

Agile with a capital ‘A’: A guide to the principles and pitfalls of agile development

February 2018

Understanding the true principles of agile can give companies the ability to work quickly, boost efficiency and product success, and, ultimately, create real, lasting value.

Agile isn't just for software nerds anymore. In this episode of the *McKinsey Podcast*, senior partner Hugo Sarrazin and partner Belkis Vasquez-McCall speak with Simon London about how the agile way of working has gone way beyond the world of software. Agile has become more prevalent among companies wishing to make swift, iterative teamwork central to their change efforts because—if done correctly—it can help people work more efficiently, delivering successful products and creating much more value.

Podcast transcript:

Simon London: Welcome to this edition of the *McKinsey Podcast*. I'm Simon London with McKinsey Publishing. Today we're going to be talking about agile, with a capital "A," which started as a set of principles and practices for developing software but is now being applied in many other areas of business. It's a world of scrums, stories, epics, and timeboxed iterations [see sidebar, "Glossary of agile terminology"]. These are concepts that are spreading well beyond the world of software. To help make sense of it all, we're joined today by Hugo Sarrazin—Hugo is a McKinsey senior partner based here in Silicon Valley—and Belkis Vasquez-McCall, who is a partner based in New Jersey. Hugo and Belkis, thanks very much for joining.

Hugo Sarrazin: It's our pleasure.

Belkis Vasquez-McCall: My pleasure to be on.

Simon London: If you don't mind, let's start with a little bit of history. I think that will help orient us. Hugo, just where and when did agile emerge?

GLOSSARY OF AGILE TERMINOLOGY

User story: A functional increments of work to do be done on a project, as divvied up by a team.

Epic: A large user story.

Scrum: A process framework used to manage product development and other knowledge work.

Timebox: A previously agreed upon period of time during which a person or a team works steadily toward completion of some goal.

Hugo Sarrazin: Software has been improving over many, many, many years. There are multiple versions, and agile is just the latest one. Before that, there was extreme programming, et cetera. There will be new versions that go beyond the current version of agile. It's part of the normal evolution.

The current version of agile, most people would date it back to 2001, when the Agile Manifesto was put together by a group of experts and colleagues who were looking for better ways to deliver software—and were frustrated with the inability to do things on time, on budget, and to delight customers. They wanted to see if there was a different way to think about the overall process and to think of it in the system way. Then they built on the great stuff that was done before and came up with a couple of principles. We'll cover some of them in a little while. These principles are helping us think very differently about how to deliver terrific products.

Belkis Vasquez-McCall: If you take the purest meaning of agile, which is how I always explain it to my children, agile is about being able to move quickly and easily. If you think about the fundamental principles of the Agile Manifesto, the core is that ability for speed and efficiency. When you think about when the Agile Manifesto was signed in 2001, it was a plea from the practitioners to shift how software development was built, to focus on the customers.

“Agile says, let’s break things into small little increments; let’s make sure we deliver value each time.”

Simon London: Do you want to say a couple of words about waterfall? Because as I understand it, and for people who are not steeped in software-development methodology, part of what agile is reacting against is what was known as the waterfall—the classic waterfall methodology for developing big software products. Belkis, just say a couple of words about waterfall, and how it works, and what the problems were with it.

Belkis Vasquez-McCall: The whole reason I fell in love with agile was after I completed my first waterfall project, which was a disaster. I was in desperate need of finding a new way of working. When you think about waterfall, it's based on a command and control approach,

pretty much telling your team members what to do, how to do it, and when to do it. It is a very linear and sequential software-development approach.

Hugo Sarrazin: It's not that waterfall is bad. Actually, waterfall is a perfectly fine way of doing things. There were good reasons why it emerged when the computer and mainframes and the cost of doing things were such that it required a lot of advanced planning, a lot of coordination, and a lot of rigidity to make sure that things were done in a way that dependencies were addressed properly up front.

The reality is, in today's context, with technology being more flexible and cheaper, you have the ability to think very differently about how you bring technology to market. In the old paradigm, it was not uncommon [for projects to be lengthy]—we all have clients and colleagues who were part of large projects that took many, many, many months.

We've done lots of benchmarking over the years. The most recent one, I think we reviewed almost 6,000 IT projects. Thirty-three percent of them were not on time. Forty-six percent of them were over budget. Very few of them delivered the original benefit they were expecting to deliver. That's not an indictment of the IT organization. It's not an indictment of the people who deliver the software. It's not an indictment of waterfall. It's just that it takes a long time. To be rigid, you're introducing risk by default.

Agile is the opposite. Agile says, let's break things into small little increments; let's make sure we deliver value each time, so that we don't wait until the very end to have the big hoopla and the great unveiling, because at every step of the way, you've delivered value.

Simon London: So that's a great segue into the principles of agile. Maybe just step back: if I see an organization team on the ground operating in an agile way, what do I see happening in practice?

Hugo Sarrazin: I'll highlight a few. I do think one of the beautiful things about these principles is you need to think of them in a holistic way. You can't just cherry pick a few of them. We can get into why that can lead to bad outcomes—and some companies are doing that today, and they think they're doing agile, but they get in trouble.

“It's about making work fun again. Imagine that. Imagine getting your team members to enjoy what they're doing and feel like they're accomplishing their mission.”

At the core, you need to be putting the customer first. You need to be clear on who the customer is, what problem you're trying to solve, what matters to the customer, and prioritize. Always come back to who the customer is. In some cases, the customer can be the internal customer. But often, you need to make sure that it's the external customer.

In typical organizations, the distance between the customer and the people doing the coding is eight layers of translation. That can only lead to wrong prioritization, compromise, and, in the end, your likelihood of delighting the customer and doing something that's "aha" is reduced. That's principle number one and incredibly important.

Belkis Vasquez-McCall: The second principle that I would add is around how to focus on people interactions versus process. So, how do you make sure that your team members don't just take a project plan on what we need to do and toss it over the wall to [another] team, but actually collaborate?

How do you come together to align on the objectives and mission that you have for your customers, and work to figure out the best solution to bring that to life? It's about making work fun again. Imagine that. Imagine getting your team members to enjoy what they're doing and feel like they're accomplishing their mission.

As Hugo mentioned in the first principle, it's about bringing the customer to the table. It's part of the interaction of processes that takes away so much of the focus on just checking a box—to more of a focus on how to serve our customers and get to the right solutions for them.

Hugo Sarrazin: I think a third principle that is very important is welcoming change—so removing the barriers that if you change, [the idea that] if there's failure, that something was wrong. Rather to turn it around and say, we've learned something. We're going to integrate that learning into the next iteration. Because at the heart, and the way most organizations will implement these principles, is quick cycle times: two-, four-, six-week sprints, which is often the name that is used; and to deliver something at the end of the sprint, you have the ability to learn. If you do a quick sprint, you focus on delivering something. You bring the customer to give you feedback on what you've delivered. You have an inspired team. Now we've taken the three principles and we've gotten them to work together. That's one way to implement the principles, although it's not the only way.

“It's scary for most organizations to let go. We have built organizations that are hierarchical, inspired from the military.... Now we're saying, no, let's flip it around.”

But you can begin to see how it's self-reinforcing. The fourth principle, I'd say, is to empower the team. The team knows more about the customers, it knows more about what it can do. If you make it autonomous, within some boundaries, you can have something special. The team performs at a new level. The quality of the product goes up.

It's scary for most organizations to let go. We have built organizations that are hierarchical, inspired from the military. Everything needs to flow up, all the way to the top, to people that have been promoted—based on past behavior and successes—to people that supposedly know more.

Every time you go up and down this chain, you have translation layers, and you lose some of the nuances. Now we're saying, no, let's flip it around. We're going to let the people who are closest to the problem, closest to the customer, make the trade-off within the scope that we've agreed is the scope that they can operate in. That's what makes it agile. That's what makes it speedy. That's what makes it flexible.

Simon London: What you just mentioned, Hugo, around the interaction between agile teams and the rest of the business strikes me as being fundamentally very important. Unless the whole business is operating in an agile manner, you've always got this layer of interaction between agile teams and, let's say—I don't want to pick on finance but—the finance function or control functions that may not be used to this way of operating. Is that something that organizations need to look out for?

Hugo Sarrazin: It is a constant struggle. It's something we see again and again, and it's one of the limiting factors to scale agile in organizations. Frankly, doing a pilot and having a team somewhere in a corner operating in agile is interesting and helpful. But scaling that is hard. Once you try to scale, you bump into the parts of the organization that are not organized in an agile way.

I don't think every part of the organization should be agile, by the way. Some can—and there are some criteria that can be used to identify where agile makes the most sense—and there are parts where it's maybe less appropriate. But when you have a part of the organization that is operating in an agile way, it still has dependencies on the rest of your organization. You highlighted finance, that's a common one. We can dive into that. HR is another one, procurement is another one, marketing is another one. So there all these other groups, and if they're working on a different cadence, a different quarterly cycle, they're optimizing against different objectives, and it is a problem. It's "mismatch impedance," to use a technical term, or, the gear trains are different. One gear is spinning really fast, and another one is spinning at a different rhythm.

Simon London: I would say that it sounds like, if you step back, it's about bringing the rest of the organization along with you on the journey. Because I do think, having been on the other side of the table from agile teams, immediately they start talking about their epics and their scrums and their burn rate. To be honest, unless they've educated you about how this works, you have no idea what they're talking about.

Belkis Vasquez-McCall: Absolutely. I love what you said about bringing the rest of the organization along, because it's not a one-time effort, it's not a one-time transformation, it's a journey. You have to bring every part of the organization along so that you're speaking a common language and so that you're shifting the way that you operate as a whole.

“If you only focus on increasing the speed of your agile teams and ignore the rest of the organization, you're going to create friction and prevent you from going as fast as you can.”

For example, one of the organizations that I work for, it needed to influence its project-management office in terms of how to leverage the new artifacts that it was creating in an agile team. One of those artifacts is called a user story.

A user story, in the simplest form, is how to take your requirements, your user experiences, your functionality that you're building for your customer and that you want to deliver to your customer, and write them in a form that is based on how the user is going to leverage them.

It needed to understand how these user stories align with what we typically call requirements and where there were detailed requirements—and if we were going to get audited, we knew where to point our finger. Just getting them to understand the similarities of the new world of the user stories and the requirements took time.

Taking the time to align the organization and bring it along was effective, because then it became a champion and change agent of the new way of working that we were trying to replicate across the organization.

One of the other things that I say is that you're going to be as fast as your slowest process. If you only focus on increasing the speed of your agile teams and ignore the rest of the organization, you're going to create friction and prevent you from going as fast as you can.

Simon London: It sounds like from what you're saying that the whole objective of agile, in some ways, is to reduce risk. The big risk, that the business as a whole—and particularly the finance function—doesn't want, is the delivery of some multimillion-dollar thing that's been produced by a traditional waterfall method that not only is over budget, but it actually doesn't meet requirements that have changed in the two years since they were collected. So, yes, I can see if you reframe it in a way that makes sense to the rest of the business, you see this is totally coherent and totally aligned with what they're trying to do. But you need to explain it.

“I see many organizations, where they'll get the input that what they're building is not right, and they continue to invest in it just because they've already invested x amount of dollars.”

Hugo Sarrazin: It is a wonderful derisker. Imagine I'm gathering requirements. I'm interviewing you, the user. And you tell me, I like these things to be this big, this size, this color, whatever the requirements you decide to use. It needs to go in x milliseconds. It needs to talk to the following set of other things. And you're at the point as a user providing that feedback—you have no understanding of the trade-offs you're asking. Zero. You're just asking your wish list; you're making your Christmas list. Then, if your dialogue is different, you say, OK, that's great, now let me convert that into user stories, or epic, or things that you would differently, Mr. User.

Then I ask you, which one do you care the most about? Which one do you want now? Then I'm going to go build that. And I'm going to get you part of the way. And then I'm going to show it to you and [say], you tell me, is that what you had in mind? And then I'm going to listen to your feedback. Not at the end, not in user-acceptance testing 18 months down the road, when I'm doing the great unveiling, and you go, holy crap, I had no idea that's what you were building.

When you do it on the first and most important thing that you said you wanted, I'm going to maybe fake it. Maybe get you 80 percent of the way or 60 percent of the way. But you're going to tell me right away if I'm on the right track—I have massively derisked the project, and I'm making sure you're going to be delighted.

Belkis Vasquez-McCall: I also see many organizations, where they'll get the input that what they're building is not right, and they continue to invest in it just because they've already invested x amount of dollars. They just feel like, "I need to bring it to the end. I need to bring it to the finish line even if it's not valuable." If it's not working, kill it, sunset it. Focus on the right things that are relevant for your business.

Simon London: I wonder whether we can say a little bit more about this idea that agile is not a menu of things from which you can cherry pick. That it's a system. Are there particular things that we see working with clients? Are there parts that they are tempted to leave out, maybe because it's hard, but the omission ends up damaging the adoption of agile?

Hugo Sarrazin: Every time I see this—and it's not uncommon, because there are a lot of companies that have tried agile in IT, in engineering, and in other functions, and they're scratching their head—the things I typically see that they cherry pick [are that] they do the easy stuff, they get a few people and anoint them scrum masters. But what they don't want to do, is they don't want to have autonomous teams. They don't want to let go. This is hard.

There's a manager, who's been promoted to be the manager, who likes to come in and make big calls, because he or she knows what's right. He forgets to appoint the product owner or be clear on who the customer is that's making the important trade-off. She doesn't want to move everybody together, because we have dispersed teams. The expert is busy doing other things, so he or she can't be full time on the team.

The person makes a bunch of these little compromises, which seem like they are not a big deal along the way. But in the end, what you have is a nonempowered team, with no clear customer who can represent the customer. People are unable to complete the sprint, because the subject-matter expert is doing six projects and is not fully dedicated.

At the end of the sprint, it's kind of a belly flop. It's nothing special. That is a big problem that we're seeing more and more in companies. If you don't take a system view, and you don't think about all these components together, you're not going to get the expected outcome.

Belkis Vasquez-McCall: And it's malpractice, because what's going to happen is that agile is going to surface the inefficiencies that you have within your organization. As you start to

cherry pick and say, “I’m going to have a team, but I’m not going to dedicate the role,” or, “I’m going to try to do more frequent releases, but only after I complete all of my requirements for six months,” the system is going to break down in terms of these key objectives that you’re trying to accomplish—and you’re not going to get the benefits that you could get from thinking about the whole system. The value that you expect from agile, you’re not going to achieve.

Or, if you assign a product owner that’s not an empowered product owner—he or she still has to go to 50 different people to be able to make a decision on what experience to deliver to your customer. These are the things that, if you experiment with agile and start to cherry pick—and then on top of that you try to scale that approach, which is not truly agile—it’s worse than not doing agile at all, because you’re confusing the organization with “I think I’m agile,” but I’m still following the traditional way of working, and now we’re scaling this.

Product owners represent the customers. They understand their customers. They set the vision for their customers. They dream about their customers’ experience and the functionality that they want to deliver to them.

Simon London: I just want to pick up on a piece of terminology used there, which I think you guys will take for granted but not everybody in the audience will know. Who is the product owner in all of this? And what is the role of the product owner? Belkis, do you want to take that?

Belkis Vasquez-McCall: Yes. A product owner is a critical role, and there is a debate in the industry around who plays the role, what is it. For me, I see it as a linchpin role, because it’s the core for these agile teams. There’s a couple of things to highlight.

One, product owners represent the customers. They understand their customers. They set the vision for their customers. They dream about their customers’ experience and the functionality that they want to deliver to them. They help to drive the team toward the vision that it has and encourage it as it goes on.

Second, product owners have what I call organizational capital. They’re able to influence the organization and bring it along on their vision of where they want to go. As they’re thinking about what they need to deliver in terms of functionality, they start to pull in the marketing team: How do you start to share the vision? The compliance team: How do we start to build the vision together? So, they start to rally the troops on the vision that they have for their customers.

The third one is, it is a leadership role. They help to guide the team and they are the leaders for how to make sure that we’re exciting our team members and that they’re rallying around the vision that they have. Because if you don’t put in the efforts that you need in terms of getting

the right person for the role, it ends up being a waterfall team, because the person that you assign will just continue in the traditional way of thinking and guide the team in that direction.

Many times, what I tell product owners is that they need to work in three different worlds. I know, because I've lived it. I've been a product owner before. They have to live in the present, because they need to make sure they work with their teams and they're focusing on the user stories; they need to work in the past, so any functionality that was already built, they can start to validate that with either the rest of their stakeholders or the customers themselves; and they need to live in the future, because they need to be thinking about the next set of capabilities to include in the product. What are some of the user patterns that I'm seeing that should influence my future road map?

Simon London: In a very practical sense, should the product owner come from the technology organization? Should the product owner come from the marketing organization—who, I think, in a lot of companies, would assert that they are closest to the customer and have the most insight? Or frankly, does it not matter? Is it more about the individual and having the right mind-set? And, as you say, the organizational capital to get the right to have that autonomy?

Hugo Sarrazin: The answer depends. There are individuals who are fantastic at bridging and playing these three worlds in any organization. On balance, most of these typically come from the business, because they have that organizational credibility. They have that understanding of the customer.

But if they are not able to have the credibility with the engineering team and the technical team, and if they are not curious to want to go into the details of understanding that, they're not going to be successful. You do need that special blend of skill. What you also need is good leadership. At the end of the day, the PO, the product owner, one of the main roles is she needs to protect the team. The reason the team is agile—there are all sorts of reasons—but one of the main reasons is you have dedicated people that are not being interrupted by others. What you need is a PO that can come in and shield the team from all of this wonderful attention, well-intended, from different stakeholders around the firm. Managers, cross-functional teams, et cetera. The PO needs to run interference, because if the team can run uninterrupted in a focused way—and to deliver what has been agreed at the beginning is the backlog of stuff for that sprint, and doesn't allow anybody to change that backlog during that sprint—shockingly, the team will deliver.

Agile teams are dedicated teams. They have a single purpose, a clear objective, a protector, a product owner who guides and shepherds the team along. These guys and gals work together on the single objective.

Simon London: One of the interesting implications of that is that there needs to be somebody within the business who can dedicate herself or himself full time to this and has the right skills.

I think the interesting question is, most business teams are running lean themselves. Do they need to create a role that is dedicated to this and bring somebody in with the right skill sets? Because it's hard for people to just drop everything and focus on this one product.

Hugo Sarrazin: So this is where it gets really interesting. This is why it ends up, in the end, being transformative. That's why you can't compromise on some criteria. You need to be willing to make a choice that this is important. Therefore, this team member, he or she, has to dedicate her time to this. That is counter to everything we do in a large organization.

In large organizations, the power structure gets defined in how many fingers you can put in everybody else's project. How many emails are you copied on? There's enough research that demonstrates that [with] fragmentation—and with technology we have more fragmentation now than ever—that we are reducing everybody's productivity.

Agile teams are dedicated teams. They have a single purpose, a clear objective, a protector, a PO who guides and shepherds the team along. These guys and gals work together on the single objective. If you're not willing to go there, and you're not willing to make the trade-offs and the choices and say, "This is a project that is so important that I'm going to free up everything else"—don't do it. It's not going to work.

Simon London: Sadly, that's all we have time for today. But Hugo, Belkis—thanks so much for making the time to talk to us today.

Hugo Sarrazin: My pleasure.

Belkis Vasquez-McCall: My pleasure. Agile is not going away. It's going to be front and center for many years, so I'm excited to be talking about it.

Simon London: Excellent. And thanks everybody for listening. If you want to learn more about McKinsey's work on agile and related topics, please visit [McKinsey.com](https://www.mckinsey.com). □

Hugo Sarrazin is a senior partner in McKinsey's Silicon Valley office, and **Belkis Vasquez-McCall** is a partner in the New Jersey office. **Simon London** is a member of McKinsey Publishing and is based in the Silicon Valley office.