Free Chapter

Quick Glance At: Agile Anti-Patterns

David Tanzer

This is a free chapter of the book “Quick Glance At: Agile Anti-Patterns” by David Tanzer. Before this chapter, the book explains how companies even get into the situation of experiencing anti-patterns. And how some people start to spot that things are now working as they should be.

The company has "successfully transitioned to Scrum". You work in two-week sprints and do all the rituals. But behind the scenes, most things in your company work exactly like before.

Antipattern #1 Agile Theater

Your team is “doing Scrum” now. You do all the rituals. You have the meetings: The daily stand-up, sprint planning, review, retrospective and the backlog refinement. Your company has renamed all the roles in your development teams. Your iterations are now called “Sprints”. You write all work items for the team in a way to fit the user story template “As a ... I want … so that …”

On the surface, it looks like you are doing everything that’s in the Scrum Guide. But when you take a closer look, you can see that nothing has *really* changed.

All the Scrum rituals are just a stage play. Your company is playing “agile theater”.

Problems we See

People in some – small or large – pockets of the company have started doing Scrum. They behave according to the Scrum rules – at least some of them.

But the company has not changed yet. So, the problems we see boil down to problems of a company that is not agile.

Decisions still take a long time. They require sign-offs that are hard to get. Or you must discuss the topic in countless meetings. Maybe even with people from different departments, who have their own agenda.

For example, you cannot simply buy new laptops or better software or even just a new keyboard for your developers. The purchasing department takes care of that. But it will take some time before you get it.

Or, getting new user stories into the backlog is only possible during official project kick-offs, because only there all the stakeholders are available. This delays decisions about scope.

And there are more ways how decisions get delayed in companies that are starting to adopt agile processes…

Implementing changes still takes a long time. Even when there finally is a decision, you cannot start to implement it right away.

Maybe the right people to implement the decision are not available – You first must form a project team. Or the team that should implement the decision is currently overloaded with even-more-important work.

Long lead times for all kinds of deliverables. You should deliver something to internal users or to customers. How long does it take, from the moment when you know you must deliver it, until the actual delivery?

For example, when a user story enters the backlog, how long will it approximately take until the software is changed and the change is tested and deployed to production?

If you are still in “Agile Theater” mode, you probably also have long lead times.

Large Releases. Releases to production are painful. They take a long time to test and roll out, and you must coordinate them with different departments (operations, testing, centralized release management, …).

Because of the pain, you do them less often. And because you do them less often, they become larger: There are more features in every release.

The company cannot react quickly when its environment changes. When the environment changes, you first need a decision to react. And decisions take some time.

Then you create a project for the company re-org. With a transition plan, a core change team, meetings with managers on all levels of the hierarchy, …

And then, when the company has *finally* changed, you can only hope that the environment has not already changed again.

Why this is a Problem

If you work in an agile theater, you do not get any of the advantages and benefits of being agile, but you get all the downsides.

Yes, there are some downsides to working in an agile way.

For example, when you introduce Scrum, you suddenly have more meetings. And you have one person, the Scrum Master, who has a new role that did not exist before.

Those things reduce your capacity to deliver software: Your teams have more “unproductive time”.

When you do Scrum right, the meetings and roles give you much more benefit compared to what they cost. But in an agile theater, there is only the cost.

Root Causes

People in your company follow the rules, processes and artifacts of your new, agile way of working.

But they do not yet “think agile” – They do not yet have an agile mind-set.

Wrong organizational structure. To be good at software development, you need an organization that is optimized for software development.

For traditional companies that try to become agile, software development is often not their core business. They do banking, logistics, …, so they have an organizational structure that optimizes for their core business.

And they often have had their basic organizational structure and hierarchies for decades.

And because of this structure, IT teams cannot work closely together with other stakeholders: Users, operations, marketing, legal, …

The organizational structure impedes communication, collaboration and fast feedback – It impedes agility.

Middle management impedes agility. Sometimes, when companies try to become agile, people in middle management fear losing their responsibilities – losing their power.

So, they try to create additional checks, sign-offs and quality gates. They create additional steps in the process.

These additional steps slow down delivery. And they even make it hard to deliver more often: The people whose signature you need are not available to sign a new delivery every day – probably not even every two weeks.

Annual budgets / KPIs / bonuses. The company is planning budgets on an annual basis. People in the company get bonuses based on KPIs on an annual basis.

Your company now has one cycle that lasts an entire year, and probably another two-year cycle. One where people delay decisions during bad years, to maximize their bonus for the *next year*.

This delays decisions and deliverables. It makes it hard or impossible for the company to quickly change when the environment changes.

A desire for bureaucracy and micro management.

We all want to be treated like adults. But when it comes to treating others, we often think that we can do the job better. We want to micro-manage them or even just do it ourselves.

Good managers learn to resist that urge.

But in traditional companies, the ability to micro-manage people might be the reason why this person has become a manager in the first place.

Managers must unlearn their micro-management skills, and the company must let go of most of its processes and rules.

Too many processes and rules. Nobody starts a company with the goal to create a bureaucracy. But most companies become more and more bureaucratic the older they get – Some of them faster, some slower.

So, by the time people decide to transform the company to become more agile, there are already many rules and processes in place.

Those rules and processes limit how agile the company can become. They prevent self-organization, they impede effective communication and they make it impossible to take decisions quickly.

Yet, your company is not willing to replace those rules with something more agile, or to abandon them altogether. They were created for a reason, so you cannot abolish them without a reason.

How it starts…

In my experience, the agile theater is often a natural step on the way to becoming agile.

Some people / teams start with a new methodology – e.g. Scrum. They learn about it and start applying it mechanically, while the rest of the company works more-or-less like before.

Now you already have an agile theater: A company where some – or even most – teams work “according to Scrum”. But the company did not really change its way of working yet.

It only becomes an anti-pattern when people stay in this step.

Unfortunately, staying in this step is also a natural thing to do.

You are now doing “Scrum by the book” – You’ve made it! You do the stand-ups, the reviews, the planning, the retrospectives. And you even find action items in every retrospective.

But these actions are only local optimizations. What you are doing *looks exactly right* from the outside. And because of that, your company stays in the “Agile Theater” phase, far away from the global optimum.

Getting Better

To get better, you first must realize that you are in the “agile theater” phase. Then you can start to remove the road blocks that prevent your organization to become agile at the company level.

Beware of annual budgets / KPIs / bonuses. You might think that those techniques motivate people to plan ahead and to do a better job.

But they are really holding you back. They slow down decisions, they create competitive situations where people should work together, they reward people for *not* being team players, and they often create incentives for optimizing the wrong things.

Consider this: Your KPIs for your annual bonus were decided up to one year ago. The world probably has changed since then. So, your KPIs incentivize you to reach your company’s last year’s targets, not what you should do now.

Stop having KPIs that become targets. They hurt your ability to become agile.

Do **not** stop all planning. Instead of annual budgets, plan more often and become more flexible when allocating budgets.

Short communication paths. In larger companies, there are often unwritten (or even written) rules for who you are allowed to communicate with.

Like, if you want to contact someone from the XY department, contact their manager first, or the manager will think you cut them out.

If you want to communicate something to middle management, ask your own manager. She will do so at an appropriate time and filter your message.

Take efforts to stop this behavior – To create a culture where everyone can communicate with everyone else. And make communication rules that you folow right now visible.

De-centralize decisions. Say your team wants to take a decision. If you need a sign-off from a manager higher up the chain, this slows you down.

Centralized decisions slow you down, and things that slow you down prevent you from becoming agile.

Push more decisions down the hierarchy. Allow “line workers” and teams to take decisions and implement them right away.

And teach and coach them to do so in small, reversible, safe-to-fail steps.

Turn the organization by 90 degrees. Instead of having specialized departments (like IT, sales, marketing, development, testing, …), create teams where all the people required to solve a problem are working together as a team.

Move the people who can solve problems closer to the people who have them.

A team that can solve one of your company’s problems completely on their own needs many roles and specialists: Software developer, tester, operations people, maybe legal, marketing, …

Allow them to work together as a team, not as specialists from different departments.

Remove rules and processes. “The rules and processes were created for a reason, so you cannot abolish them without a reason.”

Well, your reason now simply is that the rule prevents your company from getting the benefits of working in an agile way.

Now, try to abolish the rule or replace it by something better *in a safe-to-fail way*.

Find out why the rule was there in the first place, and design a small experiment to find out whether you can get the same benefit with a simpler process, or no rule at all.

Did you enjoy the free chapter? Then ***get the full book here****, for only ~~€ 14.90~~* ***€ 9,90****:*

https://gum.co/qga_agile_antipatterns/wx05c3fc1

About the Author

“There is too much bad code out there. And there is too much bad management out there. And there is too much frustration out there. I try to help my clients to **develop better software** and to **get better at developing software** while **enjoying their work**.”

David Tanzer has been working as an independent software consultant since 2006.

He helps teams to get better at working together, understand customer requirements better and produce higher-quality code.

He works together with teams at his clients as a trainer, consultant, coach or player-coach.

Services

Training: I come to companies and teach their developers, testers, scrum masters and product owners new skills.

Consulting: I come to companies that realized something is not working well and help them to find solutions to their problems.

Coaching: When a company already knows how to solve their problems, I help them achieve their goals.

Contact

Find out more about me on my website:

http://www.davidtanzer.net

Contact me / Hire me / Ask me any question:

Twitter: @dtanzer
Email: business@davidtanzer.net

I speak at conferences and meetups. If you see me at one of these events, say hi. I love meeting new people!

About the Illustrator

Irina Linuza is a graphics designer / game-artist / illustrator who expresses her talent and fantasy through her artworks. She lives in Riga - the capital of Latvia, a small European country. **And Irina says: “This has never prevented me from working with clients from around the world as a freelancer.”**

My experiences range from creating fantasy illustrations, 2D game characters, backgrounds, maps, to designing game interfaces and logos.

I have professional knowledge and experience working both online and offline.

Before I start working together with my clients, I discuss the main idea and the target of the project. **What can I say, I like to get to know my clients! My workflow is based on open communication and feedback.**

Portfolio

Upwork:
https://www.upwork.com/o/profiles/users/_~01678d123d582fc42e/

Behance:
https://www.behance.net/irinalinuzc3b6

Contact

Contact me / Hire me / Ask me any question:

Upwork:
https://www.upwork.com/o/profiles/users/_~01678d123d582fc42e/

Email: irina.linuza@gmail.com