Your Agile Blind Spot Guide

Easy Agile | Gil Broza

Thanks for downloading a copy of the 'Your Agile Blind Spot' guide!

This 7-part guide is brought to you by Agile thought leader Gil Broza and reveals the greatest and often invisible weaknesses of Agile implementations and what you can do to fix things.

This guide will help your team and organization achieve meaningful agility: a good flow of valuable solutions to real customer problems; drama-free, affordable adaptation; and a healthy, collaborative, engaged human culture. As you'll see, getting there is rarely a matter of process or of trying long enough!

This guide will take you through 7 critical conditions that are often misapplied or misunderstood, but are essential to achieving meaningful agility.

Each of the 7 sections in this guide contains a pure-protein, three-minute, explanation of the implications of not fulfilling the condition properly, what needs to change, specific actions you can take, and their expected benefits.

At the end of each section, there is also optional resources such as podcasts, further readings, interviews with experts, and more designed to help you continue on your journey to achieving meaningful agility.

About Gil

Gil Broza helps tech leaders achieve meaningful agility in product development and supports their non-software colleagues in creating real business agility. Gil has helped over 100 organizations make Agile work well in their unique contexts by using his Right-Fit Agile system, which is based on the three pillars critical to agility: mindset, culture, and leadership.

About Easy Agile

Easy Agile tools help teams be Agile. We deliver powerful, native solutions that help teams deliver value and align with the needs of their customers. We help remote, co-located and hybrid teams focus on common goals to run effortless sprint planning, story mapping, backlog refinement, roadmapping, PI Planning, and more. Trusted by more than 100,000 Agile teams across the world.

Easy Agile TeamRhythm

Easy Agile TeamRhythm's highly-visual story map format transforms the flat Jira backlog into a meaningful picture of work. This makes sprint or version planning, backlog refinement, and user story mapping much easier.

Easy Agile Programs

Make scaled cross-team planning and execution easy with a complete PI Planning solution seamlessly integrated with Jira.

Empower collocated, hybrid or remote Agile teams to plan but also deliver value at scale without leaving Jira.

Easy Agile Roadmaps

The simplest and most flexible roadmapping tool for Jira.

Easy Agile Roadmaps enables teams to align around the vision for a product and how they'll sequence features for delivery to customers.

Easy Agile Personas

Easy Agile Personas enables product managers and Agile teams to capture their customer archetypes alongside their project and Agile board in Jira. Product Managers can identify which persona is related to a given user story, and the value the persona places on that user story.

Contents

PART 1

Mindset Alignment

Say your team, management, and stakeholders have rallied behind a common objective. The team uses Agile tactics — practices, processes, roles, and meetings. You're on track to deliver great business outcomes ... right?

However, are the daily standups (Scrums) really just boring status updates? Is the backlog essentially a project plan divided into two-week deadlines? Is the Scrum Master a task-master and process administrator? And what does the product owner really own?

You may check all the boxes on tactical, visible Agile, yet experience little agility. The team may even seem less productive because of all those meetings. And everybody gets frustrated because this Agile thing is supposed to make things better!

The reason this happens (everywhere!) is that it's not enough to align to objectives and tactics. Everyone involved needs to **approach work with the same mindset.**

Specifically, the team, management, and stakeholders must operate by a common set of **values** and agree with the same key **beliefs**. Values and beliefs are the fundamental drivers of people's choices and decisions as they engage each other in work. For example:

- Focusing on making a difference (vs. maximizing the team's workload)
- Keeping options open as late as possible
- Preferring collaboration over delegating to experts
- Testing and learning, with the humility that not all ideas are great

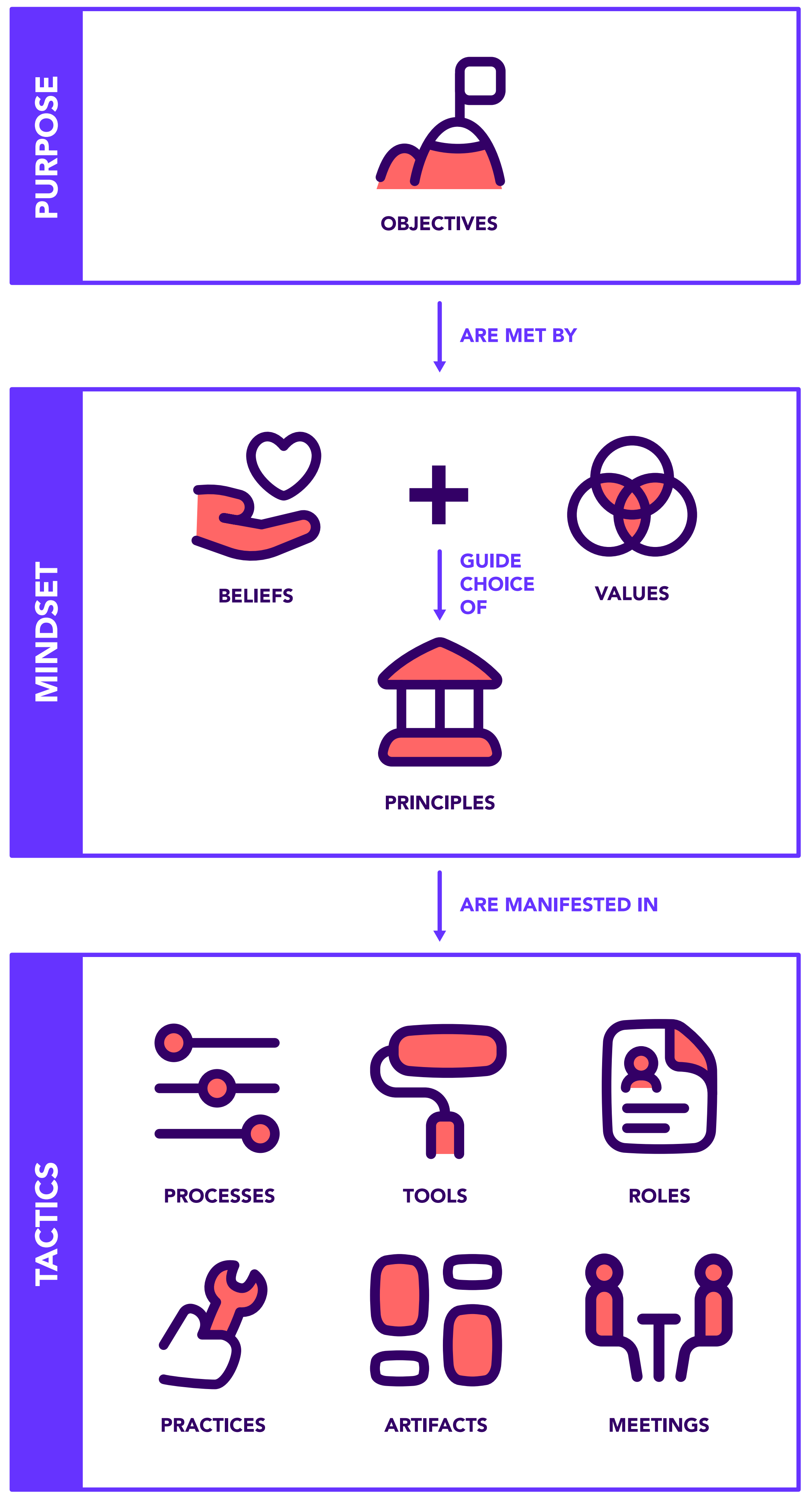
See how these abstract choices are not immediately apparent from popular tactics, such as sprints, velocity, and backlogs?

Everyone has different ideas, preferences, and experiences when it comes to doing work. Where one person seeks and applies feedback, another moves ahead without questions.

Where one believes in discovery, another worries about surprises. If they work on the same team, there will be conflict and frustration because everyone thinks their approach is better for accomplishing the mission.

What to do? Get everyone, all the way up to the sponsors, into a meeting early on. In that meeting, help folks to articulate their beliefs about the work, to agree on a shared narrative, and to identify the few values most likely to achieve the objectives given the beliefs. And finally, despite personal concern or discomfort, to mutually commit to abide by those choices.

This simple process drives intentional, shared alignment to an explicit mindset. For the investment of only a few hours, you'll maximize the team's chance of succeeding together.



PART 2

Agile Leadership

Teams and organizations depend on people-first leadership for real agility. Reflect for a moment: How would you characterize the approach to leadership in your organization?

Over the years, a certain model of Agile leadership has evolved. It is based on the servant leadership approach, which realizes that **it's complex human beings (not controllable "resources") who produce business results.**

Several beliefs lie at the heart of servant leadership:

- People's engagement is voluntary, and they need some autonomy.
- People need purpose, which should be meaningful to them.
- People take better action on ideas of their own.
- Those closest to the work should make most of the decisions.

When leaders create environments and systems that respect these beliefs, people can do their best work in service of the organization. This thinking is meant to be pragmatic, not touchy-feely.

I work with many people who find these ideas compelling but challenging to implement for two main reasons.

First, there's plain old change resistance. Managers got where they are by managing a certain way. It's what they know and what their organizations reward, while servant leadership (though already a 40+ year-old concept) seems like an unproven gamble with a weird name. Many managers remain accountable for results, so they're afraid to relinquish control. Others (and I see this often) don't truly trust their teams enough.

And second, 99.99% of organizations will never be anywhere near flat, and small teams can only know and do so much within the bigger context.

While organizations have indeed been allowing teams greater control of their work, many have landed on unhealthy and ineffectual leadership patterns. Do you see any of the following in your team or organization?

- Team-level leaders, such as Scrum Masters, act more like servants than leaders. Or, they lead in a bubble, while the more-powerful organization around them does not employ Agile leadership.
- Managers allow folks to choose their tasks from a sprint backlog, but don't empower them further.
- Senior managers allow teams to use Agile practices, while saying, "I don't care how you execute as long as you deliver on the plan."

Avoid or overcome these patterns by adopting a richer view of Agile leadership. Consider these additional Agile beliefs:

- People won't bring their best selves to work if they don't feel safe to do so.
- Teams should have autonomy — with direction.
- Self-direction does not imply smooth flow.
- A team may not evolve to greatness without help.
- Over-optimizing for a team can sub-optimize for the organization.

Put another way: don't just think about a group of individuals performing work. **Think of leadership as enabling an entire complex human system inside a bigger complex context.**

This expanded perspective will help you rethink roles and the needs fulfilled by leadership. It reframes team-level leadership as enabling real teamwork, among willing participants, connected to the larger dynamic and purpose. It focuses mid-level and senior leaders on enabling larger groups' culture and systems. These leaders are definitely not servants, but **leaders who focus on others before themselves and serve the greater whole.**

PART 3

Psychological Safety

The team is busy, work products are moving along, and productivity metrics look good. However:

- Are planning meetings generally lively, and every opinion has a chance to be heard respectfully?
- In retrospectives, do team members go beyond dry process matters and discuss personal concerns?
- If a team member is ill, do they feel comfortable taking a day off?
- In planning or review meetings, do team members sometimes push back on questionable ideas from managers or stakeholders?

If some of your answers fell short of "yes," your team has a psychological safety problem. **While folks work hard and attend all the meetings, they are holding back, not truly bringing their best selves to work.** They direct their efforts toward doing what they think they're supposed to do, rather than what they think is right.

Agile isn't about moving stories faster into the "done" column. It's about doing more of the right and less of the wrong. Knowing what's right usually requires multiple sources of information, different perspectives, healthy questioning, and tolerating unknowns and ambiguity.

The safer people feel, the more they'll participate in determining what's right. Conversely, **low levels of safety breed a "check the box" or order-taking mentality and thereby low agility.**

The matter of psychological safety is relatively new in management circles. Its roots are in physical labor, where unsafe working conditions can be deadly. Low psychological safety clearly has different effects in knowledge work, and it's also harder to detect. If people don't feel safe, they won't tell you!

To compound the challenge: you might honestly believe you've created a safe team environment, but members may feel differently. Often, that has to do with whether they perceive threats from other players in the organization.

Nevertheless, do your best to maximize safety. I recommend starting with a simple, powerful habit:

View every action you're about to take through the lens of "how could this reduce safety?"

The following examples are all based on well-meaning clients who unwittingly reduced safety:

- Do you share each team's "percent delivered vs. committed" metric in all-teams meetings? At one client, a team referred to this as public shaming.
- Do you attend stand-ups, even as a silent observer, and notice that all participants say that everything's fine? They might be censoring themselves in your presence.
- If a team member tells you that a plan is in jeopardy, do you respond in a way that could be perceived as personal blame? They might not bring such news to you again.
- Do you document or record the proceedings of retrospectives (beyond the agreed-upon action items)? People may think twice before saying something.

Psychological safety takes time to build and little effort to destroy. However, if you're visibly intentional about it, it will be relatively easy to recover from faux pas. If you aspire to great team agility, keep safety front and center.

PART 4

Real Teamwork

Your Agile team is cross-functional and folks have all the necessary skills. They attend meetings together and work on items from a single plan. If working in an office, they might sit in the same space. They're competent, engaged, and professional.

And still … **are they really a team, a single unit with a common purpose?** Or are they only a group of individuals doing related work?

To answer this, imagine all team meetings went away. Instead, members send the Product Owner (PO) their individual questions and estimates about backlog items. The PO and the Scrum Master (SM) determine the sprint plan and assign tasks. Instead of a daily standup and a retrospective, members send their input to the SM, who synthesizes and distributes a summary and action items. The PO/SM give demos to stakeholders.

Would this significantly hurt the outcomes that the team produces?

If your honest answer is "no," you have a group, not a team, and could be missing out on considerable potential.

That potential includes: tackling harder problems, producing better-fit solutions, stopping wrong work early, avoiding perfectionism, increasing responsiveness, and building resilience. And on the personal front: faster professional growth and greater engagement.

Another way to think about it is that in a group, performance is additive, and the potential exists for equating people with resources. On the other hand, the following three Agile beliefs express an expectation that **a real team is a performance multiplier:**

- Teams may be smarter and more productive than the sum of their members.
- Teams may make smarter decisions than their managers.
- Individuals will make mistakes; supervision isn't the best way to mitigate that.

Do you agree with these beliefs? Do all of them apply in your situation? Do all the members think the same way?

Many people subscribe to these beliefs and try to act on them by forming cross-functional teams and having frequent touchpoints to plan, review, and improve together. Yet, these tactics alone are not enough.

What's missing is the connective tissue between beliefs and tactics: principles. These principles tell team members and leaders how to engage with each other, both during team touchpoints and outside of them. In Agile, these principles are self-organization, collaboration, transparency, respect, psychological safety, and trust.

It will take intention, patience, and support to make these principles a way of life in your team, but the benefits are substantial:

- People who question, suggest, and explore options instead of acting as order-takers
- Daily standups that are collaborative micro-planning activities instead of person-by-person status reports
- Retrospectives that produce meaningful changes instead of a complaint-fest or a listing of obvious superficial improvements
- Members who can solve more problems instead of relying on ever more siloed experts
- A team that tackles challenges and impediments instead of just coping with them

PART 5

Outcome Thinking

In conversations and meetings, do your team members talk about story details, code, defects, impediments, velocity or cycle time, the next deployment? Of course they do.

However ... is that all they talk about?

If so, your team is **focused on output.** They're a human machine that turns user stories into deliverables.

The (all too common) risk is that your team spends time and goodwill on deliverables that don't solve users' real problems, or don't solve them usefully enough.

In other words, **what's missing is talk about outcomes,** for both the customers and your business. (Note: there might be discussions of value, but that is not enough; value is only a measure of the hoped-for outcome.)

Does the identification of outcomes, and the determination of outputs that will achieve them, happen upstream from the team? That might be okay; it might also be a lost opportunity to benefit from the team's brainpower.

The fundamental risk is in moving too fast from the problem space to the solution space.

See, Agile isn't about being fast; it's about doing more of the right and less of the wrong. "Right" means delivering better solutions to real problems, needs, and goals. "Wrong" is being busy with stuff that may not matter.

For example, take this guide. I didn't approach it as "produce seven short pieces of content on Agile challenges." I approached it as "Give Agile leaders, who think things are going fine, a view into common but rarely noticed areas of trouble." I hope I'm succeeding — otherwise, this was an expensive exercise in writing.

In an effective Agile environment, outcome thinking permeates everything: leading with purpose, allocating funds, managing portfolios, launching initiatives, sequencing deliverables, splitting stories, defining acceptance criteria, designing meetings, and more.

Clearly, defining and producing output has to be part of the process. But are you giving enough attention to outcomes? Take a good look at your processes and identify where the focus is more on outputs than on outcomes. What are the consequences? Where are they greatest? What would it take to change the process? Is that worth it? How will you know?

Make outcome thinking a habit for three important gains:

- First, your work will make a bigger difference.
- Second, you'll be able to deliver simpler solutions sooner, so you'll be more productive.
- And if you also break big work down into a sequence of mini-outcomes, you'll make frequent delivery more meaningful.

PART 6

Useful Feedback

Does your team regularly consume enough feedback in their Agile diet? As you know, feedback is a staple of good agility and quality products.

In my experience, **most Agile processes aren't designed for enough useful and timely feedback**. Even though teams discuss things daily, demo progress, and fix issues, they generally proceed as if they have the answers to all the important questions about the problem and solution. (Sometimes, teams act this confidently because more powerful people subtly expect them to.)

When people are certain that their ideas are correct, they generally value predictability and efficiency more than they value adaptation and learning. If their ideas turn out to be correct, efficiency was indeed the right call. But if their confidence turns out to be unwarranted, the losses -- at least in terms of sunk costs and opportunity costs -- may be substantial.

Traditional work management aims to achieve predictability and correctness through up-front requirements, analysis, and design. Agile aims to discover what's right through evolution, based on frequent communication, collaboration, and feedback.

And yet, can you recall an Agile project that delivered so-so results despite iterations and demos?

Putting this idea of evolution into practice can be quite difficult, but not for process reasons. As a professional, you're supposed to know what you're doing. You might feel unsafe to voice certain doubts or to change direction, especially if promises were made. Getting honest and usable feedback takes a lot of work, and you may not like what you hear. And, you can't ask the same people for feedback too many times.

No wonder people settle for minimal, superficial, or infrequent feedback. The risk is that they continue doing their work with unjustified certainty or unproven assumptions.

To mitigate the risk, you need to cultivate humility. Inspire more and more people to regularly check assumptions and validate choices. Specifically, inject "how do we know" questions into your interactions with them. How do we know that...:

- Our understanding of the business problem was (and remains) correct?
- Our intended solution is viable?
- People would like and use it?
- The feedback we've solicited is honest?
- We're spending our time well?

Ask these questions respectfully, helpfully, and safely. They will change the conversation!

PART 7

Affordable Change

You've been Agile for a while, and everything seems fine: the team frequently releases high-value, quality product updates. Yet out of sight, a growing problem affects the team's ability to work.

Have you noticed how everything an Agile team does is basically a change? They pursue new or different outcomes, replace solutions, add/change features, improve product efficiency or robustness, and correct mistakes.

Gradually, making such changes becomes more costly and less safe. Quality gets harder to maintain. Expectations and needs remain high, but the team struggles to meet them. They respond to mounting pressures by rushing. It's a vicious cycle. Actual agility diminishes.

This situation has happened in most Agile implementations I'm aware of, despite empowered smart teams, prioritized backlogs, and retrospectives. Change becomes expensive and unsafe as a result of low Technical agility (TA): the team's approach to their technical work is not Agile enough.

Don't let TA stay low for too long. Fortunately, it's possible to spot low TA early. Think carefully about your situation; do you see any of the following signs?

- Instead of the design evolving to enable business goals, it grows haphazardly (corresponding to a patchwork of user stories).
- The product's complexity grows faster than the team can manage it.
- Developers don't keep their code clean enough as they rush to meet sprint deadlines and commitments.
- The code ought to work, and seems to work locally, but isn't proven safe to deploy.
- Automated test coverage (even if nominally high) is ineffectual, fragile, or not maintained.

You can't increase Technical Agility by training your developers, hiring more experienced ones, or leaning on them. Skill, buy-in, and accountability are rarely the reasons for low TA.

If your team has low TA, examine the sources of tension between three parties: development, product, and management. Are there unvoiced assumptions, unreal expectations, or lack of partnership? Does your process framework emphasize moving work along, but call little attention to agility in technical execution? Do people struggle to communicate their specific challenges and needs across role divides?

To establish and maintain high TA, first get the three parties to make an intentional, explicit, and mutual commitment to it: a social contract, as it were.

Next, make the commitment actionable, with strategies and tactics. For instance, expand planning conversations to consider likely future changes and the cost and safety associated with making them. Be strategic about taking on technical debt and actually pay it off. Notice when you need to make assumptions about the product's future and make them collaboratively.

The cost of change will never be zero, and the safety to make changes will never be 100%. But by minimizing the former and maximizing the latter, you'll make agility real and sustainable for the long term.

Your Agile Blind Spot, What’s next?

The guide started by defining **meaningful agility:** a good flow of valuable solutions to real customer problems; drama-free, affordable adaptation; and a healthy, collaborative, engaged human culture.

Gil Broza

Agile Mindset & Leadership Expert

All teams and organizations need to overcome many challenges and problems on the way to Agile. Many end up in a state that looks okay but actually falls short of meaningful agility -- and they don’t realize it. In this guide, you’ve learned about seven critical conditions that are usually in the blind spot:

1. Mindset alignment
2. Agile leadership
3. Psychological safety
4. Real teamwork
5. Outcome thinking
6. Useful feedback
7. Affordable change

The first step to creating change is awareness. Having read the guide, which matters struck you as “Yes, that’s our problem!” and which ones need deeper assessment?

Put these matters high up on your Agile journey backlog, and be prepared: **addressing them takes time, effort, and influence -- more than it took to adopt processes, practices, and tools.** Nevertheless, achieving meaningful agility is realistic, and not an all-or-nothing proposition: every small step will improve your culture, your way of working, and ultimately your results.

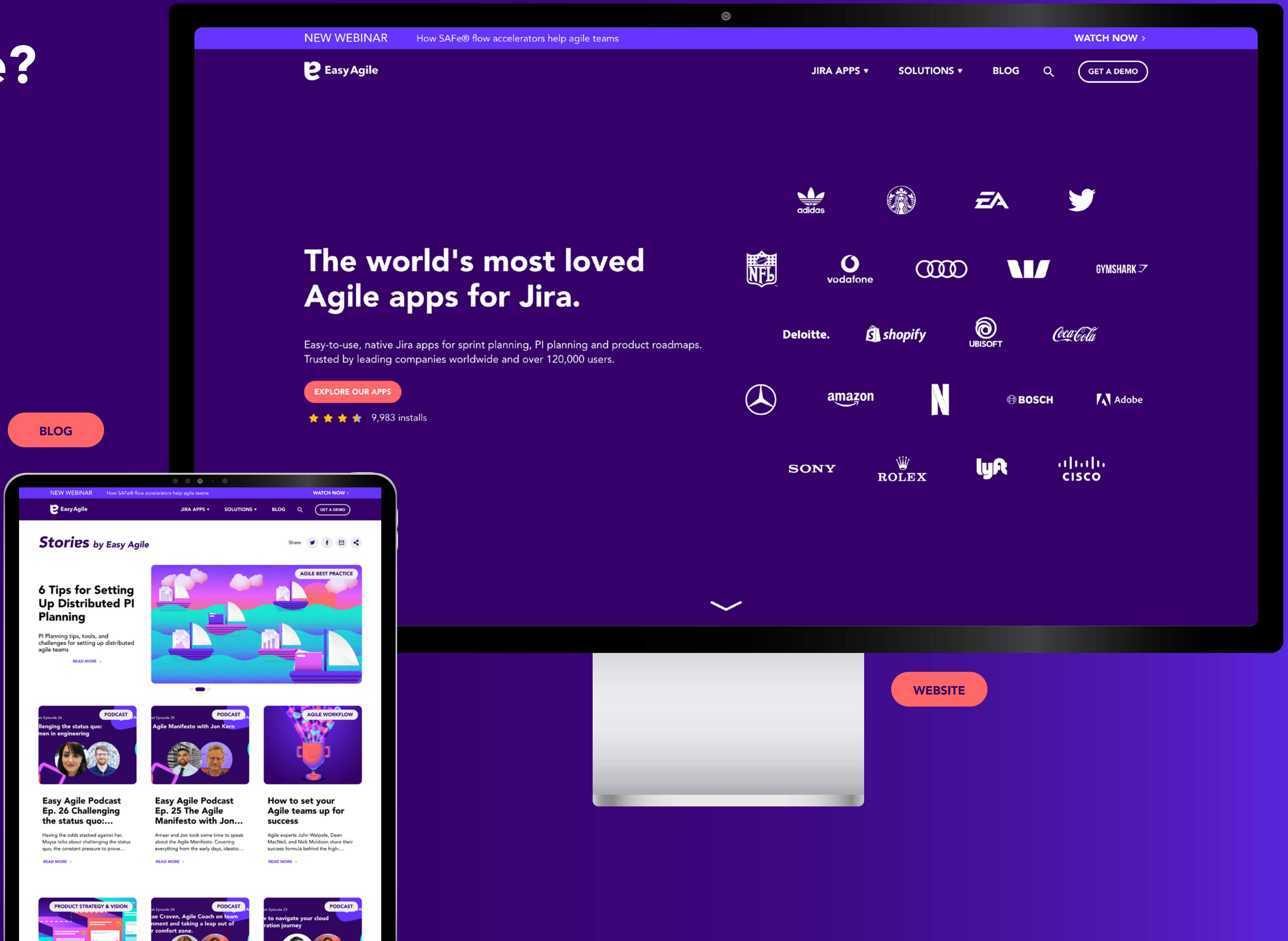
Did you enjoy this guide?
Stay in the loop!

For more Agile related content, be sure to subscribe to our blog and follow us on socials.

Use #easyagile to let us know what you thought of the 'Your Agile Blind Spot Guide' with Gil Broza.

BLOG

Youtube	**LinkedIn**	**Twitter**
@EasyAgile	Easy Agile	@EasyAgile



NEW WEBINAR How SAFe® flow accelerators help agile teams WATCH NOW ›

Easy Agile JIRA APPS ▾ SOLUTIONS ▾ BLOG GET A DEMO

The world's most loved Agile apps for Jira.

Easy-to-use, native Jira apps for sprint planning, PI planning and product roadmaps. Trusted by leading companies worldwide and over 120,000 users.

EXPLORE OUR APPS

★★★★ 9,983 installs

Stories by Easy Agile Share

AGILE BEST PRACTICE

6 Tips for Setting Up Distributed PI Planning

PI Planning tips, tools, and challenges for setting up distributed agile teams

READ MORE ›

PODCAST

Easy Agile Podcast Ep. 26 Challenging the status quo:...

Having the odds stacked against her, Maysa talks about challenging the status quo, the constant pressure to prove...

READ MORE ›

PODCAST

Easy Agile Podcast Ep. 25 The Agile Manifesto with Jon...

Amaar and Jon took some time to speak about the Agile Manifesto. Covering everything from the early days, ideatio...

READ MORE ›

AGILE WORKFLOW

How to set your Agile teams up for success

Agile experts John Walpole, Dean MacNeil, and Nick Muldoon share their success formula behind the high-...

READ MORE ›

PRODUCT STRATEGY & VISION

PODCAST

PODCAST

WEBSITE

Easy Agile | Gil Broza