From IT Manager to Entrepreneur

Real service by real people

Ruben van der Graaf
ServiceChanger

From IT Manager to Entrepreneur

Real service by real people

My journey from support, teams and Shift Left
to ServiceChanger and the Service Automation Hub

Ruben van der Graaf
June 2026

Contents

Foreword

This book is not a technical manual. It is not a marketing brochure either. It is my personal story about a journey that is still in motion.

That journey starts in support. From there it moves through application management, IT operations, teams, leadership, Shift Left, automation, AI and eventually entrepreneurship.

I am not a writer. I am not a typical book reader either. Still, I wanted to capture this story. Not because everything is finished, but because I am right in the middle of the movement. Some ideas have been with me for years. Others came from mistakes, people I met, projects that worked and projects that did not.

If there is one thread running through this story, it is this: technology only becomes truly valuable when it helps people do better work.

For me, service management is not ultimately about tickets, tools or dashboards. It is about people who need help and people who want to help.

Prologue: Above the Clouds

Sometimes a journey literally starts above the clouds.

In 2019 I was on a plane from Orlando back to the Netherlands. I had just visited Service Management World. My head was full of presentations, conversations, ideas and especially the feeling that service management could be done differently.

During that flight I took out a piece of paper and started sketching. Not a fully developed product. Not a business plan. Not an investor deck. Just a first portal concept, drawn from an airplane seat, above the clouds.

Looking back, that sketch was more than a drawing. It was the beginning of an idea that would years later become ServiceChanger. The vision was still simple then: free IT service desks from repetitive administrative work. Less manual work. More value. More impact.

Six years later, that idea has become much bigger. It is not only a platform. It is a way of looking at service, people, automation and AI. It has become an ecosystem where strategy, software, capacity and visibility come together.

This book captures that journey. Not as a polished success story. Not as a story without mistakes. But as an honest record of learning, building, doubting, starting again and continuing to believe that service can be better.

Part 1: Where It Started

Chapter 1: The Person Behind the Technology

IT became my hobby. People became my work.

If you only look at the outside, it may seem as if my story starts with technology. Entra ID, Active Directory, automation, AI, ITSM, agents, monitoring and dashboards. But the origin is somewhere else. The origin is people.

I was involved with computers from an early age. Building websites for people nearby, solving computer problems, trying systems and getting things to work. But technology was never the whole story. There was always a human side to it.

I wanted to do something with people. First through sports, later through work, eventually through IT. Even during the years I worked in a nightclub, those two worlds came together. On one side there was lighting, sound, computers and systems. On the other side there were guests, atmosphere, energy and people who needed to have a good evening.

Looking back, that was already an early version of the combination that would later become central: people and technology.

During my studies I noticed that I was not the best programmer in the room. I did not have to be. In project groups there were often people who were much stronger technically. They wanted to write code and build systems. I enjoyed bringing the whole thing together. Presenting, organizing, giving direction and making ideas understandable.

That pattern never disappeared. Later I grew from support engineer to application manager, from application manager to team manager and eventually to IT manager. In that role I

became responsible for international IT operations, multiple teams and thousands of end users.

That is where I learned something that comes back throughout this book: technology is rarely the only problem. More often the real challenge is in people, processes, collaboration and communication.

One question became more and more important:

Why do we actually do it this way?

That question seems small, but it is fundamental. It forces you to step back from daily pressure. It forces you to look not only at tickets, processes and tasks, but at the system behind them.

Why does this ticket keep coming back? Why is someone doing this manually? Why does a user have to figure this out alone? Why does nobody know where the process gets stuck? Why are smart people spending every day on the same administrative work?

That question became the foundation under everything I later started building.

Chapter 2: From Support to IT Manager

I became an IT manager relatively young. Not because I was the best technician. I was not. There were always people around me who went deeper technically, saw more details and were more specialized.

My strength was somewhere else. Giving direction. Seeing the bigger picture. Connecting business and IT. Getting people moving. Creating room to build.

In a management role you do not only get processes and systems. You get people. Their ambitions. Their insecurities. Their home situations. Their growth. Their frustrations. Their potential.

That makes leadership beautiful, but also heavy. Especially during reorganizations, role changes or moments when people can no longer keep up with technology. At that point IT management is not a technical profession. It is people work.

One important lesson for me was that a manager sometimes needs to be a firewall. Not between people and the truth, but between the team and unnecessary noise. Politics, emotion, business pressure, escalations and unrest can paralyze a team. A manager needs to absorb, translate and filter that pressure.

The goal is not to protect people from everything. The goal is to give them room to do good work.

There is another lesson in that. I am direct. Fast. Red, in the language of color tests. In incidents that is useful. Then someone needs to take control. But in change, directness can also create friction. Not everyone moves at the same pace. Not everyone thinks in the same way.

That is why the chameleon became important. Not losing yourself, but learning to switch. Sometimes you need to push. Sometimes you need to slow down. Sometimes you need to explain. Sometimes you need to listen.

One of my biggest mistakes as a manager was being too far ahead of people. The vision was clear. The direction was right. But others were not there yet. And change does not work if you are right but nobody follows.

The lesson: one meeting too many is better than one meeting too few. Especially with major changes such as automation and AI. People need to understand why something is happening before they take ownership of it.

Part 2: The Lessons

Chapter 3: Shift Left

Moving work is better than stacking work. But eliminating work is even better.

Shift Left did not start for me as theory. It started as a practical answer to pressure.

Developers were busy with operational work. Application managers were stuck between management tasks and daily questions. Service desk employees had many recurring tasks, but also wanted to grow. End users wanted to be helped faster. And the organization wanted to deliver more without constantly adding more people.

The question became: can we distribute the work more intelligently?

The idea was to move tasks to the place where they could be done most effectively. Work that sat with development could sometimes move to application management. Work from application management could sometimes move to the service desk. Work from the service desk could sometimes move to self-service.

Development to application management. Application management to service desk. Service desk to self-service.

On paper that sounds simple. In practice it requires trust, training, good processes and clear boundaries. People need to understand why it is happening. It must not feel like work is being taken away. It must feel like growth is being made possible.

For developers it meant less operational noise and more time for projects. For support teams it meant more challenging work and more growth perspective. For end users it meant they could

arrange certain things themselves, at the moment that suited them.

The first success moment was surprisingly simple. Service desk employees came to ask whether there was still work for them. It was getting quieter. Tasks that had returned all day before had been moved or automated. Users could do more themselves. Tickets were routed better.

That was the moment I thought: this really works.

But the real power of Shift Left is balance. Not only moving work down, but letting the whole chain move with it. If you only empty the service desk into self-service, you solve only part of the problem.

Shift Left is not meant to push work to the cheapest place. It is meant to make work land in the right place.

Chapter 4: The Self-Service Trap

Not every form of self-service is service.

Shift Left worked. Maybe too well.

What started inside IT as a way to work smarter also became interesting for other departments. HR, Facility, Finance, Data and Product Content saw that self-service could help standardize and route work.

For each department individually, that made sense. For HR it was efficient. For Facility it was efficient. For Finance it was efficient. For IT it was efficient. But nobody looked closely enough at the total picture for the end user.

The user got more forms. More portal choices. More processes. More tasks that used to be handled by departments.

That is how the self-service trap appeared.

One sentence stayed with me:

This new portal just means I have to do IT's job myself.

That sentence hits the core. Self-service can help users. But self-service can also become a way to move internal work outward. Then a department looks more efficient, but the organization as a whole does not become better. The work does not disappear. It only moves.

That is not transformation. That is displacement.

Good self-service helps the user. The task is recognizable. The process is simple. The user gets immediate value back. The request becomes faster, clearer or better handled than through an older channel.

Bad self-service feels different. The user has to search. The task rarely occurs. The questions are written from an internal perspective. The user feels that he is taking over work from a department.

That is why self-service is not a goal. It is a means.

The conclusion became sharper: do not endlessly shift work left. Eliminate unnecessary work. Automate repetition. Keep human attention where it matters.

Chapter 5: Service First

The service desk is the face of IT.

Not the architecture team. Not the roadmap. Not the cloud platform. Not the development team. For most employees, IT is the service desk. That is where they ask for help, where they get stuck, where they become frustrated or where they are helped well.

That is why the service desk is much more than a ticket factory.

A service desk builds trust or breaks trust.

I like to use the metaphor of an emotional bank account. Every positive interaction is a deposit. An employee who is helped kindly. A new colleague who receives a working laptop on the first day. A user who quickly gets clarity. An incident that is handled with empathy.

Every poor experience is a withdrawal. A ticket that sits too long. An employee who is sent from one person to another. A blunt answer. A bad handover. A problem that has been going on for hours without communication.

If the bank account is positive, IT has credit. People are more likely to accept that something goes wrong once in a while. If the bank account is negative, every problem becomes bigger.

Service First means taking that bank account seriously.

A good example is onboarding. The first working day of a new employee is an opportunity to build trust. The account needs to be ready. The laptop needs to be ready. Access needs to be right. Software needs to be available. And someone from IT needs to take time to walk through the basics.

Twenty minutes of personal attention can create months of positive experience.

That is service.

That is also why the ideal service desk employee is not necessarily the best technician. One of the strongest examples is the story of the postman who became successful on the service desk. He did not have a classic IT background, but he had work ethic, friendliness, communication, interest in IT and willingness to learn.

Technology can be trained. Service mindset is harder to teach.

Chapter 6: Automation Starts With Understanding

Automation does not start with tooling. Automation starts with understanding.

Many organizations start automation in the wrong way. They see a task that is irritating and immediately think: we need to automate this.

Sometimes that is true. Often it is not.

The first question should not be whether something can be automated. The first question should be why the work exists.

What is happening here? How often does it happen? Who does it? Which systems are involved? What is the impact? Is this an exception or a pattern? Can it be solved structurally? Is automation the best answer or are we hiding a process problem?

Only after that can you decide whether automation makes sense.

A task that happens two or three times per year usually does not deserve automation. A task that happens twenty times a day is interesting. Not because automation is trendy, but because there is volume. There is time. There is frustration. There is opportunity.

That is why good automation starts with data. Ticket data, process data, queues, repetition, lead times, categories. Where is most of the work? Where is the manual effort? Where do errors occur? Which tasks are predictable?

The biggest mistake in automation is that organizations automate isolated steps without understanding the chain.

Take onboarding. Some organizations automate account creation and think onboarding is finished. But onboarding is much

broader. There is an HR trigger. There is an account. There is access. There are licenses. There is hardware. There is communication with the manager. There may be a service desk ticket. There is data retention during offboarding. There are compliance agreements.

If you only automate the account, you automate a piece. Not the process.

Real automation looks at the full flow.

The biggest misconception about automation is that it replaces people. In my view, automation strengthens people. It removes repetitive work so people can focus on tasks with more value.

Part 3: Building in Practice

Chapter 7: From Rules to Buckets

Why make it difficult when it can be simple?

ServiceChanger started with access automation. That was logical. Access, licenses and assets create a lot of service desk work. Access management in particular is time-consuming, error-prone and sensitive to pollution.

A common practice is copying rights from an existing user to a new user. Someone searches for a colleague with the same role and copies the groups. But what if that colleague has had multiple roles over the years? What if old rights were never removed? Then the new employee immediately receives too many rights.

That is not only inefficient. It is also a security and compliance problem.

The first version of ServiceChanger was rule-based. If someone has this job title, give this group. If someone has this location, give this group. It worked, but it quickly became complex.

The second version combined rules. That seemed smarter, but again it created complexity. Rules on rules become hard to understand and hard to maintain.

The third version brought the breakthrough: buckets.

A bucket is a set of rights linked to an attribute or combination of attributes. Think of location, department, cost center, job title or a combination. Amsterdam plus IT Manager can get a different bucket than Hamburg plus IT Manager.

The idea is simple. Users have attributes. Buckets contain rights. If the user has the attribute, the user gets the right access. If the attribute changes, access changes with it.

The power is in simplicity. The model is understandable. It can be managed. It works with Microsoft Entra ID and Active Directory without sitting in between systems.

There are also safety mechanisms. The Attribute Change Queue prevents unexpected HR changes from immediately causing large access changes. The Data Cleaner helps keep attributes clean. Temporary Access makes short-term access possible without polluting the model.

ServiceChanger is not a classic heavy IAM project. It is service desk automation for the operational reality of Microsoft environments.

Chapter 8: Production Is the Truth

A test tenant is not a production tenant.

One of my most important entrepreneurial lessons did not come from a success, but from a crisis.

ServiceChanger was live at a customer. I was on-site. It was around four in the afternoon. There was a pizza evening. The atmosphere was good. Then people started leaving the room. Calls came in. Users reported that access was disappearing.

The energy in a room changes immediately when that happens.

What is going wrong? Where is the problem? What is the impact? What needs to stop immediately?

Together with the customer, the Microsoft team and my own development team, we investigated what was happening. The first step was not to fix immediately. The first step was to stop. The sync had to be disconnected so the problem would not grow.

After that the real work began. Understand where the fault was. Restore the structure. Correct the data. Only then bring the sync safely live again.

The evening became night. The night became morning. Around half past eight it looked solved, but then it started going wrong again. Back into the cycle. This time it became clear faster where the issue was and later that day it was structurally solved.

It was heavy. Screen tired. Emotional. Not because it was just another incident, but because it was my own product at a real customer. As a supplier you cannot say: we will look again tomorrow. You continue until it is safe.

The lesson was clear: production is the truth.

You can test. You must test. But a production tenant with real data, real users and real dependencies is different from a test environment with dummy data.

This crisis became an important lesson for ServiceChanger. Not only technically, but mentally. Build stop mechanisms. Take production seriously. Be honest. Solve. Learn. Improve.

Chapter 9: Hyperautomation and Agents

A chatbot is not the beginning. A chatbot is the endpoint.

Hyperautomation is the combination of automation and AI. Traditional automation follows rules. AI adds understanding. That means processes can be automated that previously required too much context.

In service management that matters. Tickets are messy. Users do not always write clearly what is going on. Screenshots sometimes contain the most important information. A solution depends on policy, location, role, application and earlier incidents.

That is why ITSM Autopilot did not start as a chatbot. A chatbot has a high maturity threshold. It talks directly to users. It must be reliable. It must know what it does not know. Without knowledge, data and tools it will guess or give generic answers.

The better route is agents.

Agents can start in Shadow Mode. They watch, analyze tickets, write private notes for operators, suggest additional questions, search knowledge and build trust. The end user sees nothing yet. The organization can calmly assess whether the AI is good enough.

That is why Shadow Mode is not just a technical feature. It is a trust mechanism.

Another principle is Knowledge Before Intelligence. Without knowledge, AI is worth little. Today that knowledge often sits in tickets, old documents and especially in people's heads. ITSM Autopilot tries to capture that knowledge and make it reusable.

Every ticket contains value: problem, context, communication, solution and frequency. If you structure that well, the knowledge base improves. If the knowledge base improves, agents improve. If agents improve, tickets are handled faster and better.

That is the Knowledge Flywheel.

Ticket to solution. Solution to knowledge. Knowledge to better AI. Better AI to better service.

The goal is not that people disappear. The goal is that administrative work disappears. People remain needed for exceptions, empathy, physical work, coaching, service improvement and control.

Chapter 10: Visibility for Everyone

Everyone understands a traffic light.

ITSM Radar comes from a simple practical experience. Monitoring started with a few tiles. Is the webshop online? Is production running? Is an important process green?

Three tiles became six. Six became twenty. Eventually, dozens of production pages existed with monitoring for different processes.

It worked because it was understandable.

Many monitoring tools are technical. They show logs, error codes, alerts, metrics and dashboards that are mainly understandable for engineers. That is valuable, but not enough. A service desk employee, manager or business stakeholder often wants to know something else: is the process still working? Where does it get stuck? Are users affected?

That is why ITSM Radar is built around the traffic light model.

Green is good. Amber is attention. Red is action. Grey is unknown or no data.

Everyone understands that. Not only developers. Not only infrastructure engineers. Also service desk, management and business.

It is not about replacing Azure Monitor, SCOM, AWS, GCP, databases or API logs. It is about a layer above them. A layer that translates technical signals into understandable operational insight.

That can be infrastructure. But it can also be an order flow. Customer portal to Order API, to ERP, to Finance, to WMS, to Transport. If a queue fills up somewhere or an API slows down,

you do not want to discover that three hours later.

The right signal at the right moment for the right person. That is the value of ITSM Radar.

Visibility is not a luxury. You cannot improve what you cannot see.

Chapter 11: The Service Automation Hub

Direction. Automation. AI. Capacity. Visibility.

At first glance, the parts of ServiceChanger Group may seem separate.

ServiceManagementPartner is advisory. ServiceChanger is software. ITSM Autopilot is AI. RemoteServicedesk is capacity. ITSM Radar is monitoring.

But together they solve five fundamental problems.

Many organizations do not know clearly where they need to go. That is where ServiceManagementPartner fits.

Many organizations have too much manual work. That is where ServiceChanger fits.

Many organizations have too many tickets and too little knowledge. That is where ITSM Autopilot fits.

Many organizations have too little capacity. That is where RemoteServicedesk fits.

Many organizations have too little visibility. That is where ITSM Radar fits.

Together they form the Service Automation Hub.

The hub is not a random collection of products. It is an operating model for modern service organizations. It combines strategy, automation, AI, capacity and visibility.

That matters because no single part is enough. Strategy without execution remains a plan. Automation without vision can make the wrong things faster. AI without knowledge becomes noise. Capacity without improvement remains expensive. Monitoring

without understandability remains technical noise.

The power is in the combination.

A customer can start with RemoteServicedesk because there are temporarily too few people. During that support it becomes clear that many tickets are about access and licenses. Then ServiceChanger can help.

A customer can start with ServiceChanger and discover that many tickets still come in that could be triaged, enriched or answered better. Then ITSM Autopilot fits.

A customer can start with ITSM Radar because processes are unclear. Monitoring shows where work gets stuck. Automation can follow.

A customer can start with ServiceManagementPartner because direction is needed first.

The Service Automation Hub is therefore not a product line. It is a way of looking at service management.

Part 4: People and the Future

Chapter 12: The People in the Story

A strong team is not made of the same people.

If there is one thread running through all my stories, it is that people make the difference.

Not only customers. Not only users. Also engineers, operators, managers, teachers, students, suppliers, partners and colleagues.

I especially admire people who are strong where I am less strong. People who see details. People who bring calm. People who finish things properly. People who manage the people side better. That is not a threat. That is the strength of a team.

A strong team is not made of copies. Vision needs detail. Speed needs structure. Directness needs empathy. Builders need finishers.

One of the best lessons came from an employee who had applied several times but had never been hired. When he finally got the opportunity, he grew into service desk manager and later team manager. He turned out to be exceptionally strong on the people side.

That taught me something important: sometimes an organization only sees talent when someone is trusted enough to grow.

Partners also play a major role. Good partnerships go beyond contracts. They become personal and trusted. They make possible what you cannot achieve alone.

The same is true for education and students. Working with young people shows how much energy is created when students are taken seriously and given real practical questions. Students

motivated, team motivated.

This human side fits the core of ServiceChanger. Technology is important, but it is never the whole story. The real value appears when technology helps people grow, collaborate and deliver better service.

Chapter 13: The Future of Work and Service

The future is not human or machine. The future is human and machine.

The future of service management will not be determined by one tool. Not by ChatGPT, not by Copilot, not by a new ticketing system and not by a new framework.

The future will be determined by a new operating model.

Today, people do a lot of work manually. They read tickets, collect information, route requests, search knowledge, write answers, create reports, check dashboards and execute standard actions.

In the coming years that will shift.

AI agents will take over more of the repetitive layer. Not as one big magical system, but as specialized agents that do small parts of work well. Triage, knowledge, clarification, problem detection, sentiment, reporting, monitoring, API actions.

The human does not disappear. The role changes.

The human becomes an orchestrator.

The orchestrator understands processes, data, service and technology. He does not execute everything himself, but directs systems, checks quality, improves workflows, intervenes in exceptions and makes sure service stays human.

In 2030 the service desk will be hybrid. AI and people side by side. Much administrative work will happen automatically. People will focus on exceptions, empathy, guidance and service.

IT Operations will move toward 80 percent automation and 20 percent human control, orchestration and service. That is not an

exact prediction, but a direction of thought.

The biggest mistake organizations can make is waiting. Not because every AI tool is mature today, but because learning takes time. Structuring data takes time. Building knowledge takes time. Bringing people along takes time. Building trust takes time.

The organizations that win start now. Not by putting everything live, but by experimenting, organizing data, structuring knowledge, understanding processes and bringing people along.

The speed of change is underestimated. That is why preparation matters more than prediction.

Chapter 14: The Manifesto

Customer Happiness plus Employee Happiness.

In the end, service management is not about tickets.

Not about ITIL. Not about tooling. Not about dashboards. Not about AI.

It is about people who need help and people who want to help.

The user wants clarity, speed, empathy and a solution.

The support engineer wants meaningful work, growth, better tasks and less repetitive administration.

The future of service management must improve both.

Customer Happiness plus Employee Happiness. That is the core.

That is why technology must take over the repetitive layer. Not so people disappear, but so people can grow.

That is why self-service must help users and not burden them.

That is why automation must start with understanding.

That is why AI must start with knowledge.

That is why trust must come before autonomy.

That is why monitoring must become understandable.

That is why capacity must be flexible.

That is why strategy must be connected to execution.

This is what the Service Automation Hub stands for.

Real service by real people. Administrative work by machines.

If ServiceChanger may be known for something in 25 years, I hope it is not only software. I hope it is the change it brought in

how organizations look at service desks.

From cost center to value point. From ticket factory to service organization. From reactive to proactive. From manual to automated. From unclear to visible. From pressure to space.

That is the future of service management. And that is why I am building ServiceChanger.

Afterword: Where I Am Now

This is not the end point. This is the beginning of the next phase.

This book describes a journey that is still ongoing.

From support engineer to IT manager. From Shift Left to hyperautomation. From a sketch in an airplane to an ecosystem of products and services. From manager to entrepreneur.

A lot has already been built. ServiceChanger is running. ITSM Autopilot is taking shape. RemoteServicedesk adds capacity. ITSM Radar makes visibility understandable. ServiceManagementPartner brings direction and experience.

But the real ambition is bigger than any individual product.

The ambition is to change the way organizations look at service desks. Not as cost centers. Not as ticket factories. But as the face of IT and as the foundation for better service.

This book is therefore not an end point. It is a snapshot. A photo of a vision in motion.

And maybe that journey literally started above the clouds. But it has not landed yet.