



# The challenges with agile

## About the author



Simon Buehring is Knowledge Train's owner and founder. Simon has worked on projects since 1989 in the UK, Southeast Asia, New Zealand and Japan.

Since graduating in 1988 with a Masters' degree in Information Technology from Kingston Polytechnic, Simon has gained a wealth of real-world experience. His work on projects has spanned over 30 years and includes working as a software engineer, team leader, consultant, technical director, and project manager including time working for the BBC, HSBC and IBM.

In 2005, Simon set up Knowledge Train with a passion to help others learn the best ways of managing projects. Since then, he has established Knowledge Train as a highly reputable PRINCE2 project management training company with a focus on delivering the very best in training and customer service.



<https://www.knowledgetrain.co.uk>



20 Old Bailey, London, EC4M 7AN, England, United Kingdom.



+44 (0)207 148 5985



[info@knowledgetrain.co.uk](mailto:info@knowledgetrain.co.uk)

Copyright © 2005-2023 Knowledge Train Limited.  
Registered in UK: 5566983.  
Registered VAT: GB872413526.

Knowledge Train® is a registered trademark of Knowledge Train Limited.

# Contents



Introduction	01
Emanuele Passera	02
Andrea Fryrear	04
Martin Wickman	07
Mario Lucero	10
Tanner Wortham	13
Barry Overeem	16

# Introduction



Organisational change can be full of surprises, especially when new methods such as agile are implemented. As an Agile Project Management training provider (and firm believers in the agile mindset) we were interested to learn the common challenges faced by organisations when they transition to agile.

We consulted a handful of industry experts who were happy to share their most common struggles when using agile. If your organisation is just beginning to use agile, their stories might resonate with you and you'll no doubt find their suggestions helpful.

This ebook will help you understand the most common agile challenges, and the solutions you can use to overcome them.

## Project Manager and Software Engineer

*"Sometimes the organisation isn't ready for agile"*

The most common challenge I usually face during a transition to agile is when the organisation isn't ready for the application of agile.

The second challenge is that it is not always easy for the business side of a company to engage. [Agile project management](#) simply cannot fit every organisation. They already need to have the correct enterprise environmental factors in place to apply an agile approach successfully. Alternatively, you must receive strong commitment from senior management.

*"Working in different time zones can be difficult"*

Distributed teams are becoming the norm within [agile projects](#). However, I sometimes find working with a distributed team very challenging. Coordinate meetings or concurrent work in different time zones can be difficult and not always feasible.

The time zones problem can be eased with tele/videoconference tools and distributing necessary common work in a geographically sustainable way (contiguous time zones). In my experience, it is impossible to avoid some out of work call. A little sacrifice and good will is required.

Moreover, distributed teams tend to have less sense of belonging. The sense of belonging is hard to create, but I have seen that even a couple of face-to-face meetings can do the trick. Distributed teams tend to work together in a better way if the members know each other personally.

There is no "one size fits all" approach.

Still, I believe distributed teams are a great opportunity. We are able to reach, engage and work side by side with the best talents from all around the world.

*"Team members can be scared of failure"*



In my experience, estimations are always tricky. People tend to fear being tied to something that could easily be wrong.

To make this easier I believe that you should create an environment based on trust and respect, in which people are allowed to make mistakes. In a trusted environment, everybody is eager to commit themselves to an estimation. This is also true for managing issues with honesty and accountability. In an environment based on trust and respect, everybody is eager to assume responsibilities.

Sprints need to be kept short enough so that they won't have a big impact on the project if they fail. In my opinion, it's better to abort a sprint than to waste money in producing something wrong.

## Speaker, Author, Agile Marketing Consultant

*"We can't mess up or the Scrum Police will come and take us away"*

I think with [Scrum](#) in particular there's a fear of doing it "wrong," especially among marketers. We have a sense that it's a sacred way of working that we can't mess up or the Scrum Police will come and take us away. But really every team — dev, IT, or marketing — makes adjustments as they go along. That's the whole point of continuous improvement.

*"Marketers sometimes don't play well with others"*

On marketing teams you nearly always get one hold out, someone who just doesn't truck with this new-fangled way of doing things. They're actually the most dangerous, because they aren't so opposed that they speak up, they just dig in their heels and quietly hold up the whole team.

Marketers sometimes don't play well with others. We like to get credit for our work, and with [Scrum](#) it's all about the team. If there are team members like this, you've got to work extra hard to praise their individual impact in a one-on-one setting so they feel appreciated and validated. However, the primary emphasis outside those cheerleading sessions remains on the team's achievements and problems.

I don't know if conflict is more common on marketing teams than on dev teams, but the Superman complex that many marketers harbour can definitely get in the way of a team-centric approach and cause some friction.

*"Lack of trust in the agile process"*

I find that marketers can often lack trust in the agile process. They sometimes have a hard time believing it's really going to work, or that it's really going to do all the great things they've heard about. A couple of iterations in, and that usually goes away.

*"Managing relationships with freelancers and agencies"*

There are skill gaps that marketers are accustomed to filling with freelancers and/or agencies, so if those relationships start to interfere with the team's ability to successfully complete sprints, they will become a challenge that needs to be addressed.

In my case, outside interference is far and away the most common cause of sprint failure. Executive/stakeholder "emergencies" combined with demands from other departments are the bane of just about every agile marketing team I know.

### *"Creatives tend to struggle with estimations"*

Creatives definitely have more trouble estimating. Sometimes it takes them half a day to complete an article, and other times they struggle with it for three or four days. It's a legitimate problem, and one that we typically solve by relying on data — how long does it usually take them — and communicating heavily if it looks like they're going to miss that mark.

Choosing the best estimation method has also been particularly difficult in my experience — we've tried t-shirt sizes, Fibonacci, and even hours, and there isn't one that seems to work for everybody. It's just a matter of trial and error, finding the method that reflects how a team thinks. Getting tasks and stories down to standard sizes really helps too.

### *"Snacks, chocolate, or beer help to keep meetings varied"*

I often experience problems with employee engagement during daily Scrum meetings. This can happen particularly if part of the team (the content marketers, for example) are involved with projects that don't touch the other team members. It's hard to stay focused when you believe that someone's updates aren't going to affect you. We've tried using a more Kanban-style stand up where the discussion centres on planning and strategising for the day, rather than the traditional 3-part yesterday/today/block structure, which seems to help.

As a [Scrum Master](#) I take great pains to vary the format of meetings without changing their objectives, particularly with retrospectives. Asking different questions, using different conversation starters, and just basically being creative helps people look forward to them. I also recommend bribery via snacks, chocolate, or beer.



If you suspect someone of being less than truthful, the meeting is not the appropriate context for discussing that. The Scrum Master and/or marketing manager would need to address the situation with the offending team member separately, with a focus on getting to the root of why they felt they couldn't be honest with the team.

## Senior IT Consultant Developer and Agile Coach

### *"Too trusting of early estimates"*

It's hard to accurately estimate software tasks. Really hard. Organisations often request estimates that span months or even longer. In my experience, new development teams can estimate a couple of days' work with precision, a week with some luck and maybe two weeks with some dignity still intact. More than that, and we're entering unknown territory and are pretty much guaranteed to be wrong. That's just how it is and we need to accept that. Unfortunately, estimates are often given as deadlines to the customer which then expects the team to deliver on the mark.

But the nice thing here is that self-organising teams do become better with time and practice. Give them two-three sprints, allow them to fail, allow them to grow and allow them to learn. After a while, estimates will get better and the team's velocity becomes stable enough to make more educated projections.

Note the word projection here. The idea is to be able to say something like: "We're confident that we can finish these 20 stories, and we're 60% certain that we can complete 25 stories, but it's near impossible to finish all 40 stories". Now the organisation can make real business decisions based on reasonable projections.

### *"Skimping on retrospectives"*

Scrum teams run retrospective meetings. It goes like this: Sit around for a while. Talk some. Vote on some items. Get back to work. Done.

Sorry, but that won't cut it.

You see, the only way to improve is to get honest feedback and then act on it. That's how we can figure out better ways to work and avoid making the same mistakes again. Retrospective meetings are created for that reason – to improve how your team works by looking back at what happened and then decide how to change based on that.

Skipping this part is one of the main reasons team fail. It's worth spending some time learning about this. Your team will thank you for it!

**"Not understanding the process"**

Lots of project managers fresh out of school (or just, well, old-school) don't understand what agile is about. What I see is the tendency to simply map known concepts with this weird Scrum thing and then carry on as if it's business as usual.

So daily Scrum meeting becomes daily status meeting, product owner becomes project manager; stories equals tasks; team members becomes interchangeable resources; sprints becomes schedules and estimates turn into hours and deadlines on a spread sheet, and so on. The result is disaster. It's easy to fix though. All it takes is just one hour.

Print out a copy of the Scrum guide and read it. It's like ten pages of text. Ten measly pages. It's not a hard read. It can be a bit scary at first, but not hard. Just do it. Please.

*"Teams need guidance to reach awesomeness"* All teams have to pass through a bunch of phases on their journey to greatness. These phases are called forming, storming, norming and performing. Long story short: forming is when they meet and everything is new and exciting. After some time, the team enters the dreaded storming phase. This is where the struggle for power and control happens. People show distrust, disagreement and form alliances. It can be nasty.

What happens now is they start bouncing back and forth between the forming and storming phase. You see, as soon as they get to the storming phase, people tend to back off and move back into the forming phase again – the safety zone. And the cycle repeats itself...

In my experience, teams left to fend for themselves almost never get past the storming phase. More time will not help here. I've seen lots of teams that have been working together for years and still don't manage to progress any further.

The way to navigate this storm is to recognise it as something perfectly natural, but also realise it won't fix itself. The team needs guidance to reach awesomeness. They need someone with experience and coaching skills to navigate this part. Maybe the Scrum master can help or maybe you need to bring in a dedicated agile coach. Whatever you do, don't let them handle it on their own.

## Freelance Enterprise Agile Coach & Change Agent

For the majority of companies I have consulted, these are the most common challenges: silos of knowledge, managers with the famous 'command and control' attitude and a lack of agile culture.

### *"Silos of knowledge"*

One of the biggest challenges agile projects face is related to silos of knowledge. The term Subject Matter Expert (SME) is an overvalued concept if you have only one SME for each product or business domain in your company. This is because the SME will be a bottleneck in the short term. In Scrum projects, we work together as a team and everybody must know about all products to avoid dependencies.

What can you do to eliminate silos? Well, as an example, let's say you hire some new developers who don't know how to deal with legacy code (most companies have one or more). Those developers will have to request the help of the SME. This situation is quite normal if you have two or more SMEs, but if you have only one SME to assist the new guys, then that SME is a great impediment.

The best approach to eliminate this kind of silo is to use a controversial technique called 'Pair Programming'. With this technique, two developers share the keyboard when they are developing code. This means the SME can fix a bug from the legacy code and the new guy can start learning the code.

Menlo Innovations, a small software company in Ann Arbor, Michigan uses the 'work in pairs' approach for everything, in order to avoid silos. In my opinion, this is awesome!

### *"Managers who use the famous 'command and control' approach don't trust their employees"*

Based on my experience, managers who use the famous 'command and control' approach don't trust their employees.

Imagine the following situation: your company starts to use Scrum when managing IT projects. During the sprint planning, Scrum team members decide to work in five user stories for the next two weeks (assuming that period of time for the length of the sprint). However, after a couple of days, the manager decides to add a couple of bugs in production and other user stories he considers necessary. The message to team members in this situation is: 'our manager doesn't trust our capacity to plan. He thinks that we are lazy and added more duties for us'.

My recommendation for this issue is that managers have to become leaders. It means that they have to build a safety ecosystem where team members can excel. Furthermore, as a leader, you should push your employees to read, participate in workshops and create a community of practices.

*To avoid each employee being concerned only about business unit metrics, you have to adopt the mantra: collaborate, collaborate and collaborate*



If a company does want to adopt agile it must build a culture to support the agile principles. Everybody has to pursue one common and big goal – the company goal.

I have come to realise that, in many companies, each business unit is concerned only in achieving his targets. Most of the time, the employees achieved their goals, but as they did not work as a team, the goal of the business wasn't reached. To avoid each employee being concerned only about business unit metrics, you have to adopt the mantra: Collaborate, Collaborate and Collaborate.

To adopt agile successfully, it is also necessary to have a culture of transparency that improves the morale of employees. For instance, there are companies (unfortunately not many) in which employees know how much their co-workers earn or salaries are listed publicly.

*"Agile is about mindset"*



Agile is about mindset, but it is being corrupted by approaches which put more emphasis on tools and processes. The key to the mindset is the ability and willingness to learn. In a complex world, your only chance for survival is learning.

## Silicon Valley Agile Coach and Author

*"Scrum is easy. People are hard"*

What's the biggest challenge when adopting Scrum? That's easy. The start. Here's why: At the start, we ask this newly christened **product owner** to write requirements from a customer's perspective. That's really, really hard, and that's often their first exposure to Scrum.

At the start, we stick a group of specialised team members in a room and ask them to understand each other's languages and focus on the team over their specialty. It hurts.

At the start, we ask a scrum master with just a two-day course under their belt and limited Scrum experience to shepherd this team to success. It's scary, and s/he is often unprepared for what's to come.

There's more than just these three points, but I think you see the dilemma. The root of the problem is simple really: Scrum is easy. People are hard. The solution? Hire someone with a wealth of agile coaching experience to model agile behaviours and educate team members, product owners, Scrum Masters, and leadership. But if you know little about Scrum, how do you know the person you're hiring is worth his/her weight? That can be deceptively difficult.

*"Myths get started about how agile teams don't plan. It's simply not true"*

That's a tough question. Narrowing a list to down just a few is a challenge. Why? Scrum often runs counter to an organisation's culture, so every adoption is different. Imagine if I told a project manager to toss out his gantt chart and risk register. We don't need it; we'll just figure these things out as we go. This PM would likely pass out! And that's also how myths get started about how agile teams don't plan. It's simply not true. We often plan more than a traditional team. But I digress. Let's talk about some of those common challenges:

*"Scrum is a problem-finding, not a problem-solving framework"*

When Scrum is first introduced to an organisation, it exposes a great deal of dysfunction. This is intentional. People solve problems, new processes and new tools don't. When's the last time you've seen a hammer swing itself? I think you see my point.

Often, this dysfunction is attributed to the organisation's Scrum adoption. After all, we didn't see all these problems until we adopted Scrum so Scrum is obviously the problem. If we do away with it, these problems will go away. Nothing could be further from the truth.

### *"Cargo cults kill Scrum"*

In World War II, Japanese and later Allied forces placed an air base on a remote island. On this island was an indigenous tribe never before exposed to modern technology and comforts. After the war, Allied forces abandoned the air base, and the tribe no longer had access to these comforts.

The tribe assumed that if they build planes themselves, the Allies would return so the tribe began building planes out of whatever they had available, usually sticks and branches. Organisations often form cargo cults. They adopt Scrum, but instead of doing anything differently, they simply relabel existing meetings and processes with Scrum nomenclature.

Here's a common example. Instead of calling it a status meeting, we'll call it a stand up. It'll continue to be 30 minutes, but we'll have more of them, and we'll stand up when we do.

### *"Many shops who claim to be agile but release only two or three times a year"*

I met Jeff Sutherland, the co-creator of Scrum, several months ago while he was in Silicon Valley. I could sense his frustration when he talked about many shops who claim to be agile but release only two or three times a year.

Part of the beauty of Scrum is that we deliver working software at the end of every sprint, which can be anywhere from a week to four weeks long. We do so because we inspire great conversations with our customers when we give them something they can touch and interact.

Additionally, it allows us a great deal of freedom when it comes time to release our product. After all, we don't need to integrate or harden our product; we've been doing so every sprint. It's incredibly challenging for teams to switch from

a mindset of delivering once a year to delivering every few weeks, but it's possible.

Our teams at Study.com release often. Not once a sprint. Not once a day. But anywhere from 2 to 5 times a day. It took time, blood, sweat, and tears to get there, but it's been worth it.

## Learning Facilitator for Agile Teams

Recently I got asked what I consider the most common challenges with agile projects. These are projects that have such a high rate of uncertainty and complexity on how and what to build, an agile approach is necessary. Although my gut feeling immediately provided an answer, I gave myself some more time to think about the question. However, this week my initial thoughts proved to be correct. Without any doubt! The #1 challenge with agile projects is ensuring the right side of the [Agile Manifesto](#) truly enables and strengthens the left side. In this article I'll explain the reasons behind this statement by exploring the Agile Manifesto and the misunderstandings about it.

*"The original manifesto remains my primary source to think about agile related questions"*

In my role as an Agile Coach, the Agile Manifesto acts as guidance. The 4 values and 12 underlying principles support me exploring options to daily issues, challenges and questions. I've read many books about agile software and product development, but the original manifesto remains my primary source to think about agile related questions.

*"Without exception, every organisation wants what the Agile Manifesto values"*

I've worked in quite a few different organisations. Without exception, every organisation wants what the Agile Manifesto values. It should be about collaboration, teamwork and open communication. Sure, working software or a working product is what it should be all about. Without doubt everyone wants a fruitful partnership with their customers. And yes, we all understand that we live in a VUCA world. A world with a high rate of volatility, uncertainty, complexity, and ambiguity. In such a world, organisations need to respond to change quickly to become or remain successful.

But we do need to respect current processes and tooling; this is simply necessary considering the legislation.

We do need to write proper documentation. The more complex the product, the more time we should spend on determining how to build it, and of course we should write down all our thoughts (guesses and assumptions).

We do need a contract that protects us when it gets tough. Even better: let's describe the desired result in the contract in as much detail as possible. At least that won't cause any misunderstandings.

We do need a plan. Where would we be without a plan? Although it's quite difficult to predict what's going to happen upcoming year, we do need a detailed plan that eases the minds of our stakeholders. Even better: let's create the plan upfront and give it to the supplier. The only thing that the supplier has to do is execute the plan. How difficult can it be?

*Ensuring the right side of the Agile Manifesto truly enables and strengthens the left side is the #1 challenge with agile projects*



There is nothing wrong with processes, tooling, documentation, contracts and plans. But they should all enable the left side of the Agile Manifesto! Not the other way around!

Processes and tools are valuable. But they should enable the collaboration, interactions and dialogues of all the individuals involved.

There's nothing wrong with documentation. But don't write everything down for the sake of documentation. In the end it's all about working software or working products. Only working software will help you to validate if you're building the right product right.

Contracts are important. They can offer everyone clarity and guidelines on what to expect from each other. But don't expect a great collaboration with your customer when the contract says "we don't trust you!"

Plans can be useful. But in the end it's all about the conversations we have with each other to create common understanding about the desired results. Remember:



“Planning is everything, plans are nothing.” – Field Marshal Helmuth Graf von Moltke.

“The #1 challenge with agile projects”

Every organisation wants to be agile. Every organisation wants the necessary agility to deal with the increasing rate of volatility, uncertainty, complexity, and ambiguity. The Agile Manifesto offers 4 values and 12 principles to support organisations build software/products in such a complex environment. Many organisations acknowledge these values and principles.

However, ensuring the right side of the Agile Manifesto truly enables and strengthens the left side is the #1 challenge with agile projects. It's a challenge all organisations must deal with, otherwise projects with a high rate of complexity and unpredictability are bound to fail!

*Although the struggle with the values of the Agile Manifesto is a common challenge, organisations do accept more often that change is necessary and inevitable*



I do want to close this blog post with a positive note. Although the struggle with the values of the Agile Manifesto is a common challenge, organisations do accept more often that change is necessary and inevitable. Doing some agile practices mechanically isn't enough. Agile projects will only become successful when the values and principles are truly embraced and put into practice.



<https://www.knowledgetrain.co.uk>



20 Old Bailey, London, EC4M 7AN, England, United Kingdom.



+44 (0)207 148 5985



[info@knowledgetrain.co.uk](mailto:info@knowledgetrain.co.uk)

Copyright © 2005-2023 Knowledge Train Limited.

Registered in UK: 5566983.

Registered VAT: GB872413526.

Knowledge Train® is a registered trademark of Knowledge Train Limited.