

Less Friction, Fewer Tickets

A better service desk starts upstream

Jan Massier · SITS Netherlands · 22 September 2026

Less Friction, Fewer Tickets - A better service desk starts upstream

1. **IT is not getting any easier. The pace of change keeps rising, and IT is getting more complex, not simpler.**
2. **If the service desk wants to keep up, and still improve its service to users, it will have to move upstream. That is where the gains are, not in the ticket flow itself.**
3. **Because there is more to gain than your KPIs alone. And that gain is not a goal in itself: it's what you need to keep up.**
4. **In the next thirty minutes I'll show you where that upstream gain sits, how to see it, and what to do about it tomorrow morning.**

About me

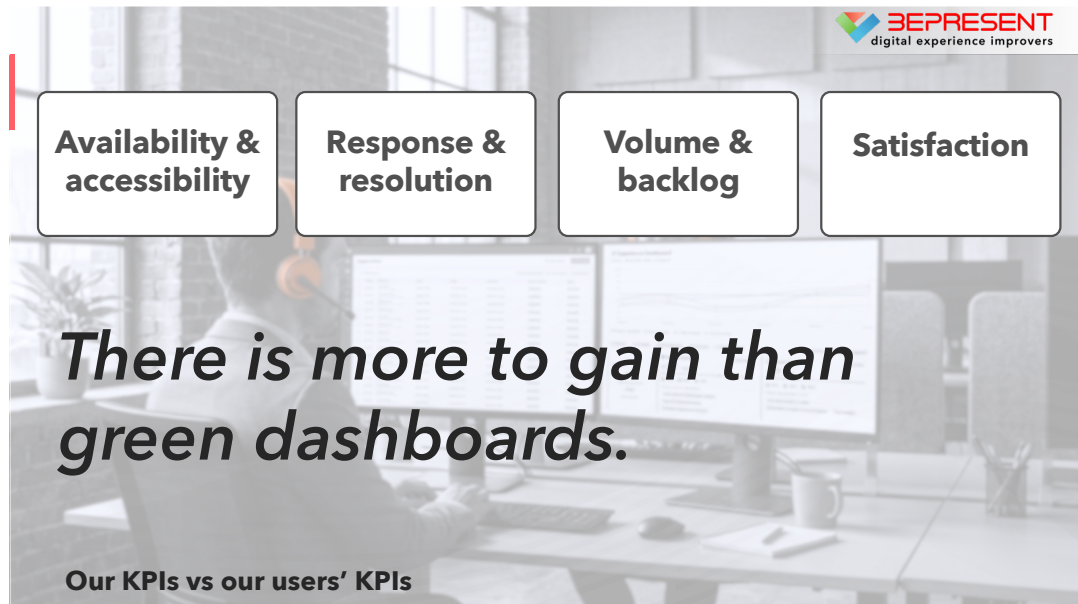
In 1995 I started on a service desk, at Apple. Headset on, phone ringing, the same question fourteen times a day. So when I talk about what a service desk sees, I'm not talking from the outside.

I then worked for an international IT consultancy, and from that moment on I have spent the rest of my career on the part of IT that never makes it into the project plan: the people. Implementation, communication, adoption. Grounded in what is known as the ABC of ICT - attitude, behaviour, culture of ICT: a predictable world in which IT development (BUILD), IT operations (RUN) and IT use (USE) don't go hand in hand, and why that is.

For 25 years I have watched organisations build well-intended systems, and then be surprised that those systems are not used, or not used well: no positive business case, a lot of lost efficiency, grumbling users with poor IT experiences.

I am ITIL- and XLA-qualified and work with a self-developed, proven and highly effective adoption approach, MoC4IT™, built on Kotter, ADKAR and ITIL.

That's the formal part. More important is this: everything I'm going to tell you here, I first got wrong myself.



The ambition of the service desk usually comes down to one question: are we meeting our SLA KPIs?

- **Availability and accessibility.** Service availability (e.g. 99.5%). Desk opening hours and accessibility per channel. Average wait time / speed to answer (e.g. 80% within 30 seconds). Abandonment rate percentage of callers who hang up before contact
- **Response and resolution - response time, resolution time, FCR, reopen rate.** SLA compliance: time to first response. Time to resolution per priority. % of tickets handled within the agreed time. FCR / First Line Resolution rates. Escalation rates. Reopen rates.
- **Volume and backlog - tickets resolved, tickets open, handling time, cost per ticket.** Ticket volumes (per period, category). Backlog / open tickets and their age. Average handling time per ticket. Cost per ticket
- **Satisfaction.** CSAT - satisfaction score after handling. NPS - "would you recommend us?"

Almost all of them are **output measures**. They measure how quickly and how neatly the desk cleans up, not **whether the mess should have existed in the first place**.

CSAT is the only one that asks the user anything, and then only about the contact itself.

That is the gap this talk is about: our KPIs are not the users' KPIs.

Because **there is more to gain than a green dashboard**. And that gain is not a goal in itself. **It is necessary, because IT is not getting any easier.**

The number of systems an employee has to be skilled at keeps growing, and the pace of change keeps rising.

If we keep working the way we work, people fall further behind.

I'm going to talk about **the two biggest risks** that make that happen: **lack of adoption** of new IT, and **digital friction** in existing IT.

Manage both well, and you eventually free up service desk capacity. That is not a saving. It is what you need to keep up.

A service is not what we deliver.

VOCR

A service is a means of enabling **VALUE** co-creation by facilitating **OUTCOMES** that customers want to achieve, without the customer having to manage specific **COSTS** and **RISKS**.

IT Service • VOCR • Business value

A service is not WHAT we deliver. It's what the user ACHIEVES with it.

A service is not output but value, and value is created in Use.

ITIL defines a service as enabling the outcomes the customer wants to achieve. Availability, resolution time and ticket counts are outputs. Value is only created the moment someone gets their work done with it - not at go-live, not at deployment. A service that is live but used with workarounds delivers no value, however green the dashboard.

So, it is **our job @IT to ensure we enable that value by facilitating the promised outcomes. This is about providing flawless services. And we are at the forefront of it (IT).**

The easiest way to explain VOCR to someone without an IT background is the taxi analogy:

When you take a taxi from A to B, you are consuming a service:

- Value: You arrive at your job interview rested and on time.
- Outcome: You have been moved from location A to location B.
- Costs: You pay for the ride (one clear price). You are not burdened with the specific costs of road tax, insurance, fuel and maintenance.
- Risks: You don't carry the risk of the engine breaking down on the way, of having to find a parking space, or of arriving exhausted from sitting in traffic.

Those risks sit with the driver and the taxi company.

Friction on a taxi ride - the ride still happens, but you're doing part of the job

- You have to hold the door shut, otherwise it swings open at every corner. (workaround)
- The driver doesn't know the way, so you navigate from the back seat with your own phone. (missing knowledge)
- The meter is broken; the driver names a price at the end. (no transparency)
- You have to explain three times where you want to go, because the driver keeps repeating the address back wrong. (no understanding)
- The card reader doesn't work; you have to find an ATM first. (failure at the last moment)
- The driver takes the route he knows, not the fastest one; you arrive fifteen minutes later than planned. (the default path nobody optimised for the user)
- Halfway, the driver asks you to get out and push, "just for a second". (manual step in a process that was supposed to be automated)
- The seatbelt is jammed, so you spend the ride holding it in place. (security control you have to fight with instead of forget about)
- The air-conditioning only works if you know which button to press twice. (feature that exists but nobody uses, 'cause nobody told you)
- The driver tells you to call the company yourself if you want a receipt. (service desk that sends you elsewhere)
- You arrive on time but sweaty, irritated and rehearsing what went wrong instead of your interview. (outcome achieved, value lost)

People report what's broken.

Nobody reports what they didn't achieve.

Tickets $\neq$ friction

Every ticket is a decision to report.

It stands for a **moment when someone decided it was worth it**. Worth the interruption. Worth the form. Worth the wait.

Look at what that decision produces. **People report what's broken:**

- the error message,
- the login that refuses,
- the printer.

They also report what they don't understand:

- how do I do this,
- where do I find that.

These are technical or usage issues.

! What nobody reports that they didn't achieve the outcome.

Nobody calls the desk to say "I'm not getting the value this system promised me". That's not a ticket. That's a task that took longer.

That's the bigger layer, under the waterline.

- The workaround.
- The ten minutes of searching, three times a week, never recorded anywhere.
- The question to a colleague instead of the desk, because she had already figured it out.
- The grumbling at the coffee machine.

! An incident stops you. Friction slows you down, irritates you and quietly eats time - and precisely because it never brings you to a halt, **it never becomes a ticket**.

I get it myself every day. Yesterday: a login verification that e-mails me the code. Four minutes of waiting. No ticket.

Ticket data doesn't show you what goes wrong in your organisation.

It shows you what people thought was worth telling you.

And the layer underneath - the one that doesn't get recorded - is exactly where experience is shaped and where the demand for support is created.

One more thing: **the service desk itself is an experience too**: the wait, the transfer, the form, the ticket closed while the problem comes back next week.

For the user it's just more friction. For you it's the kind of friction you can and should fix. To deliver the value that was promised.

Friction is the distance between "it works" and "it's done".

digital experience • digital friction

Friction is the distance between OUTPUT and VALUE.

Friction is **when something demands so much attention that it distracts you from what you want to achieve with it.**

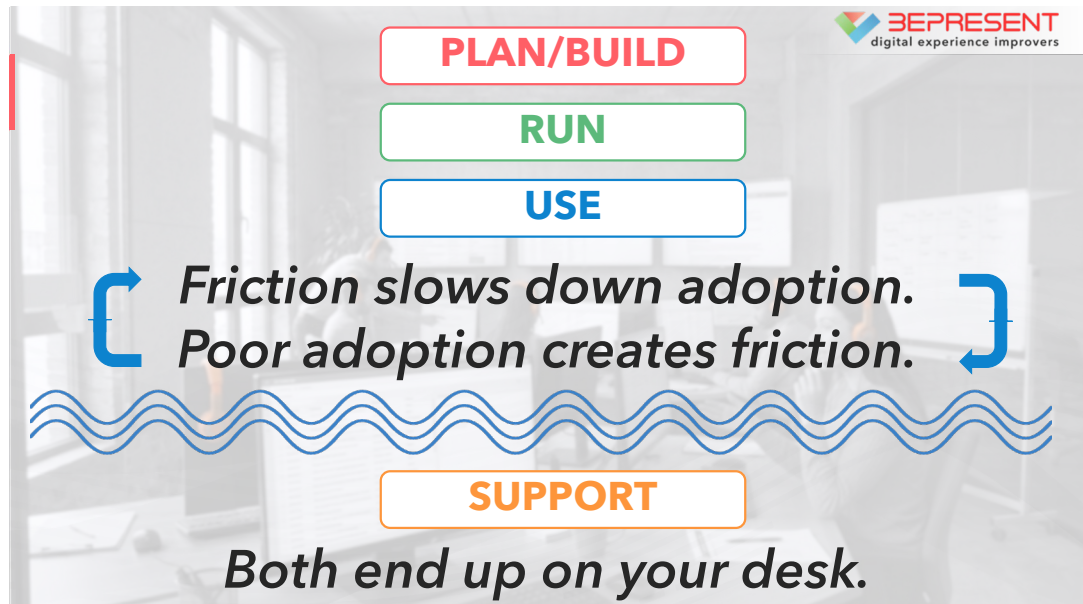
Everything a user has to do on top of a working service to reach their outcome: **the workaround, the help request, the explanation that is missing, the step that nobody understands.**

Examples of friction - nothing is broken, it just takes effort:

- A login that asks for a second password in two of the eight applications, even though single sign-on is "live".
- A verification code that arrives by e-mail four minutes later.
- An approval that takes three screens and two systems, so people batch them up and approve on Friday.
- An update rolled out overnight that moved a menu item; on Monday morning, nobody can find it.
- A form that clears itself after a time-out, halfway through.
- An error message that says only "something went wrong".
- A search function that finds everything except the document you need - so people ask a colleague.
- A screen that nobody understands without an explanation, so the explanation lives on a sticky note on the monitor.
- A spreadsheet next to the system, because the system doesn't do the one thing the team actually needs.
- A knowledge article written by IT, in IT language, for users who don't speak it.

Friction is not a failure and not an incident.

It is the measure of how far output and value are apart.



Friction, adoption and service desk demand are 1 system.

Your workload is decided before go-live. The demand for support rarely starts at the desk; it starts earlier in the chain.

- At **Build**, everyone who designs and builds the systems > where a screen is designed that nobody understands without an explanation.
- At **Run**, everyone who deploys and keeps them running > where an update is rolled out overnight and a menu has moved by morning.
- And above all at **Use**, where a **working service is used with workarounds** because **nobody looked at how the work is actually done**.

Of the 3, 'Use' is the phase without an owner: Build is finished at go-live, Run watches whether it's running, the business assumes IT takes care of it.

Friction slows down adoption: people who have to make an effort at every step use the service less, or differently than intended.

Poor adoption creates friction in return: people who don't know the service well get stuck on things that wouldn't have been a problem with the right knowledge.

The two reinforce each other, and **what comes out is demand: help requests, workarounds, late incidents.**

And **that demand lands on your desk.**

The service desk is the only party with daily contact with all users, across all services. It is the only place **from which you can see the whole system**. That defines its role: not the place where everything that went wrong upstream ends up, but the place where friction becomes visible first - and from where it is taken back to the phase where it was created.

Example 1. A large hospital. Every first week of the month, the same scene: the **desk's phones red-hot, the mailbox full, ticket numbers through the roof**. Predictable as a calendar. **Was anything broken?** No. Systems fine, availability fine. **The dashboards were green, the phones were red.**

What was happening: every month, dozens of new staff started at once and had to work immediately with complex healthcare IT they had never seen.

What did they get? An **e-learning** - neatly built, available - and one sentence as a **safety net: "call the service desk if you have questions."**

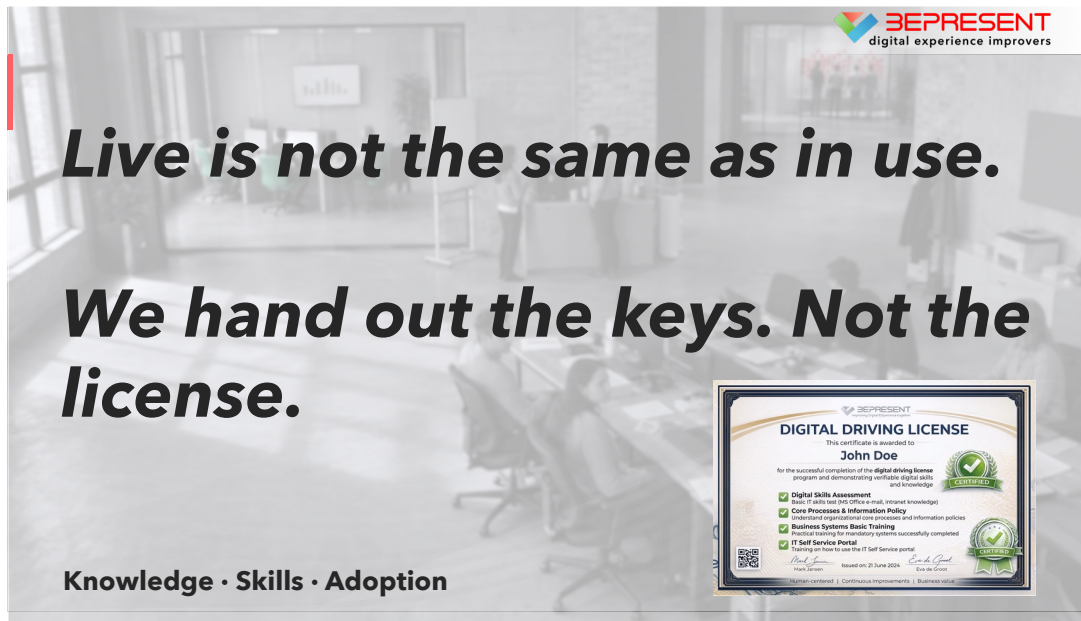
The safety net became the first line. Key users could have caught it, but there were hardly any, or they were busy with their own work. So what came in were help requests: people who didn't know how something worked because nobody had ever shown them. Tickets that formally existed but were really about something that should have happened earlier.

And the desk agents themselves had hardly any picture of how IT was used on the wards. They knew what the system did; they didn't know what a nurse did with it in the middle of a shift. **So they couldn't judge impact. What sounded like a small question at the desk was sometimes a big problem on the ward**, and vice versa. **Frustration on both sides, every first week of the month.**

Two things came together there. Nobody owned the intake: HR delivered people, IT delivered accounts, the e-learning was ready - every part was arranged, and the result belonged to no one. And there was no connection between the desk and the workplace: the people answering the questions had never seen where the questions came from.

Example 2. I recently spoke with the IT manager of a large organisation - the Build side - who said he didn't really care about users' digital skills. From his role, that is perfectly coherent: he delivers IT solutions. Skills are not in his assignment, his budget or his reporting. They are, however, the daily reality of the service organisation next door. Same organisation, different mandate.

So these are not technical problems. They are an organisational problem showing up as ticket volume. And as long as you treat it as ticket volume, you solve it again every month.



Live is not the same as in use.

We hand out the keys. Not the license.

Knowledge • Skills • Adoption

DIGITAL DRIVING LICENSE
This certificate is awarded to
John Doe
for the successful completion of the digital driving license program and demonstrating effective digital skills and knowledge

- ☒ **Digital Skills Assessment**
Basic IT skills (incl. Office e-mail, internet knowledge)
- ☒ **Core Processes & Information Policy**
Understand organizational core processes and information policies
- ☒ **Business Systems Basic Training**
Practical training for mandatory systems successfully completed
- ☒ **IT Self-Service Portal**
Training on how to use the IT Self-Service portal

Issued on: 21 June 2024

Number: 12345678 | Continuous Improvements | Business Unit

Live is not the same as in use.

Besides friction there is a second experience factor that starts upstream and costs organisations dearly. The hospital example I just gave you is about exactly this: **adoption - or rather, the lack of it.**

A system goes live. There's a date, there's an announcement, some training, the project is closed, the team is thanked. The reporting says "project done" because the project is judged on the date, not on the use.

! Once the date is met, nobody looks at the use anymore.

But **live is a technical state**. - **'In use' is a human one**. And the gap between the two is where your tickets will come from for the next three years.

! Measuring: Look at how we measure the human side.

We count trained users: participants, completed e-learning, coverage percentage. That's our proof that the people part has been handled.

! Trained is an output measure. Skilled is an outcome measure.

The number of people who sat through a session is not the number of people who can do their work afterwards without hesitation. We all know that - but we report the first number, because that's the number we can produce.

? Why is this the service desk's concern? You didn't run the project. You didn't design the training. Nobody asked you anything.

! Because a system that goes live - but is never really taken into use - keeps producing tickets.

Small tickets, for years. Repeating tickets. Exactly the tickets your team can answer in its sleep, and which are therefore never escalated.

! Adoption should be a project concern until use is proven. If the project stops at go-live, adoption becomes a service desk concern. For years.

Adoption is an HR concern too: people who never really adopt a new system work more slowly and with more frustration, and that affects engagement and how they experience their work. Adoption is not an IT project. It is a change people have to go through.

So friction and adoption are two sides of the same problem. Both start upstream. Neither is measured. Both end up with you.



***Start measuring what doesn't
get reported.***

***Manage whether users reach
their outcomes.***

ITIL v5 • Experience Management • XLA

ITIL v5 has a name for this: Experience Management. A name, though, is not a practice.

Experience is no longer a by-product of good operations but a management domain in its own right. At its core it is three things:

1. measure what users experience,
2. agree on it, and
3. improve on it.

! So "the service works" shifts from "the service is available" to "the user reaches their outcome without unnecessary effort".

That is a different question from the one an SLA asks. **An SLA measures the service: available, response time, resolution time.** It's an **agreement about outputs, met or missed regardless of what the user got done.**

And you can't manage experience on tickets alone: you'd miss the friction.

Tickets are data the user chose to share, and most friction never produces one. Those who give up don't call. Those with a workaround don't call. Those who've resigned themselves stopped calling long ago. So measuring what doesn't get reported is the first step.

One of the best instruments for this is the XLA: an Experience Level Agreement. It measures whether **users get their work done**, and **at what effort**.

Who of you already works with XLAs?

An XLA doesn't replace the SLA; it sits on top and tells you whether meeting the SLA delivered the agreed outcomes, for example,

- "eight out of ten new colleagues can do their core tasks unaided within two weeks", or
- "after a change, more than 90% of users say their daily app became easier to use".

XLAs demand their own process, tooling and, above all, strong management commitment. It takes serious time before you master it.

! A simpler alternative you can start tomorrow: short, recurring measurements per domain: the digital workplace, the business applications, the intranet, the desk itself, the IT projects in progress. **No long survey. A pulse, every quarter, three questions:**

- where does this work cost you unnecessary effort,
- how much time per week,
- and who is affected.

The result is one score per domain: positive impact minus negative impact - the share of users who say the work runs smoothly, minus the share who say it costs effort. Digital friction is that negative impact; below zero, it dominates. With it comes a top three of what hurts, in the users' own words: the service manager's work list.

For the first time you have a complete picture of your demand - reported and unreported - and a baseline. Without those, every intervention is a guess and every improvement unprovable. **Ticket numbers drop for ten reasons, most not yours; a measurement before and after is the only proof it was you. (PDCA)**

That's why measuring is the first step, not the last.

Everything that follows - the leverage, the process, the business case - stands or falls with this picture.

ROX case

Loss (year)

1.500 full time empl. / 45 weeks per year

1,5 hours loss per empl. per week

+ 100.000 hrs

Average rate €45,- per hour

+ € 4.500.000,-

What about the loss of business outcomes?

ROX: Return On Experience

So you have the data. Now bring it into the process you already own: Problem Management.

Problem Management exists to remove structural causes instead of symptoms. Today you feed it one thing - incidents - and so the root cause analysis goes **where incidents point: into the technology**. Logs, availability, configuration. Result: **everything within the norm**. End of analysis. Next month, the same tickets.

! Nowhere in ITIL does it say a cause has to be technical. So, you add the experience data, and the reflex changes.

The RCA no longer starts with "what broke?" but with "where does the work cost effort, and why?"

Sometimes the answer is technical. Often it isn't:

- a screen nobody understands without an explanation,
- a training that came too early,
- a workflow nobody asked the users about.

Those are valid root causes. They were just never measurable, until now.

It also changes the trigger. Today: no incident, no problem.

With experience data, a workaround a whole team uses or a score below zero in one domain is enough to open a problem, no phone call required. Problem Management becomes what it always could have been: proactive. A proposal, not a standard, but with Version 5 being worked out, this is the moment to ask.

And you do have a business case.

- When demand grows, more capacity works. For this year only. But **the cause stays**, so next year you need it again.
- **Removing the cause costs effort once, and the demand it removes doesn't come back.**

And the business case is bigger than the service desk. The desk pays for friction in tickets. The business pays for it in lost working hours, for every user, every week.

The cost of friction is time lost per employee, times employees, times hourly cost. Return on Experience is what you get back when you remove it. See the example on the slide. And that's before the lost business outcomes: delays, rework, fixes, and the service desk costs on top.

Next to that, the desk isn't the cost; the friction is.

Because you measured before, you can measure after. That turns an upstream fix from a belief into a proven saving, and proof is what gets you budget for the next one.

Why this works: no new method, no tool, no reorganisation. You feed an existing process with better input and widen what you accept as a cause. It's a PDCA loop: measure, find the cause, intervene, measure again. So you know whether it worked. Ticket numbers alone never show what was prevented.

What it gets you: less repetition, proof that funds the next intervention, and time for the new systems that keep coming and that you can't get to today.

And the data? Experience measurement tools exist for exactly this, but you can also start using the existing survey app in your ITSM-tool. Just know how to approach your users so they respond reliably.



***Put Experience in the
Definition of Done.***

***Don't ask for a seat at the table.
Take it.***

NEN8038 / ISO20000-18

First, one thing to do, not to think about: At your next project meeting, ask which user stories about the service desk are on the backlog. Provider? Ask your client.

If the answer is "none", you've just found your first piece of friction. No measurement needed.

The point of this message: you have data and a process, but if you're not at the table when the next system is designed, it starts again in a year.

And don't ask for a seat. Take it. Asking whether you may join makes your participation a favour someone can refuse. Instead, **state what has to be true before this system can go live.** The first conversation is about whether you belong there; the second is about whether the project is done.

What you put on the table: THE SERVICE DESK AS USER STORIES ON THE BACKLOG, NOT AS A STAKEHOLDER TO BE INFORMED.

What must be true from a support perspective before go-live: written, prioritised, estimated.

Make those stories go-live criteria:

- **you don't want to take over a system you already know will generate tickets. And bring the evidence.**
- **Without data you're someone with an opinion about usability;**
- **With data you're the only one in the room who knows what actually happens to users.**

You won't do this alone.

- **Project leads** and product owners turn your requirements into real backlog items.
- **User representatives** per persona, a cross-section of the people doing the work, not the enthusiasts.
- **Suppliers:** their support desk gains from less friction too, and regulation is coming - NEN 8038 for tendering and contracts, ISO/IEC 20000-18 in its final stage.
- **HR:** digital skills are a training question, and people who get stuck daily are less productive and more likely to leave. Communication: help texts written by people who can write.

No mandate? Probably true, and unnecessary. You're not asking for a veto, only that criteria exist and someone checks them. Nobody will argue there shouldn't be any. Start with one small project, show it works, and next time it's not a proposal but the way things are done.

This isn't about a bigger role for the desk. It's about less work that should never have existed.

In agile terms I ask one thing: put user experience in your Definition of Done.

***Fewer tickets is not the goal.
It's the proof.***



Thank you!

***To get your handout, go to
bepresent.nl/en/sits***

Conclusion

Less friction - and with it, a better experience - is the goal. Fewer tickets is the result: they show you fixed the cause upstream, instead of handling the consequences at the desk.

They are the proof that you now see what you didn't see before, and that you removed it where it started.

A better service desk starts upstream - and the desk is the only one that can start there.

Fewer tickets is not the goal. It's the proof.

About the speaker

Jan Massier started on the service desk at Apple Computer in 1995 and has spent the years since on the part of IT that rarely makes the project plan: the people. Through implementation management at an international IT consultancy, IT communication management and adoption management, he developed his own approach to change in IT environments. He is ITIL- and XLA-qualified and works with MoC4IT™ - Management of Change for IT - a practical method built on Kotter, ADKAR® and ITIL. His work centres on three connected questions: why systems that work technically still go unused, where the friction sits that never gets reported, and how to make it measurable so something can be done about it.

About BePresent

Founded in 2006, BePresent works as Digital Experience Improvers for organisations in healthcare, financial services, industry and the public sector: strengthening and accelerating IT adoption, measuring and removing digital friction, and making IT communication understandable for the people who have to work with it. Two instruments support that work. The Adoptiekompas™ uses short pulse measurements to show where adoption stalls - on must, want and can. The DEX-kompas™ measures digital friction in the workplace, across the domains where the work actually happens.

The thread is always the same: output is not outcome. A system that is live is not yet in use. A user who is trained cannot yet do it. And what you don't measure, you can't improve.

Contact

Jan Massier · BePresent - Digital Experience Improvers · bepresent.nl · jan.massier@bepresent.nl