Your guide to

Agile

Service Management

Bringing back speed, flexibility and customer focus to your IT team

**TOPdesk**

Index

Introduction

Agile is increasingly popular these days. Originated from the Development world, it's quickly gaining ground in other domains. Service Management is not an exception. But what exactly is Agile Service Management?

In this e-book, I'll give you answers to questions like: what's the difference between agile and Scrum? Can I work according to ITIL and agile at the same time? How do I put the agile philosophy in practice? And where do I begin?

Using real-life examples, I'll show you what an agile mindset can do for you and your team. And how agile will help you improve your service delivery.

I would like to thank Gerard Bakker, Steven Happee and Mark van Meurs for their contributions to this e-book — thanks a lot!

Enjoy your read,
Bas Blanken, Agile Service Management expert
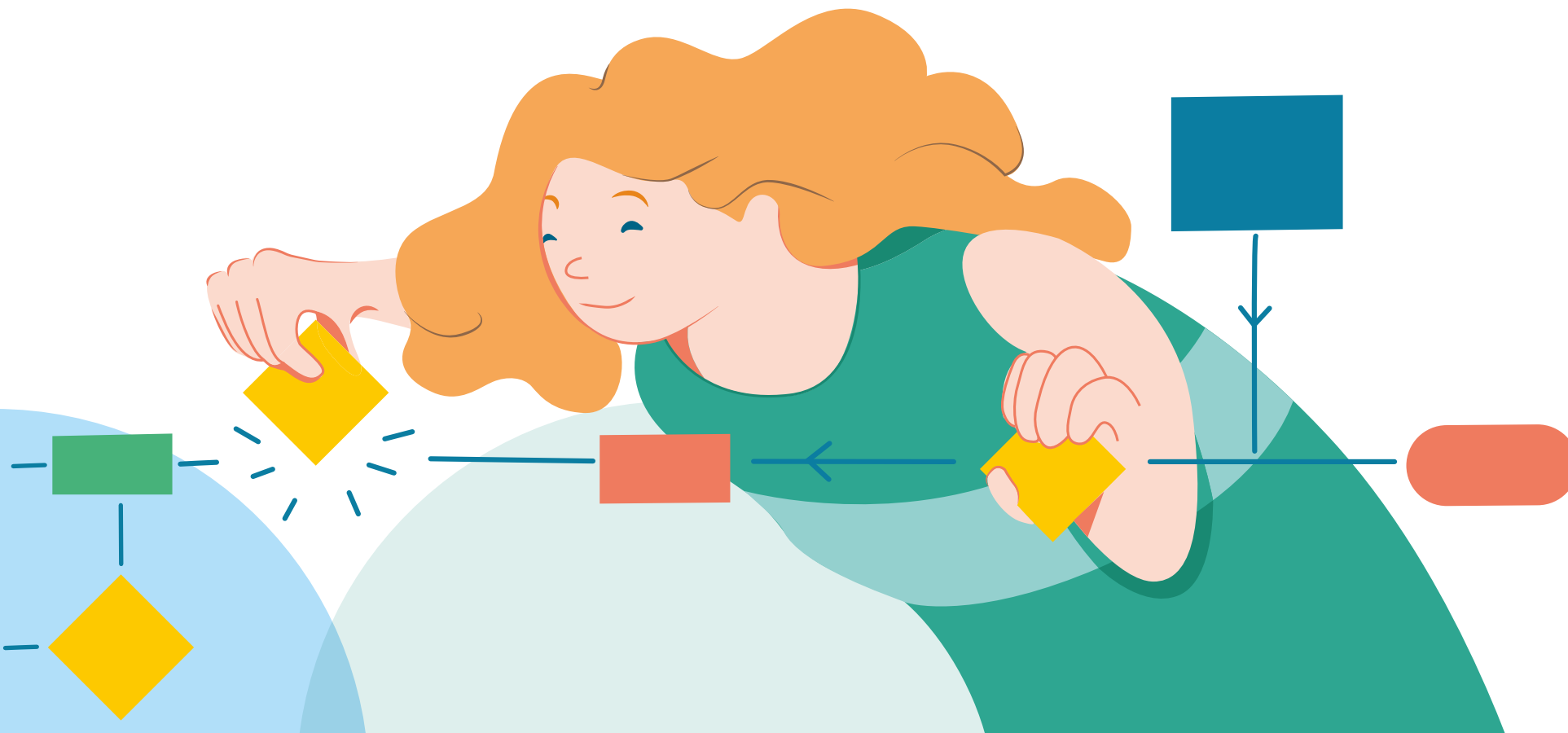
1. What is agile?

Agile is one of today's most popular buzzwords. It's also a widely misused and misunderstood term. So let's first get back to basics: what exactly is agile?

The birth of agile

In 2001, the software branch introduces the Agile Manifesto for Software Development. The manifesto comprises 4 values:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The inventors of agile have turned these 4 values into 12 principles. These principles are aimed at software development. The agile mindset is however applicable to all branches.

The most important reason to introduce agile to software development was to make large organizations more flexible. For smaller organizations, it's easier to respond quickly and meet their customers' demand.

These smaller companies don't have a set organizational structure. Large companies are a lot less flexible. They often use a waterfall structure for projects: a plan or design needs to go through different departments and management layers before it can be executed. The result? An unwieldy organization.
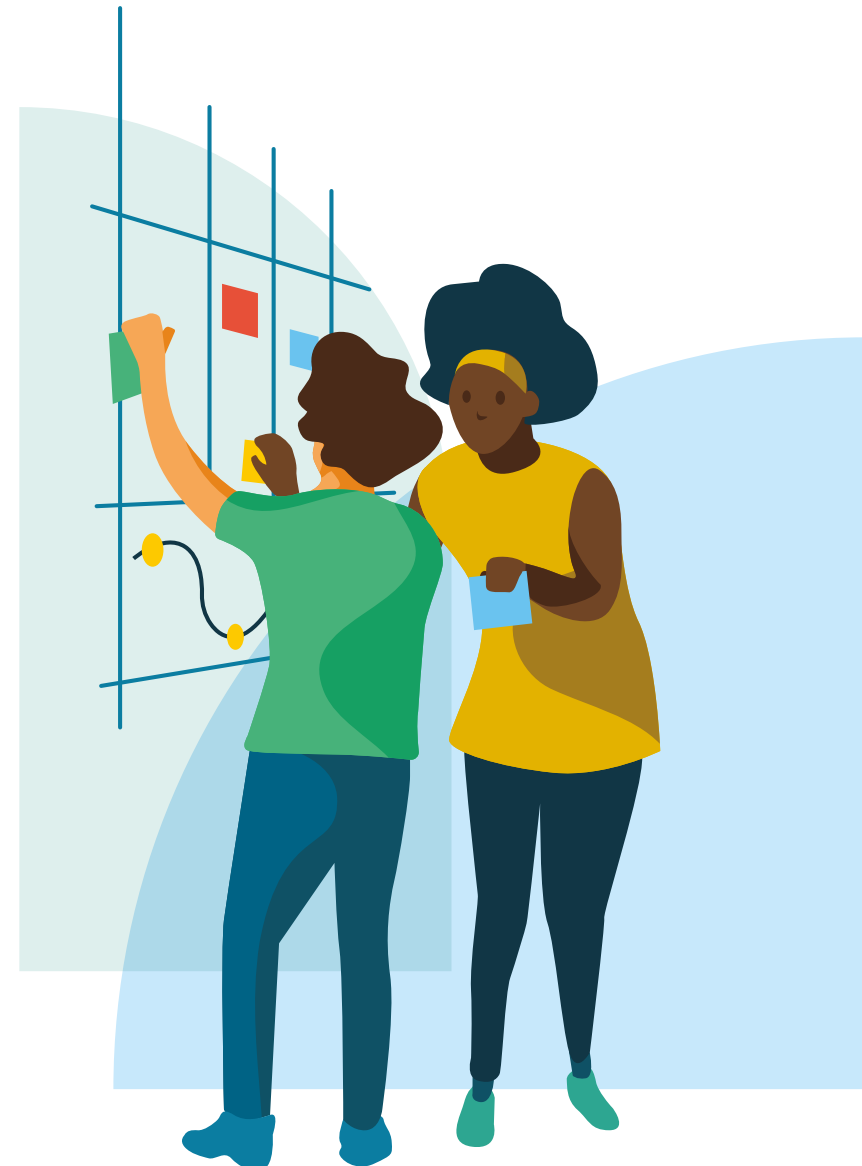
What is agile?

Agile is a mindset. The idea: if you want to survive as organization, you need to be flexible.

Let's compare this agile mindset to a jaguar.
A jaguar's instinct is to survive. And to survive he needs to be dexterous and agile enough to react quickly to the movements of his prey. For organizations, it's just as important to be dexterous, especially now technologies follow each other in rapid succession.
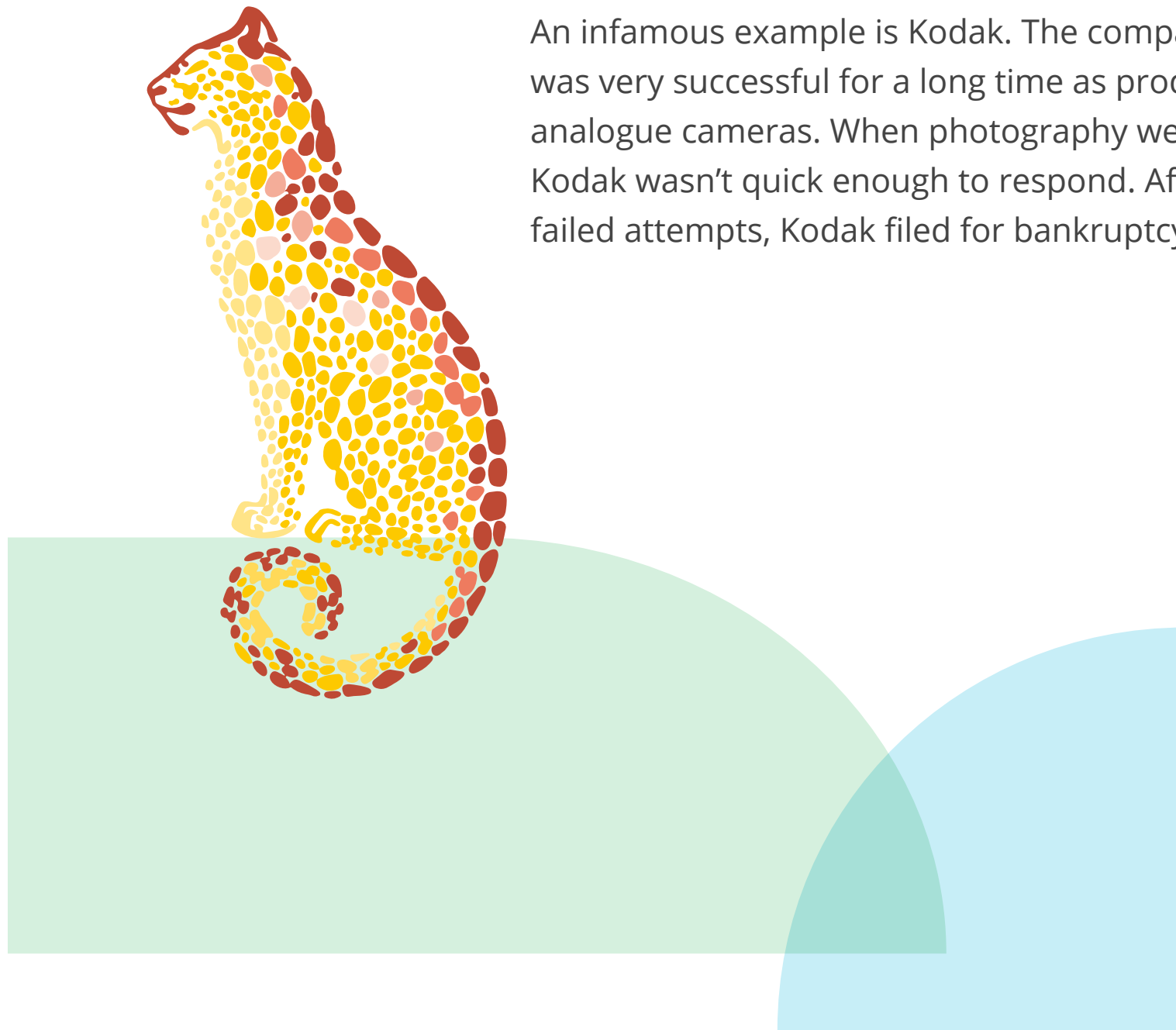
Your organization needs to be flexible enough to quickly respond to new technologies and the ever-changing demand of the customer.

An infamous example is Kodak. The company was very successful for a long time as producer of analogue cameras. When photography went digital, Kodak wasn't quick enough to respond. After some failed attempts, Kodak filed for bankruptcy in 2012.

"With an agile mindset, you assume that your plans are going to change."

The gains: bring back fast responses and flexibility

An agile mindset brings back flexibility and shorter response times to your organization. When you work agile, you strive for the least amount of bureaucracy possible. Agile requires a different type of employee as well. In an agile working environment, you want employees to share knowledge, act on creative ideas and come up with solutions. The initiative is not with the manager, but with the professionals.

How does agile work?

There is no manual you can follow to start working agile. It requires a cultural shift within the organization. The most important shift is that your organization starts embracing change. In the traditional work model, you try to keep changes to a minimum: you create a plan and try to keep to it as much as possible.

With an agile mindset, you assume your plans are going to change. You're not going to follow the same plan for two years. You have a clear goal in mind, but you can change course. Agile working means continuous improvements. Your work is never done.
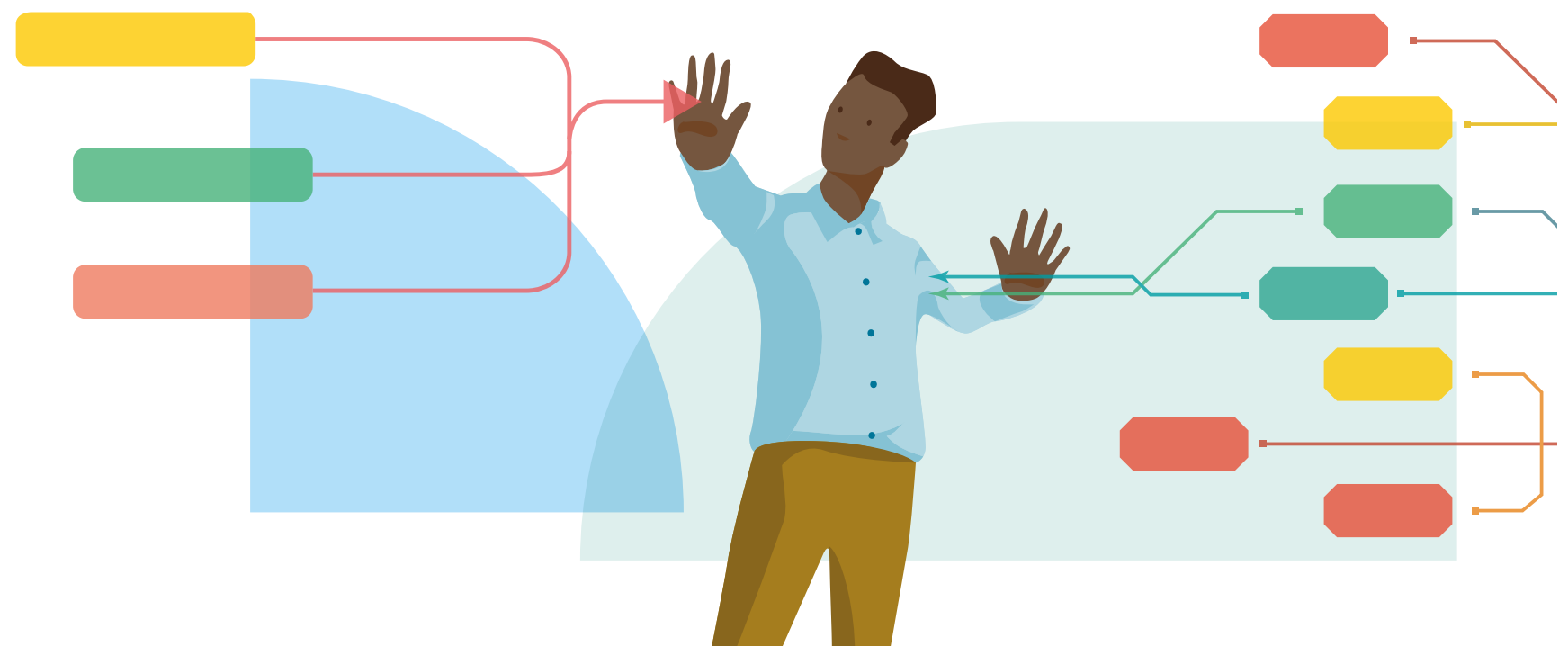
2. What is scrum?

One of the most popular agile
frameworks is Scrum. But what is it?
And how does Scrum relate to agile?

The birth of Scrum

It all starts in 1986. Ikujiro Nonaka and Hirotaka Takeuchi publish their 'The New New Product Development Game' article in the Harvard Business Review. In this article, they conclude that historically, projects in small, multidisciplinary teams book the best results.

With this conclusion in mind, Jeff Sutherland and Ken Schwaber created a development method for the software branch in 1996. Scrum was born.

What is Scrum?

Scrum is a cost-free framework for software development. The framework makes it easier for organizations to develop and maintain products in complex, dynamic environments. Scrum is an answer to the rapidly developing technology industry and the fast-changing demands of customers. The framework has an empirical starting point: you learn by doing and use your findings to determine your next step.

How does Scrum work?

Scrum works for small self-managing teams of 3 to 9 people. A Scrum team works according to a step-by-step method. The team delivers a new or improved product, or a new or improved functionality, within a set period of time (two weeks, for example). These short 'sprints' force you to constantly work with realistic deadlines.

The gains: transparency, inspection and adaptation

What's to gain from dividing your work into sprints? You plan your work more realistically. You know what you need to do and how much time you have for it. This makes for a more predictable planning of your work.

Risks are also more manageable when you use these shorter periods. You're not going to make a long-term plan with an extensive risk analysis. With each step the team takes, they show the organization what hurdles they needed to overcome, which scenarios the team can follow and what their impact is. Based on this information, the organization can adjust the course of the team when needed.

The short sprints also ensure that it's transparent what the team makes. At the end of the sprint you show your customer what you've made. The advantage of transparency? The customer gives regular feedback the team can take with them in the next sprint. This ensures that the product you make is something the customer is really happy about.

Want to get started with Scrum?

Just follow the rules in the **Scrum guide** and get started. Get some experience and discover how Scrum works in your organization.

3. What is agile Service Management?

So, now you know what agile is. But the next question arises: what is Agile Service Management? The answer is surprisingly simple.

Simply put? Agile management is applying the agile mindset to IT Service Management. Nothing more, nothing less.

In Chapter 1, I explained what agile is and which principles are at the base of the mindset. The 4 values of agile software development just need one adjustment to apply to IT Service Management:

- Individuals and interactions over processes and tools.
- Working ~~software~~ services over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

The idea is that you keep to these principles when designing and delivering services. Sounds straightforward. And in a sense, it is. But how do you apply this in practice? And how do these agile principles relate to the framework that has been a golden standard at IT departments for decades: ITIL?
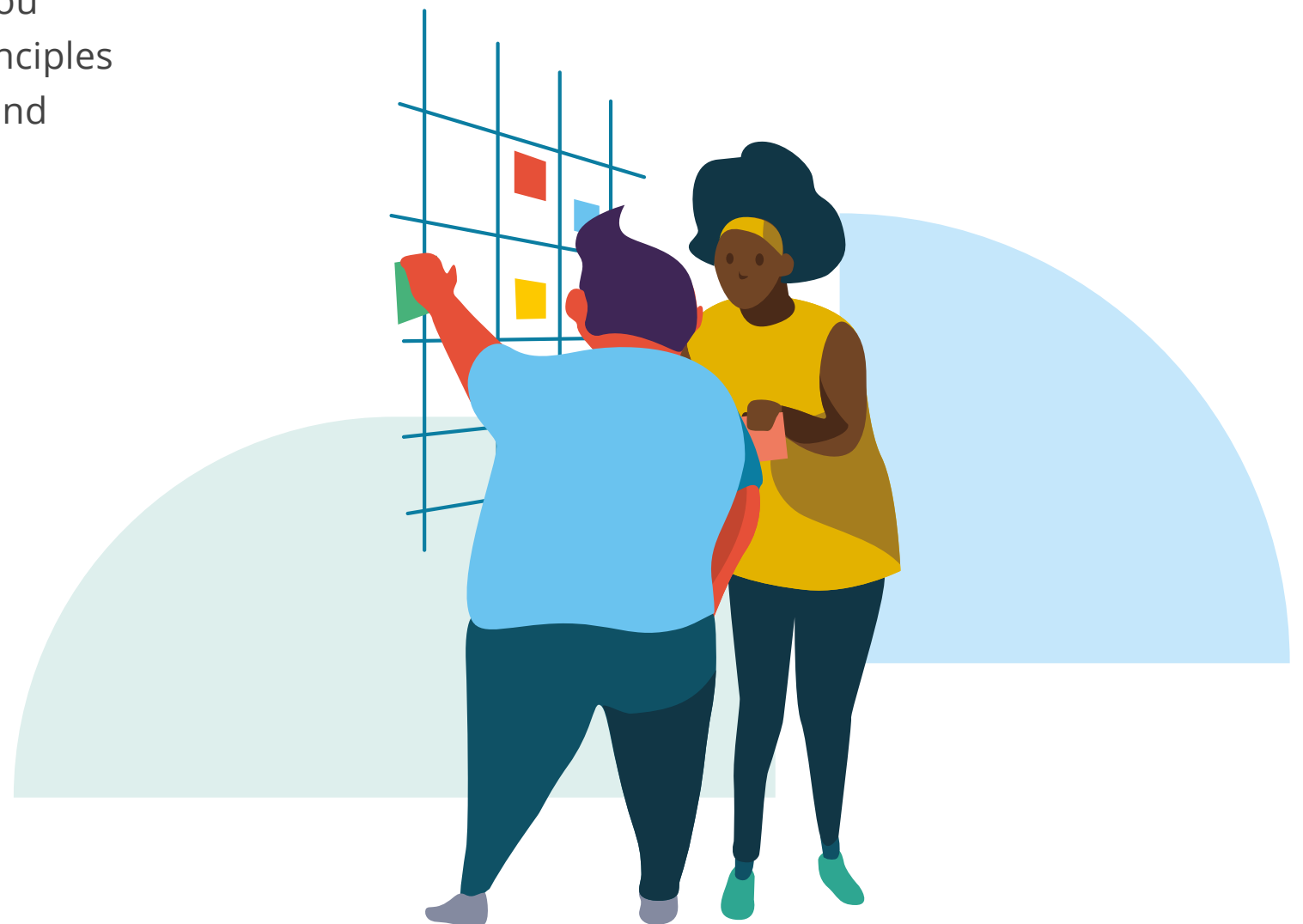
4. Agile Service Management and ITIL

In their origins, Agile and ITIL might look like totally opposing concepts. But is a happy marriage between Agile Service Management and ITIL possible?

Is it possible to pair Agile Service Management and ITIL? Absolutely, if you do it right. Compared to previous versions, the latest iteration of ITIL, ITIL 4, is focused less on prescriptive processes, and more on guiding principles for delivering value. Sounds pretty agile, right? But the fact is, most organizations are still built on older versions of ITIL, which don't offer as much flexibility.

This outdated approach to ITIL attaches great importance to everything the agile mindset believes to be less important:

Individuals and interactions over processes and tools.
While ITIL 4 has moved towards guiding principles and away from processes, ITIL implementations at organizations tend to still focus on process descriptions and systems. The goal is to get a steady quality of services. It shouldn't matter who supplies the service.

Working services over comprehensive documentation.
For most organizations, ITIL still often goes hand in hand with extensive process documentation. It took 5 books with a total of 1,300 pages to explain the 26 ITILv3 processes, after all.

Customer collaboration over contract negotiation.
Laying down contractual agreements and meeting them is an important part of how organizations implement ITIL Service Level Management. Meeting SLAs is one of the main goals for many companies and it's often the most important criterion that managers or customers use to judge the IT organization.

Responding to change over following a plan.
Older versions of ITIL were all about predictable processes. The idea: if you think out all the steps in advance and execute them accordingly, you'll always have the desired outcome. In many cases, the change management process is watertight and there's no room to deviate from the original plan.

"ITIL 4 is more agile than ever. But most organizations are built on older versions of ITIL, which don't offer as much flexibility."

ITIL is becoming more agile

While older versions of ITIL might have seemed incompatible with Agile, ITIL has become more flexible over the years. Unlike ITIL v3, ITIL 4 doesn't tell you exactly which steps to take to deliver a service but focuses more on why you're delivering that service – to provide value to your customers. The 7 guiding principles of ITIL 4 can help you determine how to deliver a service that works best for your organization.

ITIL itself has become more agile. You can see this in the move from process to guiding principles and in the principles themselves: Focus on value, progress iteratively, keep it simple – these are values that clearly resonate with the agile mindset.

An agile approach to ITIL

Approaching ITIL with an agile mindset is more straightforward than ever. You can implement the ITIL practice Incident Management with the agile mindset, for example. This means you pick the best option for your organization for each part of the setup. ITIL is actually quite suitable for an adjusted implementation. While ITIL does have a reputation for being rigid and unnecessarily complex, that was never the starting point. The idea behind ITIL wasn't that organizations would implement every aspect of ITIL to the letter. The message of ITIL was always: make sure to apply the framework in a way that best suits your organization.

5. Agile Service Management in practice?

It turns out agile and Service Management go together quite nicely. But how? How do you translate the agile philosophy to actual changes in your work? Here are 6 examples.

There is of course much to learn from other organizations. According to the Agile Service Management principles by Dolf van der Haven, I'll give you 6 practical examples on how to make your Service Management more agile.

1. Make sure everything you do adds value for the customer

IT departments too often put a lot of work into things that have little value for their customers. I recently visited an organization where the IT department had written an extensive manual for a new smartphone they offered. Sounds useful, but most of this information was already available on the internet. And the next OS update is going to make their manual outdated. When designing services or processes, service organizations make a lot

A more agile way of documenting is to keep the information in your manual limited to what is strictly necessary and first give these instructions to a small test group. Only describe company-specific information, like how to synchronize your work email with the new smartphone. Do you receive questions from your test group? Update the documentation before you officially start supplying the smartphone.

OS

2. Always work closely with your customers

When designing services or processes, service organizations make a lot of assumptions about the needs of their customers. An example: for years, a facilities organization encouraged their customers to log a call when something was wrong in the office building. They recently discovered their customers found it quite annoying to receive five to six status update emails after they logged a call. That was the reason many customers decided to stop logging calls altogether.

In Agile Service Management, you involve your customers often and as soon as possible with everything you do. This way, you avoid working based on assumptions. The organization from the example has come up with a solution together with their customers. When customers log a call, they can tick a box saying they want to receive status updates. One question and a single checkbox could have spared five years of frustration.
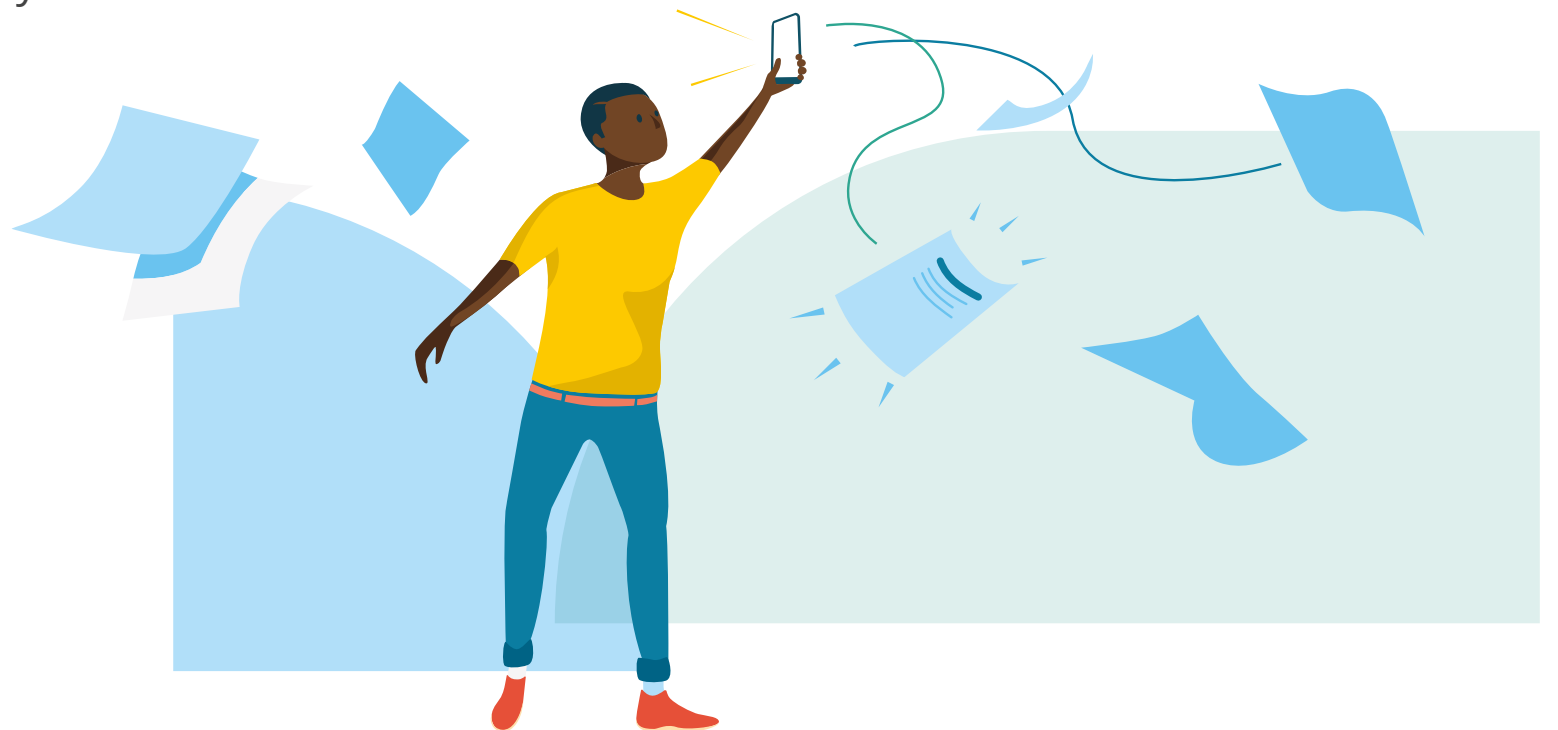
3. The right people in the right place

Many IT organizations lean heavily on processes. The goal of working with processes is to guarantee a consistent quality of services, no matter who supplies the service. Sounds good in theory. In practice, it does matter who supplies the service. An unmotivated service desk employee probably leaves a less positive impression on the customer than a happy, motivated employee. You can't cover this difference with a process.

An important part of the agile mindset is having enough time and attention for your team members. Your team only functions well with people who are good at the work they do, and when the work they do makes them happy. Is a team member no longer motivated? Talk to him or her. Maybe they're happier in a different role.

4. Make your processes as flexible as possible

ITIL processes can be flexible, but they often aren't implemented flexibly. Take change management. A Request for Change needs to go through a set number of predefined steps. The only choice you have in the process is approve or decline. There is no room to change plans. If you want to change them, you need to stop the process, make a new plan and get approval. Make sure that the processes you design are flexible enough to deal with ever-changing demands. This doesn't mean you need to implement every single change during the process. It does mean that you leave room for your team to deal with the processes in the way that brings most value for the customer.

5. Design, implement and improve your services step by step

New software or services implementations can take up months, years even. When the implementation is finally done, you've gained so many new insights you probably want to change everything. But by that time there's no more budget left, the project team members have moved on and it's up to the application manager to process all the feedback on her or his own.

Delivering new services in an agile way means you deliver something workable as soon as possible, collect feedback, and use this feedback to improve the product. At TOPdesk we do software implementations step by step. We first set up a basic version of the Incident Management process, and as soon as it's operational, we go live. The process isn't perfect, but we're okay with that. While we continue working on the next process, we receive feedback to improve Incident Management.

6. Keep your services and operations straightforward

Request processes often contain a lot of unnecessary management authorizations. The IT department assumes that management wants control over every individual request. Or the IT department doesn't carry any responsibility. This makes for a process full of authorizations and a manager who gets loads of emails with authorization requests.

The process shouldn't be this cumbersome. It works better when the IT department asks the managers how much control they really want or need. They usually don't really want to receive all those emails. An alternative solution: requests don't need to be authorized, and the manager receives a monthly overview of the costs. This way, the manager still has control and an overview, but he or she doesn't have to process a lot of emails. And the employee is helped quicker.

Tip: just get started and keep it small

I sometimes get the question: where do I need to start when I want to work more agile? To be honest, that differs per organization. Take a close look at your current services and compare this with the agile principles. Where's the friction? And what improvements are easy to implement? Start there. Make a small improvement. Ask for feedback. And move on to the next improvement.

6. Making your Incident Management more agile

Incident Management is the most important process of a supporting department. It's also quite straightforward. Is there anything you can do to make incident resolution more agile? Yes, you can. I'll tell you how.

Remove all steps that have no customer value

Are you thinking of making your incident process more agile? Your first step is to take a critical look at your current process and evaluate each step by thinking: does this step add value for the customer?

To answer this question, you first need to know what your customer wants. Your customer probably wants a quick and good solution. Every step in your incident management process needs to contribute to how quickly an incident is processed. Or to offering your customer a better solution. Is there a step that doesn't contribute to your goal? See if you can remove it.

You can compare it to the lean approach. Here, you eliminate everything wasteful from your process. The difference is, with lean your goal is to make the process more efficient, and with agile you want to add as much value as possible for the customer.
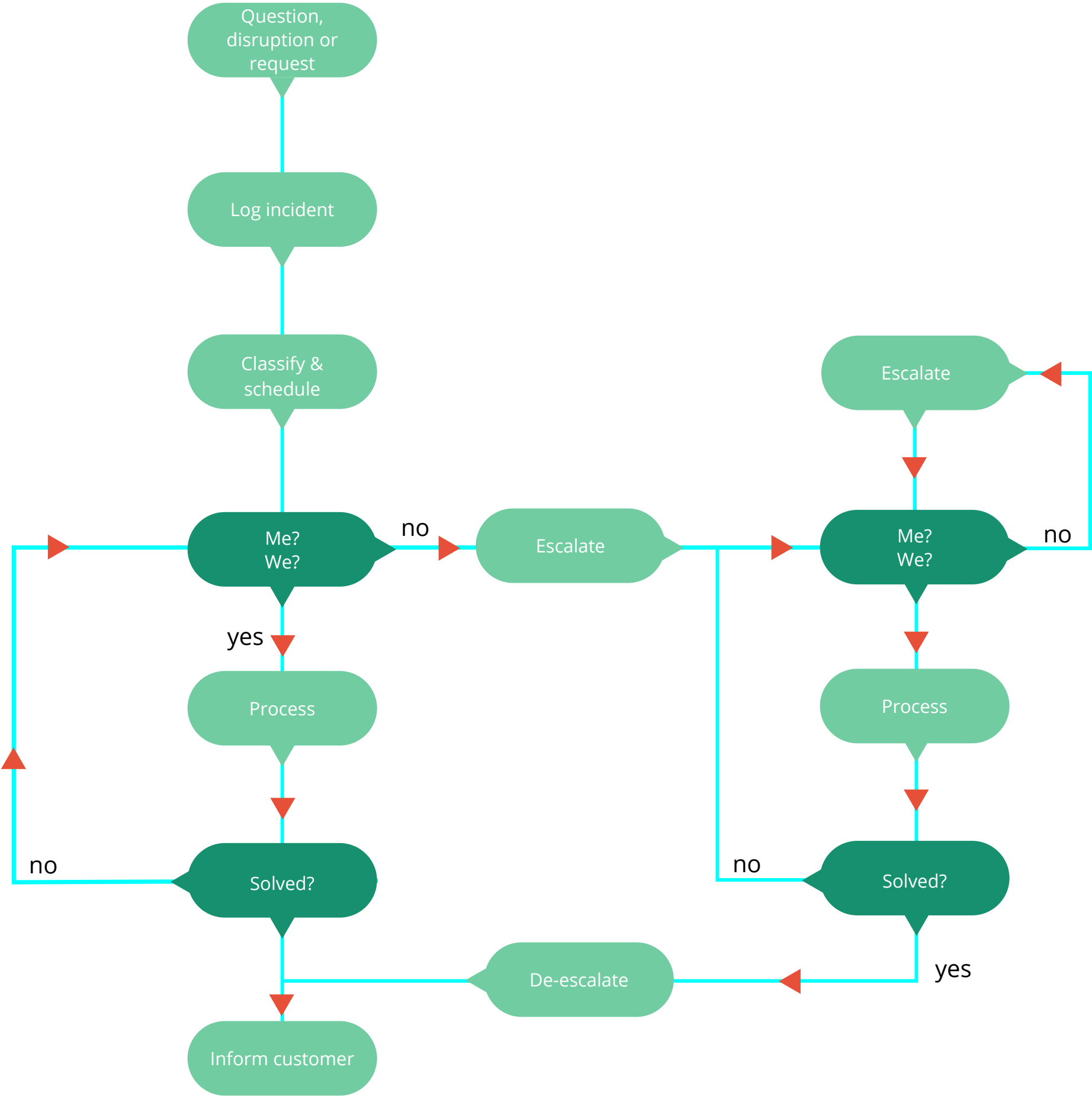
A traditional incident management process

Let's have a look at a regular incident management process.

A customer logs an incident. A service desk employee adds information such as classification and scheduling to this incident. Then, he or she checks to see if he or she can process the incident. If the answer is yes, he or she processes it. If the answer is no, the incident is forwarded to a specialist. The specialist makes the same assessment. When the incident is solved, it is returned to the service desk. There, the incident is closed and the customer is informed.

Looks quite straightforward, right? Well, it can be even easier.

You just need to ask yourself 2 questions.

```
Question, disruption or request
        │
        ▼
   Log incident
        │
        ▼
Classify & schedule
        │
        ▼
   Me? We? ──no──► Escalate ──► Me? We? ──no──► Escalate
        │                          │
       yes                        yes (no)
        │                          │
        ▼                          ▼
    Process                    Process
        │                          │
        ▼                          ▼
    Solved? ──no──►            Solved? ──yes──► De-escalate
        │                          │ (no)
        ▼                          
 Inform customer
```
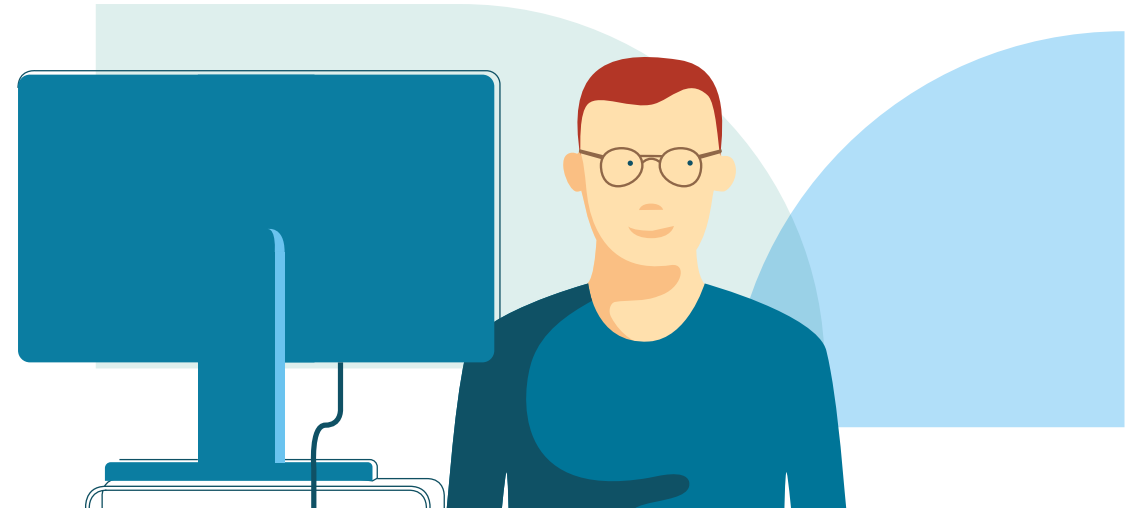
1. Why does your service desk need to process every incident?

In many organizations the service desk closes each incident, even if the back office solved the problem. The back-office specialist describes what he's done, the service desk translates his technical story to something understandable, closes the incident and informs the customer. Why? Because people assume that technical specialists aren't strong communicators and they can't describe their solution in a customer-friendly way.

This bothers me. If a technical person can't describe his solution, how does the service desk know what he did? A service desk employee needs to go through all the information to piece together an understandable explanation. A waste of time, in my opinion.

Why shouldn't the back office be able to write understandable solutions for their customers? I understand that it might be difficult for some, but it's easy to learn. Have a training session and coach each other. It's a lot more efficient if the back office can describe and close their own incidents, as opposed to having the service desk figure out what to write to the customer. Plus, the incident needs less forwarding, which makes for a shorter duration.

2. Does your service desk really need to fill out all fields?

For each incoming incident, you need to fill out a lot of data. But is this really necessary? You only need to know a few things to process an incident. The caller, the question and a deadline for the call. The rest of the data is only used for reports.

Go through all filled-out data with the service desk and management to determine if you really need it for reports:

Are there reports for this data?
I've visited dozens of organizations where the service desk filled out all kinds of fields, but they never

ended up in a report. How does this happen? During the implementation, some people might have thought: 'Let's fill out these fields, then we can report on them later'. However, the reports were never made.

Does someone read these reports?
What also happens: there are reports on certain data, but no one reads them. I once visited an organization where the IT department sent monthly incident reports to management. They suspected no one read the reports, because they never got a reply to their emails. That's

why they added the sentence 'Whoever reads this can come and get a pie at the IT department' on the second page of the report. Guess what? No reaction. After a couple of months, they completely stopped sending the report. No one ever asked what happened to it.

What happens with the reports?

Imagine there are reports on all the data you filled out, and that these reports are read. Then the question arises: do we actually do something with this data? Does management make improvements based on these reports? Do these improvements help contribute to your overall goal: find a suitable solution for your customer as fast as possible? If so, are these improvements important enough to fill out all this data for each incident?

Don't get me wrong. I'm not saying reports are useless. On the contrary, reports can give valuable information about your Service Management process. But you need to stay critical. Is some data not relevant? Don't fill out the fields, or have them filled automatically. This saves the service desk a lot of registration time.

7. Making your Service Level Agreements more flexible

While SLAs are great for reporting to management, relying on them too heavily will risk long-term damage to your organization. You get blinded by numbers and forget about the service. The solution? A shift towards XLAs.

What is an XLA?

If your Service Desk was a fruit, what would it be? An apple? An orange maybe? Try a watermelon. It may sound ridiculous, but humor me for a second! The Watermelon Service Desk was first used by Marco Gianotten of Giarte for an SLA-focused Service Desk. The dashboards are green and management is thrilled. Yet underneath simmer red warning signs of resentment and dissatisfaction.

A 5 minute average response time and 8 hour closure rate sounds fantastic. But SLA stats miss something: the experience of the user. Not delving beneath the shiny green exterior could be causing your business harm.

The value of XLAs

XLA is a concept developed by Giarte and the X stands for eXperience. It means that performance is dictated by the one person who feels it the most: the customer. The good thing is that generally, you need less XLAs than SLAs to give an indication of performance, which makes them easier to manage.

If you are planning to adopt a more agile way of working, then XLAs will compliment this perfectly. XLAs naturally focus on interactions and customer collaboration rather than cumbersome contractual obligations. The XLA is also very susceptible to change. When the needs and requirements of your customer change, your XLAs adapt too.
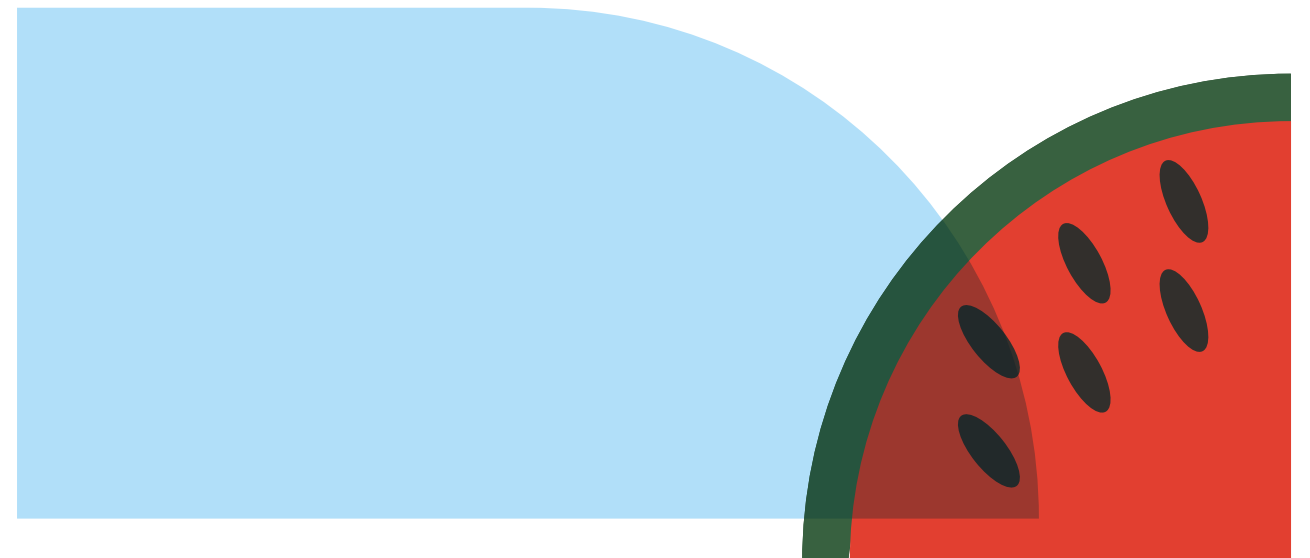
"Your service desk SLAs are like a watermelon."
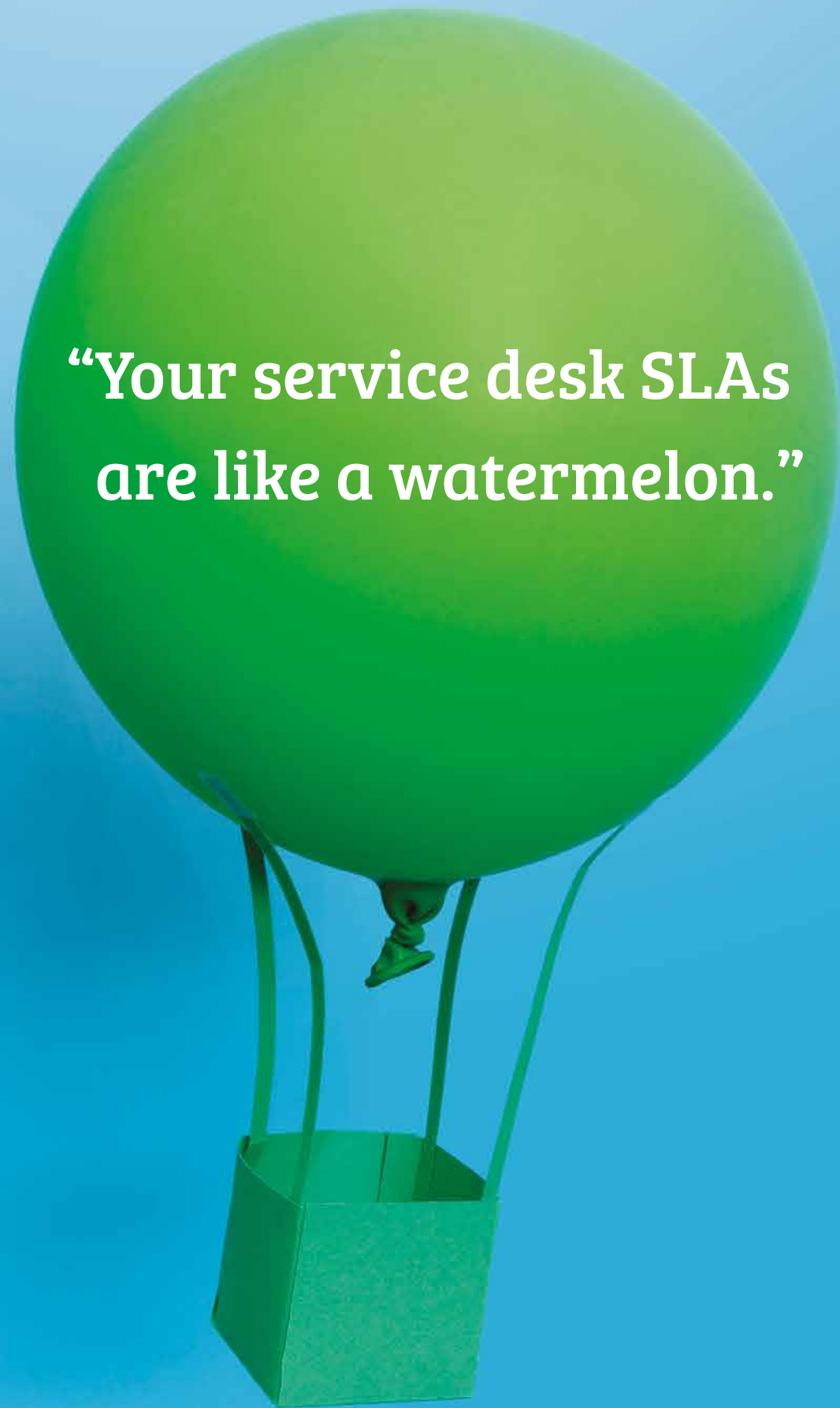
Where do I start?

There are hundreds of methodologies to measure customer experience. Start out by keeping it simple and using a single question to capture the customers experience when a call is closed. Star ratings are a useful one-click response that already give you insights into the services previously unseen by traditional SLAs.

But there is more to it than metrics. XLAs represent a change in culture by shifting the focus from performance to the experience of the customer. You have to get your operators on board through persona creation and mapping the customer journey. Over time you will notice an interesting shift; you might start missing the odd SLA target, yet your customer experience continually improves. The question is raised: what are we using all these SLAs for?

Streamlining your services

If you have left some of your heavy SLAs behind and you find yourself with a few simple XLAs that keep your customers happy, you will find yourself less like a watermelon and more like a grape: bite-size, and green both inside and out.

8. Seven most common pitfalls in agile transition

Considering shifting to an agile work environment? There's no script you can follow. However, there are experts who can help smooth the transition. Steven Happee is one of them.

Steven Happee's love for agile blossomed in the nineties, when he developed software in a small, multi-disciplinary team, in a way that would now be called agile: a lot of contact with customers, quick value delivery, and transparent work processes.

Now, twenty years later, he helps organizations make the transition to become more agile and learning-oriented. He uses his extensive experience as product owner, Scrum master, developer and manager to conduct valuable experiments. Based on his experience, Steven shares the 7 most common pitfalls for agile transitions.
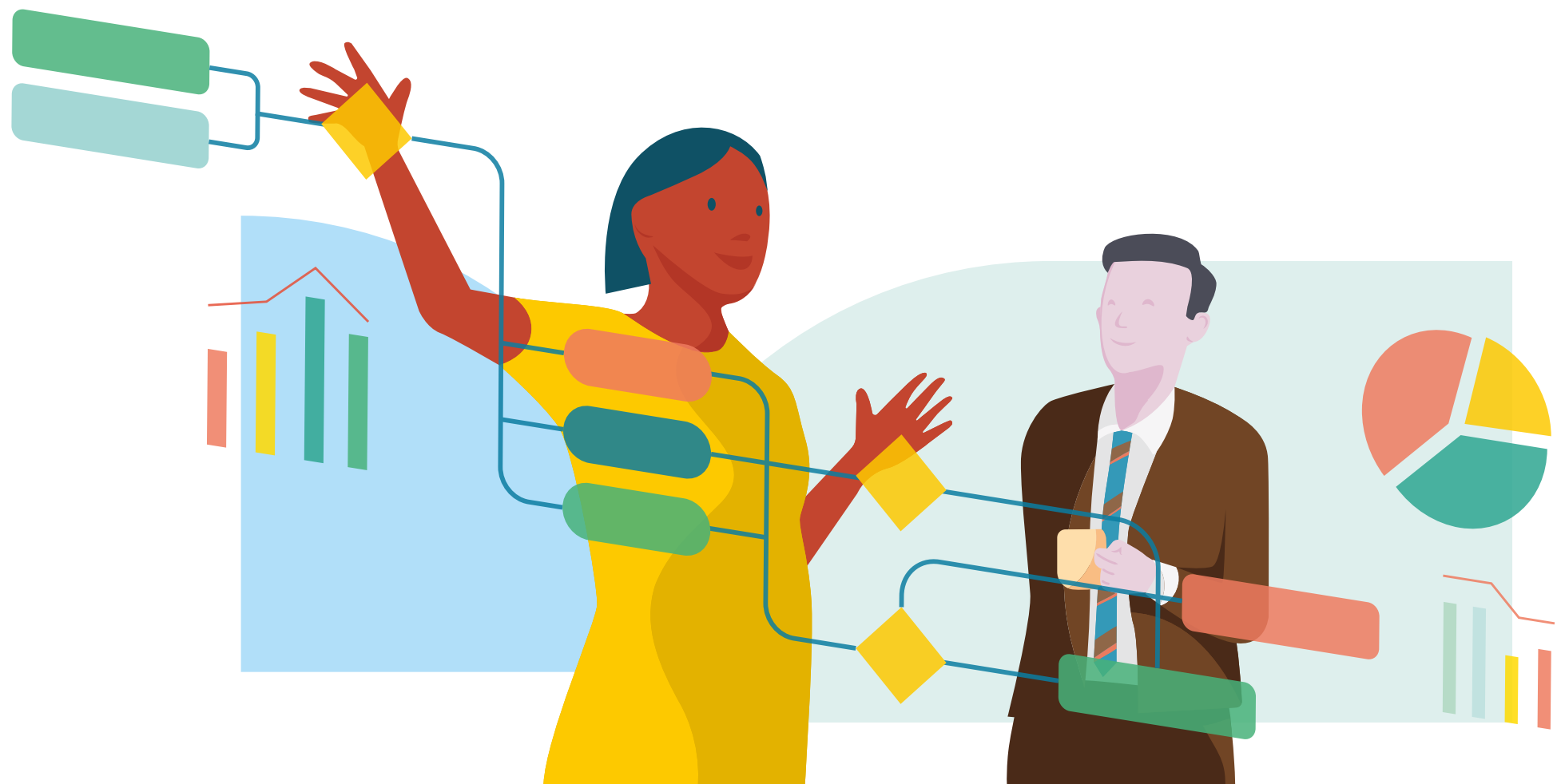
1. Only look at the agile transition top-down

You can approach agile transformations in two ways: top-down and bottom-up. A bottom-up approach means the team takes initiative, often in IT, to implement a more agile way of working with a single team.

For top-down, the initiative comes from the manager, CEO or CIO and the goal is often to make a large part or even the entire organization more agile. The top-down approach is relatively new. These past few years, agile has become a hot topic and more and more CEOs and CIOs are thinking: 'Should we do something with agility?' A good idea, but it doesn't work if the organization isn't with them.

A bottom-up approach is usually more successful. This is something you see in IT teams, where the agile mindset comes from originally. Someone in the team wants to experiment with agile, his manager is enthusiastic and a scrum team is formed. When this team performs well, more teams might follow and the organization becomes curious.

An approach that combines bottom-up and top-down works best: a team takes the initiative to implement a more agile way of working and finds a sponsor at the top of the organization.

2. Using the waterfall method for your agile transition

Ten years ago, there were a lot of people who believed that you needed to use Prince2 for an agile transition. First you make a design, implement it step by step, create milestones, etcetera. This a paradox of course.

Fortunately, most people these days are convinced that an agile transition needs an agile approach. The transition to agile is a complex project after all — meta-agile as you may call it. People start to work differently together, your way of working changes and your tools are different. It's best to have an iterative transition with a multi-disciplinary team and a change backlog. An added bonus is that you set the right example.

3. Being too strict with agile techniques

Agile frameworks like Scrum offer techniques to apply an agile mindset to your daily work. A backlog, retros — they're all very visible. But these techniques are just the tip of the iceberg. It's the visible aspect of agile working. They are based on certain assumptions on how to organize your work, and these assumptions fall back on principles from the Agile Manifesto or Modern Agile.

Some people have the tendency to be very strict with these agile techniques, but without understanding the underlying principles. That can cause friction. Some techniques might not work for your organization, so it would be a shame to follow them religiously. In the end, you need to embrace the agile values, not the techniques. That's why it's good to explain the underlying principles to someone who wants to do something with agility, so they can fall back on them.

"An agile transformation impacts all processes and people in your organization."
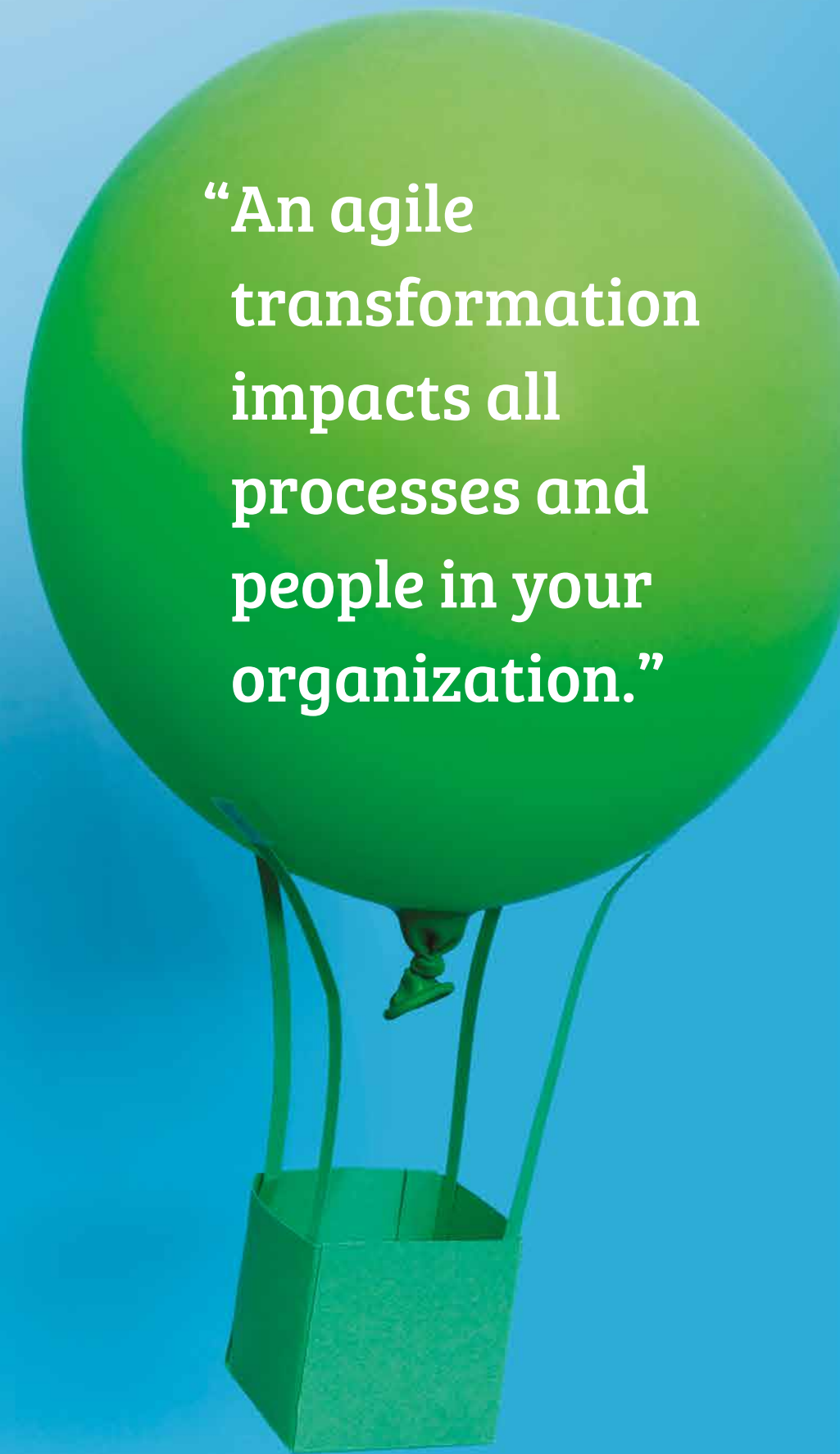
4. Creating variations on existing techniques too quickly

Yes, you need to make agile techniques your own. But not from the very start. The agile world often refers to ShuHaRi, a Japanese martial art concept on how to make something your own in 3 phases. You start by strictly following the existing technique (Shu). When you've internalized the rules, you can make variations (Ha), until you reach Ri: you're subconsciously competent and there is no longer a need for rules or techniques.

You often see that teams start varying on existing techniques, or collect elements from different frameworks: cherry picking. You hear a lot of excuses for this: 'I know the theory and I know it's not going to work for us' or 'We like to experiment, so we're combining some different techniques'. It's good to experiment, but make sure to vary from a solid foundation. Work with an existing method for six months or one year and experience what it's like. How do sprints work? What is a good retro? How do we formulate customer value? Only when your foundation is solid, you can start varying.
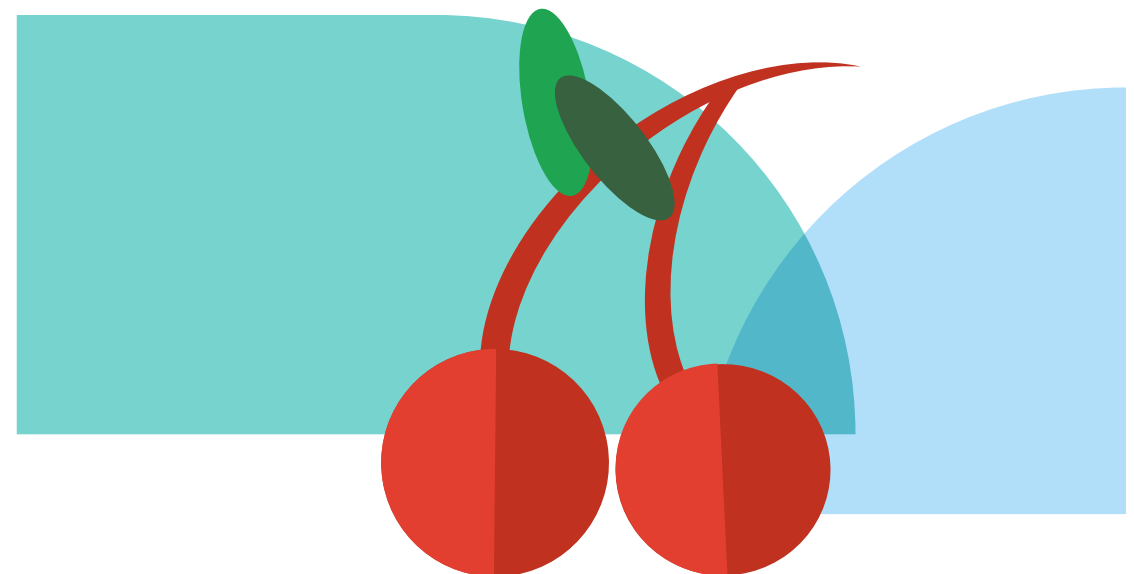
5. People working in an agile and project team

Agile working and project-based work have fundamentally different starting points. For agile you have the same teams, with team members who know each other's strengths and keep evolving together. You give the teams work. With project-based work you have work and you search for a temporary group — you bring the teams to the work. This cannot be combined with each other.

6. Think of an agile transition as an IT party

What often happens, is that people have limited views on the impact of agile working. Agile is not limited to the IT scrum team. An agile transformation is a cultural shift that impacts all processes and people in your organization, from Finance to HR to management. When teams become self-managing, what changes in the role of team leads? How do you assess people in scrum teams? Are yearly budgets and scrum teams a good match? In most cases, more and more teams in the organization want to start working more agile as well.

The sooner you understand that an agile transformation impacts the entire organization, the easier the transition. Many managers see agile as a chance to implement a culture shift, like working more customer-oriented or more ownership from employees. Managers have often tried a lot of methods to change the organizational culture. With agile techniques, changes like a customer-oriented focus or sense of ownership are more a by-product of an agile mindset.

7. Scaling up too soon

There is a lot of experience in the agile world with agile working on a team level. But if the team is up and running, how do you go up to two or three teams? Or twenty? And how does agile work in departments like Sales or HR? That's a puzzle the agile world is trying to solve right now.

There are multiple frameworks for scaling up agile teams, like the Spotify model, SAFe (Scaling Agile Framework) and LeSS (Large Scale Scrum). The most important lesson: if you don't have to scale up, don't do it. See if you can manage to let the teams work independently from each other. Don't start thinking about scaling frameworks until there is no other way. Some organizations want to scale up the minute the first people are enthusiastic. Don't do it. Only fix things when there's friction.
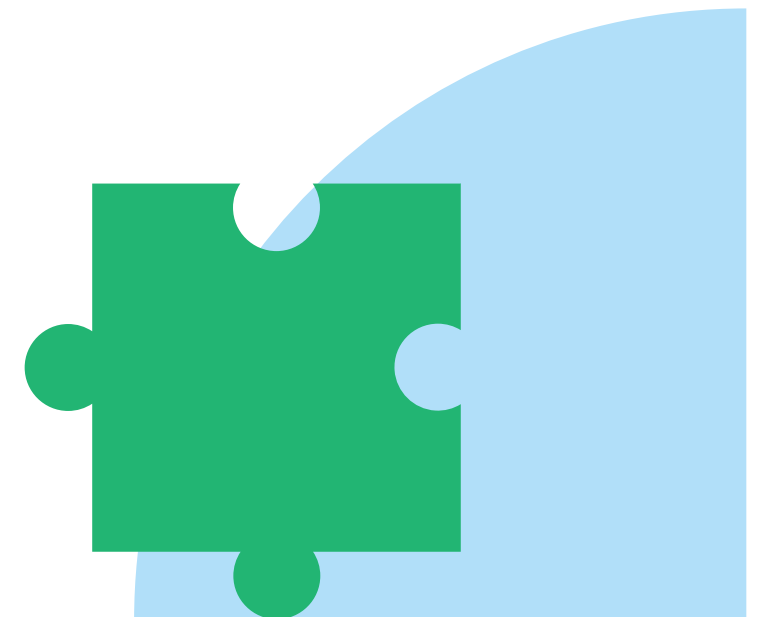
9. How to guide your IT department through an agile transition

Introducing an agile way of working means a culture shift at your IT department. Some embrace the change, others might not be so eager. What reactions can you expect from your IT employees? And how do you deal with these reactions?

How do your IT-team members respond to an agile transition? What do they like? What don't they like? And what will be their main challenge? Here to help: Mark van Meurs, Development lead at TOPdesk with extensive Scrum Master experience, joins Bas Blanken to help you answer these questions.
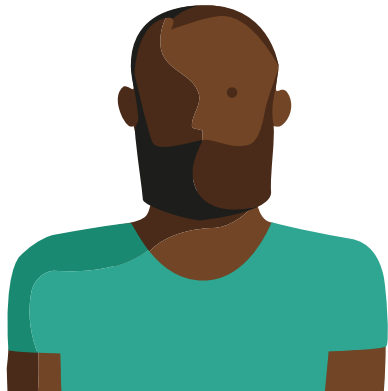
The back-office specialist: share your expertise

Bas: 'For an IT professional, an agile transition can have quite an impact. Every back office has a system or network manager who'd happily spend their days behind their desk, solving complex puzzles. They are often the only expert in a specific field. This very valuable. But working more agile means that the entire team becomes responsible for solving these puzzles. This requires the work to be transferable. No need to tackle it yourself.'

Mark: 'The big challenge for a back-office specialist is sharing knowledge, so the work becomes more transferable. IT colleagues need to open up, communicate more and explain to the team what's going on and why they choose specific problem-solving strategies. This is not always easy, because a specialist is often proud of being the only one with this type of knowledge. As an IT manager, you need to show your staff that they can add even more value by sharing knowledge.'

Bas: 'You don't have to transfer all your knowledge at once. In most cases this isn't even possible because the subject matter is complex. Start by encouraging specialists to talk about what they're doing. Other team members will start asking questions and before you know it, the first tasks have become transferable.'
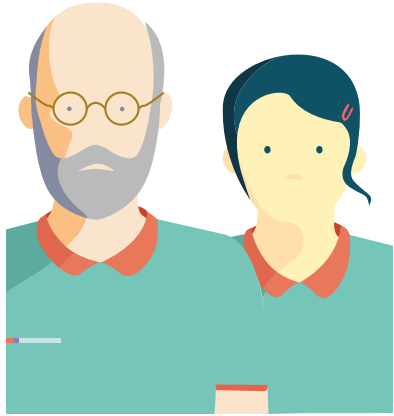
The service desk employees: get knowledge from your colleagues

Bas: 'An agile way of working often suits service desk employees, because the agile philosophy is aimed at helping your customer the best you can. This philosophy fits their natural curiosity. However, transitioning to the agile mindset can still be a challenge for service desk employees.

In a lot of organizations, I see that service desk employees feel responsible for the calls they can solve on their own. Can't solve anything? They forward the call to back office and wait for it to return. With agile working, the entire chain becomes responsible for all calls — front office to back office. This means service desk employees also become responsible for calls they can't solve.'

Mark: 'The goal of the service desk is to solve as many first-line calls as possible. For service desk employees, this means: when you can't solve a call yourself, you need to get the knowledge you need to solve it next time. Be entrepreneurial and curious. Sit next to an expert to see how you can best help the customer. See if you can solve frequent calls yourself by getting the knowledge you need.'

Bas: 'Working more agile often also leads to a better understanding between the front and back office. I often hear frustrated front office employees say that the back office communicates poorly, while the back office accuses the front office of giving them incomplete calls. When you talk to each other more often, you understand what the other needs to do his or her job.'

IT manager: give your team freedom and trust

Mark: 'Introducing agile means changing the culture at your department. When you work agile, you don't plan or design everything in advance. Instead, you leave more room for experimenting. Even a failed experiment provides valuable insights. Create a culture where team members are OK with making mistakes.'

The most important task of the agile IT manager is to create an open work environment. Allow your team members to make mistakes and encourage them to be honest about it. Never judge them on failed experiments. Give the team the trust and freedom to determine what to work on.'

Bas: 'IT managers that give their team little breathing room will have trouble with the transition to agile. Micromanaging initiatives can run a new initiative into the ground before it's had a chance to properly take off. Make your agile teams thrive by giving new projects time to grow and develop.'

Mark: 'Is there an "us versus them" mentality at the department? Then the manager needs to get rid of this mentality. In agile, everyone works on attaining the same goal: helping your customers. There is no room for islands.'
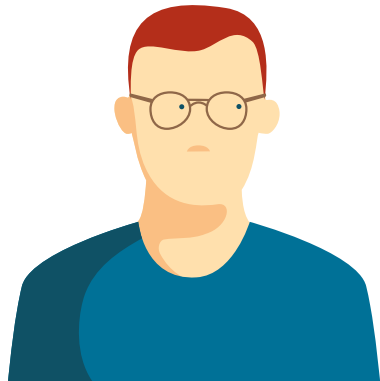
Workspace manager: contact the service desk

Bas: 'Working agile improves communication between Workplace management and the service desk. A complaint I often hear from workplace managers, is that they receive incomplete information from the service desk. They get a request to fix a certain device, but they can't see where the device is. The workplace manager needs to spend time searching for it, or just reassigns the request to the service desk. Both a waste of time.'

Working more agile means that the service desk and Workplace management see more of each other, for example. Instead of sending calls back and forth, the workplace manager needs to go to the service desk more often to have a look at a call together. This takes some effort, because the departments are often not on the same floor or location. The result however is that calls are processed quicker and better.

Mark: 'Working more agile gives most workplace managers a nicer job. They work with others to find a solution for the customer. No more daily task lists, but insight in and influence on the end result.'

Want to know more?

Introducing an agile mindset is one way to improve your IT Service Management. But there's much more you can do. Subscribe to our blog and get tips for better service management every week.

Visit **blog.topdesk.com**

Do you have questions or ideas after reading this e-book?
I would love to hear from you!
Feel free to send me an email via info@topdesk.com.

Hope to hear from you!

Bas Blanken

TOPdesk