated with our customers and consultants alike, and might help put things into perspective for your organization too. The four facets are meant to be a foundation for action, to help guide particular areas of your transformation.



Now let's explore each facet in turn, giving examples of how we approach them, explain how they enable transformation success, and offer some practical tips for how you can put them into action.



Part one: Complex thinking, simply put

When it comes to product proliferation, managerial habits, and even our personal lives, we're constantly being told to simplify, simplify, simplify. But to solve the complicated and changeable challenges facing businesses today, complexity is key.

We live with uncertainty. Volatility and ambiguity are rife, and organizations face ongoing obstacles, such as non-linear relationships, and emergent ones too – the Covid-19 pandemic being the most palpable example. These challenges demand a bold and dynamic mindset.

The need for complexity

It's impossible for businesses to always plan ahead. Instead, they must expect and embrace change, especially through input and feedback from teams and customers. This is what complexity thinking frameworks – such as SAFe, Scrum@Scale, LeSS, and OODA loop, to name a few – are designed for. You might hear these being referred to as 'processes' elsewhere, but we prefer 'complexity thinking'. It better encompasses the mindset shift required to face the unpredictable world in which we find ourselves.

Complexity thinking isn't just the framework you or your teams put in place; it's also making sure creativity, critical thought, and logic permeate your workplace. Your organization needs people capable of smart, sophisticated thinking to understand concepts, generate original ideas, and use reasoned approaches to address the problems it faces.

These strengths mean your people will be able to draw connections between information from different sources, construct and defend arguments based on facts, take risks with new ideas, and use their imagination to problem-solve. This kind of thinking is vital and can serve your organization well.

Asking the right questions

When clients approach us, they've often already identified frameworks that they're keen to use without really knowing how they've come to that decision. The framework you or your teams choose should result from understanding what is trying to be achieved and why. If you're in a similar boat, try asking yourself these questions:

Where do I start?

Think about what's prefaced the decision to explore agility or take your transformation to the next level. Does your organization have the expertise and resources in house to bring about change? Or would you benefit from agile mentoring and leadership coaching from external providers?

What challenges are you currently experiencing?

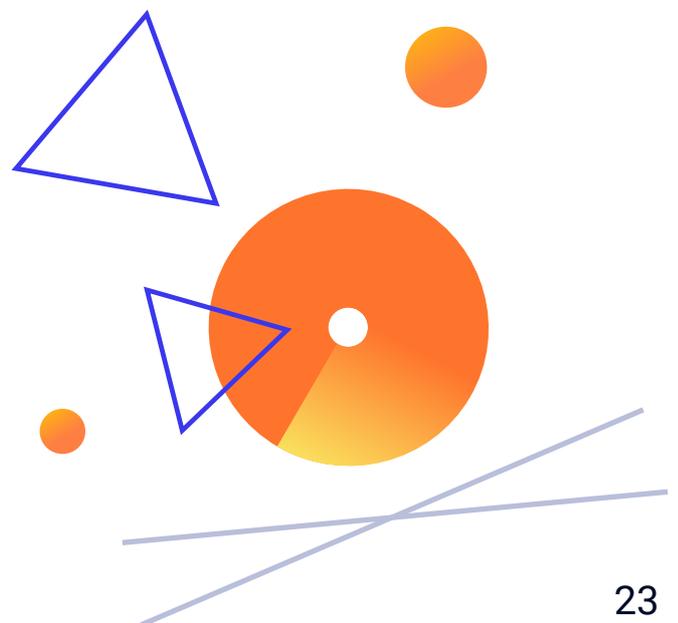
Knowing what isn't working and how a particular agile solution might be able to solve it is key. Ways of working should be put in place that seek to address the problems you're facing now. Identifying what those problems are is half the battle.

What goals are you trying to achieve?

Surviving and thriving are a given, but what deeper more specific goals do you and people within your organization have – as individuals, on a team level, and as a company? This will help inform the next steps you might want to take.

Why this framework?

Different frameworks are designed to solve different problems – don't get hung up on trying to shoehorn something into your organization that's never going to do what you need it to do. Be open to all the options. The way to find the best framework is by knowing what problems it might solve for you.





Part two: Nurturing an agile mindset

A workplace is nothing without its people. But because people like shiny things and fancy tech, when it comes to agile transformation, most organizations throw all their resources behind processes and tools, and forget about the humans required to make everything happen. They fail to consider that leaders and teams alike need an agile mindset, and as a result, their agility will always be half-baked.

What exactly is an 'agile mindset'?

This term isn't new – we've been talking about its importance for years. But some people are still in the dark when trying to define it. As recently as May 2020, Forbes contributor and author of *The Age of Agile*, Steve Denning, listed a number of sometimes contradictory definitions and was left asking, 'Is it possible that we in the agile community have been putting too much emphasis on such an undefined and ambiguous term?' We think the answer is a resounding, 'No!'

In her [2016 TED Talk](#), writer Julia Galef examined the differences between a soldier and scout mindset in relation to good judgement. The former is rooted in trying to make some ideas win over others, motivated by one's own preferences or biases. The latter chooses to seek out an accurate picture of reality, even when it's unpleasant or inconvenient. An agile mindset has a lot in common with a scout

mindset. Thinking with agility means not just doing what's comfortable or easy, or what we've been conditioned to do, but working through difficulties, challenging the way things are done, and facing problems head-on.

When it comes to the complex challenges businesses face, rewiring our thinking can't be emphasised enough. Conventional thought processes are severely lacking. Our research shows that when the complexity in our environment exceeds the complexity of our psychology, problems inevitably arise. From high anxiety that hampers performance to low morale and feeling unfulfilled, leaving agile processes unmanned by agile minds is angst waiting to happen.

Constructing new meaning

It's simply not enough to put people through their paces with a Scrum or SAFe class and assume their journey to an agile mindset is complete. Agile is not something you 'get' in an instant, whether or not you have the certificate to prove it. Don't just take our word for it. We asked psychologists to help us better understand how humans make meaning of the world and whether we're able to shift our thinking.

It turns out, as we develop in adulthood, we move through different identifiable stages of psychological development. The [Leadership Development Framework](#) (LDF) is a tool built on the premise that we all have the potential for continued growth. Through motivation and support, we can find new ways to problem-solve, interpret, and interact with our environment.

There are two broad strokes of adult development and meaning-making: conventional and post-conventional. We've been using this framework to help understand how our clients are likely to interpret situations and, as a result, how they will act.

Conventional

This is what most people fall into. It's the frame of mind concerned with control, measurement, and prediction. If you believe you can disassemble something into its component parts and put it back together again, you probably fall into this camp. You pursue goals in terms of output; you feel like you can control situations and it's possible to predict outcomes.

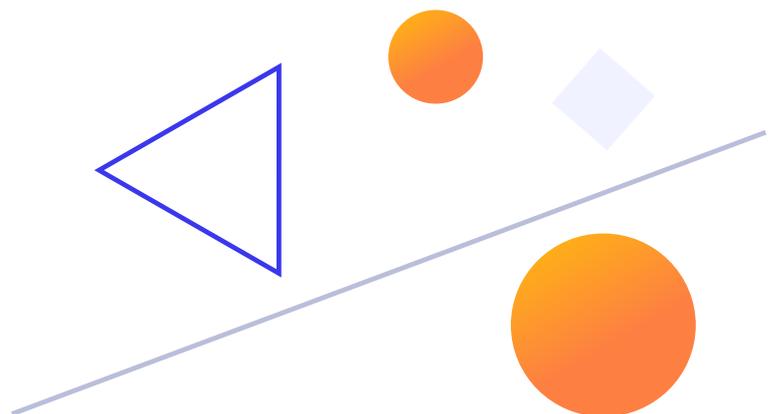
Post-conventional

Post-conventional thinking is much rarer – only about 10 percent of the population achieve it. With a post-conventional mindset, you understand situations are complex and interact in many different ways, realising you cannot simply disassemble and reassemble. Post-conventional thinkers see the world as a living system and most situations as impossible to predict and control. And, even more, they recognize their own assumptions about the world.

Take a guess at which of these worldviews suits an agile mindset? Post-conventional thinking is vital for successful agile leadership. But to help C-suiters like you to develop, you need to know where you're at and how to move forward. We use dialogue partnering, a collaborative inquiry into how people make meaning, to support business leaders on their journey to a post-conventional point of view.

'Dialogue partnering is something we've developed to help people shift their mindset to a more post-conventional, or agile, mindset. We consider that they might have hidden commitments to not changing the way they think and work. We work with them in a series of sessions to uncover their mindset and how they might think differently about their thoughts and actions.'

John Turley, Agile Consultant

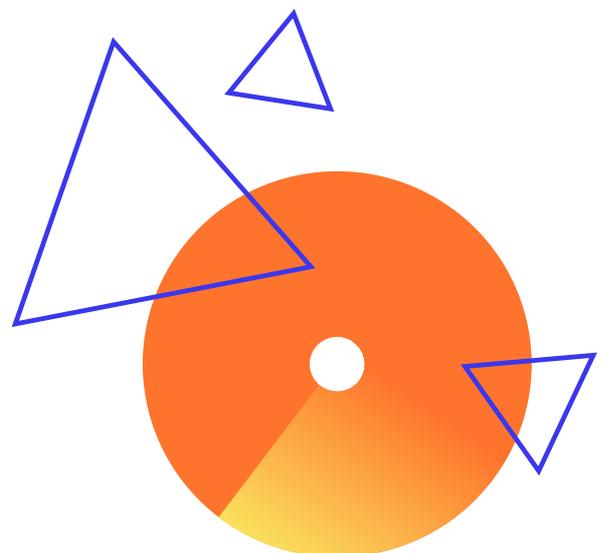




Part three: Creating a culture for new processes to thrive

At first glance, technical and cultural practices might feel like two disparate ideas awkwardly bumping up alongside each other. What does technology have to do with workplace culture? The answer is everything. Of course delivering valuable software requires good engineering, but that can't exist in isolation – it depends on cultural practices that allow it to thrive.

Meanwhile, culture can't exist without people. While people and leadership covers the evolution of an agile mindset, this is about making a cultural sea change through technology adoption and team behaviour – people can't embrace agile if their actions are stifled by the organization itself. We will point out some common practices here and explain how to achieve a cultural shift.



' I like to simplify agile to a couple of key concepts. The first is all about reducing the “gap in time” between taking an action and getting feedback. Be it showing rough mockups to customers, writing automated acceptance tests to drive development (aka, Behavior-driven Development), or employing test-driven development (TDD) and starting with failing tests. And it is equally important to have a large amount of humility. You might be better off treating more things as “hypotheses” rather than certainties. And then go about to find the easiest, least costly way to test those hypotheses. '

Jon Kern, co-author of the *Agile Manifesto* and Agile Consultant

The cultural conundrum

Some of agile's key technical practices run contrary to the way many people have worked for years, so they might feel counterintuitive. Let's take software development as an example, which can be rooted in traditional ways of working and siloed thinking. It's easy to see how extreme programming (XP), with its core list of interconnected practices, including collective code ownership, regular refactoring, and pair programming, could feel alien or uncomfortable to some. XP sees software development as a team sport relying on face-to-face

discussion and constant feedback. It challenges traditional development practices, recognising that if the whole team works in collaboration on something, it will be done faster and better than if it's left to a lone wolf.

Let's take a closer look at some of these agile-associated practices, and the culture shift required to make them meaningful.

Test-driven development (TDD)

What it is: Most teams write code first and then write test code after, or not at all. TDD, on the other hand, humbly starts with writing some expected outcome (i.e., a test). In TDD, coding, testing, and micro design are tightly interwoven. This level of unit testing leads to lower defect rates, reduced cost of new features, and a higher degree of technical quality in the code.

Developers run the tests, and the test suite is run in an automated fashion as part of the deployment pipeline process.

What it needs: A culture where quality and the long view are valued. If your organization only cares about speed, why would programmers spend time on TDD, where quality is encouraged? Valuing quality will contribute to a more effective organization – time spent fixing defects after they are in the wild is said to be up to **100 times more expensive to fix**.

How it benefits: TDD reduces the cost of change and increases the team's ability to fearlessly develop new code and refactor existing code, acting as a safety net. It also reduces the likelihood of introducing new defects, and the tests serve as living documentation for the code.

'Agility in software development requires a holistic approach. To deliver value consistently and predictably you have to get a lot of little things "right". Firstly, you'll need a strong understanding of the problem domain and the user's context, and then learn to build just enough (or slightly less, I like to say), get feedback ASAP, and apply a recursive, iterative approach. Mix in numerous technical practices with your lightweight process, automate as much as practical in a CI/CD pipeline, and you really can build a foundation that will serve your needs now and into the future. Rub a lot of "humility" liberally into the above list, and you'll be set for the long haul.'

Jon Kern, co-author of the *Agile Manifesto* and Agile Consultant



Behavior-driven development (BDD)

What it is: While TDD tests the inner working of the code, BDD is more akin to user acceptance testing. BDD attempts to bridge the gulf between the business and the development team. One of the more important aspects of being a high-performing agile team is to understand what the 'definition of done' looks like. The BDD approach uses a simple language format (technically a domain-specific language) to ensure development is aligned with business outcomes.

BDD encourages collaboration across roles, from quality assurance to business stakeholders to developers. It ensures that application behaviours directly contribute to achieving those outcomes and provides a precise method of communication between the wider team, using simple notations that are accessible to everyone. It encourages teams to use conversation and concrete examples to arrive at – and document – their shared understanding of how the application should behave.

BDD also replaces the need for mundane manual user testing of the basic functionality of the application. Now traditional QA/testers can focus on truly valuable testing (for example, exploratory testing and user experience). Developers run the tests, and the test suite is run in an automated fashion as part of the deployment pipeline process.

What it needs: A culture where cross-functional teams are encouraged to communicate. This set of practices is focused around having better conversations about how the system behaves and delivers value, so it's vital that team members have the autonomy to do this.

How it benefits: Explicit language constructs increase shared understanding and reduce mistakes caused by misinterpretations. BDD also reduces the need for documentation to be duplicated and tailored for different roles because the language is both clear and useful for both a non-technical and a technical audience.

Pair programming

What it is: This is where two programmers share a single screen, keyboard, and mouse. One 'drives' with the keyboard, while the other 'navigates' the current task being worked on, switching roles every few minutes or so. This teamwork approach leads to improved code quality as the pair 'program out loud', articulating challenges and solutions as they go. Knowledge is diffused and shared, so that it is easy for any team member to work within any part of the code base. Though not the primary purpose, there is a side effect where less experienced developers can upskill more quickly on the code base.

What it needs: A culture that doesn't encourage one-person-one-project thinking or code ownership, where teams have complete ownership of their work, and which recognises the benefits of real teamwork. If programmers are rewarded for finishing tasks individually, then there's no incentive for them to pair up with someone else. It also needs management that doesn't ask: 'Why should I allow two people to deliver half the work?'

How it benefits: There is less need for code reviews, which are a common bottleneck in delivering value. Therefore, on all but the most risky features, pairing means much less time is required before being able to deploy the code. Though peer reviews usually don't take a long time, often this activity is de-prioritized by the team, adding significant delays to the overall process. Improved quality is another big benefit, as is increased empathy and understanding in the team overall.

Mobbing/Swarming

What it is: Mobbing or swarming takes pair programming even further, where a handful of people work on the same thing at once, in one room, using the same computer. It is not uncommon to mix in some mobbing when the need calls for it. Some team mobbing can also collaborate on defining stories, designing, testing, and deployment.

What it needs: A culture that encourages open communication, goal alignment, collaboration, continuous code review, and a self-organizing team structure. While unconventional and uncomfortable for some, this method removes the need for peer reviews and separate testing, ensuring knowledge is shared for optimum customer value.

How it benefits: More people working together on a problem leads to better solutions. Mobbing means reduced defects, zero handoffs, and shared knowledge of everything across the team. And software can be delivered by a cohesive unit. By focusing on a goal together, instead of as individuals working separately, teams can achieve more.

User stories

What it is: A user story is an informal explanation (at least compared to standard 'requirements') of a software feature written with an end user's perspective. Its purpose is to define a bit of client-valued functionality.

What it needs: A culture where end-users are the focus of every conversation, in all parts of the company. All work should centre around adding value for the customer. This could challenge the current culture of valuing silos, busy work, and task completion over delivering something that has a real customer impact. If the culture sidelines the customer's needs, then why would product owners and developers write user stories (or care about customers) at all?

How it benefits: User stories act as a focusing mechanism and provide the 'why' for what developers are doing. It means fewer misunderstandings about the end goal or value, which can cause re-work when what was delivered doesn't meet expectations.

Radical thinking and untrusting leadership

Some leaders find this thinking and these practices radical. In fact, a VP once told us that if they allowed a tester to pair upfront with a developer, the developer would find a way to sneakily code around the tester. When, in fact, we find that pairing a developer with a tester before starting to develop the user story results in huge decreases in defects. Both people have an understanding of how the feature will be tested and how they're planning to approach the work.

We can blame ignorance to an extent. Sometimes leadership just doesn't know that teams should be doing these practices. But where they do recognise the benefits, more often than not, they assume programmers 'learn on the job'. We have rarely seen that to be true, especially in environments where there's immense pressure to deliver. People don't have time to research these new ways of doing things and get themselves up to speed without expert assistance. Organizations must view software development holistically. If they don't allow time for on-the-job training and learning, and openly support their people to do this, teams will fall back on the familiar and the organization could fail to realise all the benefits these ways of working could bring.

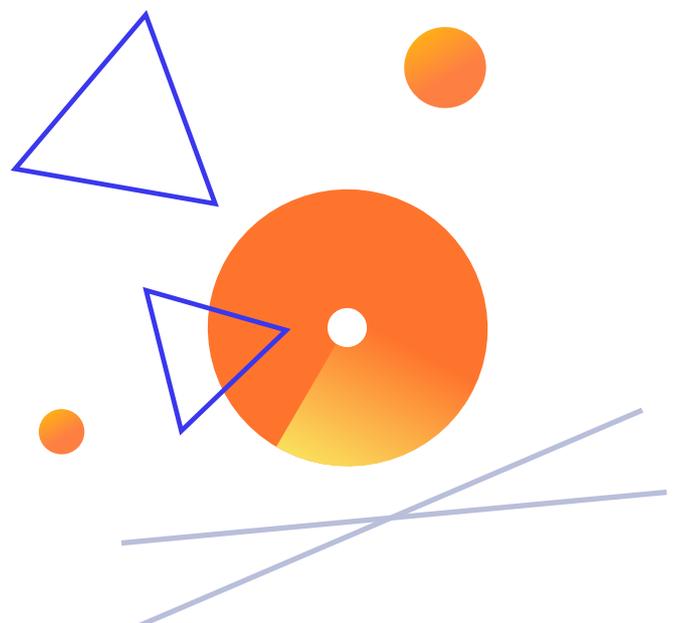
Go team!

Many technical and cultural practices relate specifically to teamwork. 'Team science', espoused by the likes of Richard Hackman, Patrick Lencioni, and Amy Edmondson, and organizations like Google, centers on the idea that a great team will always outperform a group of random people working on the same project without a team approach. Surprisingly, this is brand-new information to some organizations, who choose short-lived teams and priority-focused reorganizations to dictate how people work together.

But naming it doesn't make it so: it's not enough to gather a group, call them a team, and leave them to figure it out on their own. People need to learn how to operate in an emerging ecosystem built on complex situations and multiple interactions. Beyond this, they need to understand the factors that make their teams effective:

clarity of mission and purpose, boundaries, psychological safety, and diversity of thought.

Without team science in play, there are no guarantees people will allow diverse thinking to flourish, handle conflict appropriately, or even be able to make healthy decisions. Our appreciation for organizational culture and scientific approach to team strategy and implementation help our clients determine how effectively existing teams work together, and whether the conditions are in place to help them succeed.

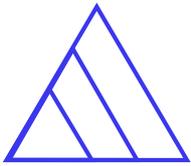


Take teamwork seriously

A practical checklist for team leaders throughout your organization to prioritize best practice:

- **What's the mission?** Does the team know why it exists? Does everyone know it? Without a team mission, people will be working seemingly at random, which won't be inspiring to anyone. Teams need a purpose that orients and engages them, as well as helping them feel committed to the objectives.
- **Is there structure and clarity?** Are goals, roles, processes, and execution plans clear for the whole team? Is the work small and measurable so that risks are mitigated? Do people feel a sense of accomplishment?
- **Do people feel safe and trusted?** Can team members take risks without feeling insecure or embarrassed? Can everyone count on each other to do high quality work and be conscientious?
- **Are they a true team?** A true team needs to work together to accomplish its work (not just hand off work), and is stable in membership long enough to accomplish something meaningful together. Does the team prioritize true teamwork? Do they have open conversations about how they work together?
- **What is the point?** What is the meaning of the work and what impact is the team having? Do team members believe that the work they're doing matters? Are they working on something that is personally important for everyone?





Part four: Take the tools off auto-pilot

The tools organizations choose to use should help them to achieve great things. They're there to provide visibility into the pipeline of product delivery so teams and companies can understand and improve based on what they're seeing. And automation is vital too, requiring strategy, a value stream, and a clearly defined road map. But both are implemented by people, who are prone to falling back on the familiar and deferring to hierarchical ways of working.

Organizations need to look at these components holistically, rather than as a reflection or replacement of manual processes. And they need to continuously learn about and improve the tools and automations they have in place.

Tooling around

[Conway's Law](#) states that we design systems that mirror our own communication structure. If the strategy and goals are not clear, there is no compelling need to collaborate, and people will optimize for their own silos. And tools are part of this! Without cross-company tools that foster clear, concise, and meaningful communication, the software an organization produces will lack that same cohesion and fail to deliver what customers need.

When tools aren't set up right, collaboration is negatively impacted. When we're called in to assess a tooling set-up, we often find project teams and multiple handoffs across different departments and a lack of team understanding about what or how to improve.

One team we worked with was handling a lot of incidents when code was pushed to production, and they frequently had to roll back releases. They had devised a 12-column Scrum board. It included columns not only for the development work in progress but also a peer code review, a senior code review, and three kinds of QA testing – but there were still a high number of production incidents. They decided they would hire a release manager to try to prevent things from being released in a bad state. Every time something went wrong, they added tools, processes, and people to manage and block, rather than fixing the underlying problem (a lack of automated testing). No surprise: that was not going to help them succeed.

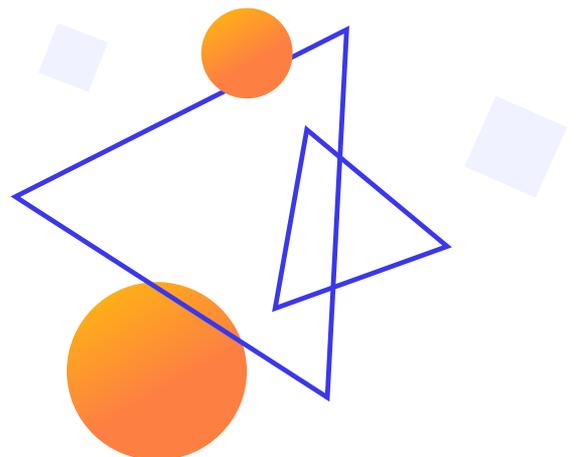
Measuring for success

Effective tools also provide the data and information organizations need to make fact-based, strategic decisions. Problem is, often people don't know what they should be measuring, how, or why. It's important to look beyond standard metrics and focus on 'flow metrics' – measuring the movement of value right to the customer – assessing cycle time, throughput, and time in status. Also consider how your organization can integrate feedback loops into its delivery metrics – if teams have a rigid, fully allocated quarterly plan – that doesn't include time for customer input. Only then can you make sure you're continually delivering the most effective solutions.

Scaling up your tooling strategy

Teams need a holistic view, stated goals, and clear roadmap for tooling solutions to help realise their potential. Even with the best agile intentions, poor tooling will curtail any noticeable benefits. Are your software products more complex than the development tools you're using? Have you adopted SAFe and other scaled agile frameworks but can't coordinate your value streams? Or are you struggling to marry up the work with your strategy or value delivery?

Good tools offer the benefits of agile development – faster time to market, fewer failed initiatives, better collaboration, and transparency into enterprise initiatives. Beyond that, they provide insight outside the team level, offering visibility at scale.





Embrace agile with Aligned Agility

At Aligned Agility we help companies adopt and scale agile ways of working – from small start-ups embarking on rapid growth to 100-year-old organizations with over 200,000 employees around the world. We work with organizations who are willing to sit in the weeds for a bit, to work through the tough issues, to truly hear what employees think, embrace change, and work differently. And for these companies, the pay-off can be significant.

Our tried and tested approach involves workshops and hands-on guidance. And unlike other agile consultants, who might simply offer Scrum training, we get down to the nitty gritty of the day-to-day work, using concepts like value stream mapping to uncover how you deliver value for your customers and look to improve the speed and processes.

Finding the right fit for you

When it comes to businesses that help organizations transition to agile, the options are wide-ranging. Some agile consultants deliver their set of solutions out of a box and impose rigid practices upon people. That's not us.

We don't offer a cookie-cutter solution. Instead, we customize our approach to best suit the people we work with, involving them in the

decisions. After all, your people know the most about your own workplace challenges and needs – we bring in our expertise to question their assumptions, surface top challenges, and uncover high priority needs in your organization.

Helping voices be heard

We focus on engaging teams and cross-sections of people so that voices on the front line – staff typically closest to customers and the daily challenges with delivery – are heard. Engaging people in solving their challenges helps them feel ownership of the results, not that processes are being ‘done to them’, which is key to increased engagement.

And we don’t just work with software developers – we have clients who manufacture consumer electronics and pharmaceuticals. In fact, our consultants have experience nurturing agility in places as wide-ranging as television shows, legal departments, and roofing companies.

Our process is about experiencing new ways of getting the job done – in the context you’re most familiar with. Our workshops are safe spaces where we actively encourage everyone to speak up and share their opinions, however much they might differ. Don’t get us wrong, it’s a messy process at times, as team members come together to co-create meaning. But the result is always worth it.

Transformation before our eyes

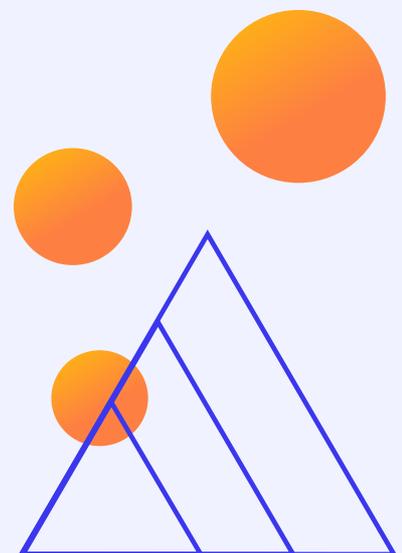
Over four remote sessions totalling 16 hours with a recent client, we were able to witness the team dynamic evolve right before our eyes – becoming more cohesive and open to each other's points of view. By the end, everyone was on the same page, excited to take their next agile steps. Approaching this whole process from the client's perspective is the best way we can transfer our complexity thinking knowledge and, frankly, work ourselves out of a job.

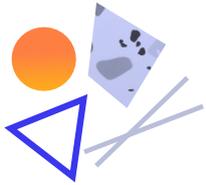
Our workshops and discovery sessions help the entire team identify next steps, instead of executives dictating them. Work is done incrementally, in a transparent way, with purpose and clearly defined outcomes. Input from everyone allows the right tools and processes to be introduced with maximum buy-in. And, as a result, culture improves. Arming teams with this wider knowledge promotes alignment and autonomy, increases productivity, raises engagement, and bakes in complexity thinking.

Coaching and community

We also offer embedded coaching, where we spend time with the teams and help them understand key principles and practices for agile ways of working in their day to day work. But we don't just add coaches – we focus on setting a strategy to make ourselves obsolete. We look for people who are ready and willing to grow into new roles as Scrum Masters and Agile Coaches. We find champions who understand the principles, and get them on board with the changes. These people often act as internal coaches when we have gone.

We recommend setting up learning events called 'communities of practice' where people can come together and talk about their shared challenges and learn from each other. These often take a while to gain momentum inside organizations, so we lead them initially until your people grasp the concept and are speaking up on their own. During these events, we can also offer mini training sessions.





Unlocking the Four Facets of Agility

The Four Facets of Agility aim to bridge the knowledge gap between leaders like you and the teams trying to implement agile across your organization, helping create alignment on where you stand versus where you want to be.

Rather than agile being an empty word, naming these four aspects – complexity thinking, people and leadership, technical and cultural practices, and tools and automation – can help you think differently about agile. Now you can have more meaningful, productive conversations about the needs and challenges faced by your organization.

The ultimate goal of agile is to become a ‘deliberately developmental organization’ – embracing learning, continuously improving processes and tools, and analysing and adapting your culture and thinking as you go. As a result, you can expect improved well-being, an increase in team and organizational performance, and a workplace where it’s not just about getting the job done; it’s about caring *how* it gets done.

Ready to face the Four Facets head on?

From developing an agile mindset to ensuring you have the right tools for the job, rewiring the way you work is complex and continuous.

Wherever you are on your agile transformation journey, Aligned Agility's experts are here to help.

[Get in touch today](#)



Too many agile consultations end with reams of notes and no action – we’re here to change that.

Many organizations think an enterprise agile tool will solve all their problems. Unfortunately when it comes to tooling, there is no silver bullet. However, we can show you how incremental process and mindset changes combined with the right tools become a golden spear to delighting your customers.

We’re not focused on one tool, nor tied to one framework. Instead we empower your teams to think differently about how they work to drive lasting change and tangible business results.

Learn more or get in touch:

AlignedAgility.com

 Jira Align

 Jira

Aha!

 Gtmhub

 **monday.com**

