



# Scaling at Private Equity Speed



THE IT LEADER'S HIDDEN PLAYBOOK



# The Pattern Nobody Talks About

- **Month 6:** The new CTO presents their integration roadmap. Slick deck. Clear milestones. Board loves it.
- **Month 18:** Three months behind schedule. £4M over budget. The acquired company still can't close their books.
- **Month 24:** Leadership starts looking for "fresh perspective."

If you've worked in PE-backed businesses, you've seen this before. Different companies, different industries, same ending.

Here's what nobody tells you: The IT leader usually does everything "right" - and still struggles.

They rationalise systems. They consolidate platforms. They deliver the projects on the roadmap. But somehow, they still become the bottleneck when deal three arrives faster than planned.

This isn't about working harder or moving faster. You're already doing both.

It's about the invisible traps that competent IT leaders walk into - and the uncomfortable decisions that separate success from struggle.



# What You'll Learn

## CHAPTER 1: WHY GOOD IT LEADERS STILL STRUGGLE IN PE

- The invisible trap: Why the value creation plan you received is already obsolete
- The three ways IT actually creates value (everything else is noise)
- The test that reveals if you're ready for the next deal

## CHAPTER 2: THE £10M DECISION NOBODY WANTS TO MAKE

- Why "let's keep both systems for now" costs more than standardising
- The 90-day window (and why it closes faster than you think)
- How middleware becomes a permanent tax

## CHAPTER 3: WHY YOUR "SINGLE SOURCE OF TRUTH" IS NEITHER

- The three failure patterns that kill data warehouse projects
- The one metric that reveals if your data is catastrophically broken
- The incremental approach that builds enterprise platforms without trying  
(AND HOW TO FIX IT)

## CHAPTER 4: WHY YOUR IT TEAM IS THE BOTTLENECK ↩

- Why traditional IT operating models kill PE speed
- The five shifts that make teams move faster
- How Azure landing zones enable repeatable integrations
- The transformation roadmap that actually works  
(EVEN WHEN TECHNOLOGY WORKS)

## CHAPTER 5: WHY CHANGE PROGRAMMES FAIL ↩

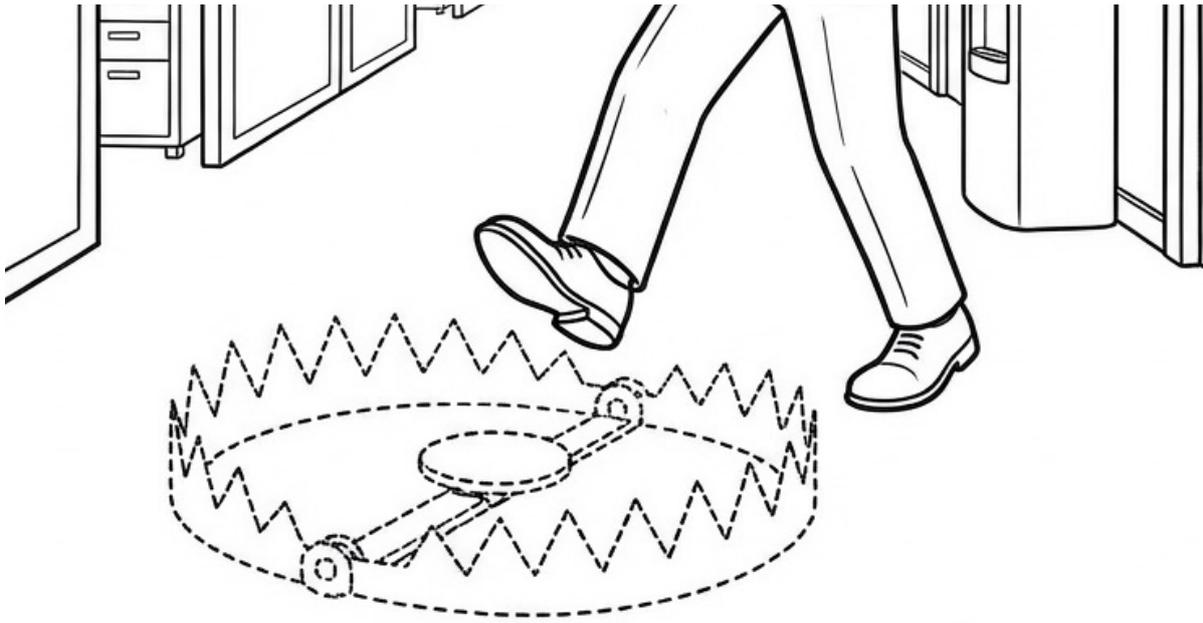
- The adoption crisis nobody budgets for
- The local hero strategy that actually scales
- How to make people want to switch systems

## CHAPTER 6: THE ONLY KPIs THAT MATTER

- Why investors don't care about your SLAs
- The three numbers that predict exit valuation
- The scorecard that keeps you funded

# Chapter 1: Why Good IT Leaders Still Struggle in PE

The problem isn't competence. It's that most IT leaders are solving yesterday's problem while PE investors are already pricing in tomorrow's deal.



### THE INVISIBLE TRAP

Here's what nobody tells you: the value creation plan you receive on Day 1 is already obsolete. Not wrong - obsolete.

It was written assuming static conditions. But PE doesn't operate in static conditions.

The real brief isn't "deliver this three-year roadmap." It's "build a machine that can absorb three acquisitions, two carve-outs, and a geographic expansion - in any order, at any time - without breaking."

Yet most IT leaders still plan linearly. They treat the current estate as the final estate. They build bespoke integrations for "just this one acquisition." They convince themselves the next deal is 18 months away, so there's time to tidy up first.

There isn't.

### THE TUESDAY MORNING TEST

It's Tuesday morning. You're in the weekly ops review. The CEO's phone buzzes. She steps out. Returns five minutes later: "We're acquiring CompetitorCo. £120M deal. Closes in 60 days. They've got 400 staff, their own ERP, and a highly customised CRM. Integration timeline is aggressive. Questions?"

***If your first instinct is to start scoping -  
you're working with the wrong operating model.***

The IT leaders who succeed in PE don't start planning when the deal lands. They pull up a pre-built playbook, assign the standing integration squad, and start onboarding Monday morning.

No scoping phase. No discovery workshops. No "let's understand their environment first." Just execution.



## WHAT ACTUALLY DRIVES VALUE

You've seen the slides: "IT as a value driver." "Strategic enabler." "Digital transformation."

In PE, IT creates value exactly three ways:

### **1. You make the next deal possible faster than planned**

The business case assumed 6 months to integrate. You do it in 10 weeks. That's 16 weeks of synergies brought forward. That's millions in NPV.

This isn't about heroics. It's about having infrastructure that absorbs new entities like Lego bricks - fast, repeatable, no drama.

### **2. You find money the deal model missed**

The investment committee modelled £3M in SaaS savings. You find £7M because you looked across the full stack - duplicate Salesforce orgs, forgotten Azure tenancies, that ERP module nobody's used since 2019.

You've just improved the IRR.

### **3. You de-risk the exit**

Three years from now, when it's time to sell, the acquirer's tech due diligence needs to be boring. No legacy time bombs. No "we'll fix it next quarter" technical debt. Clean, standardised, documented.

That's what protects valuation. A messy tech estate can torpedo an exit or cut the multiple.

Everything else - the dashboards, the "innovation labs," the digital strategy deck - is theatre.



## WHY SMART PEOPLE MAKE THE SAME MISTAKES

### **Mistake 1: Optimising for the current estate instead of the next deal**

You inherit a mess. Your instinct is to tidy it up before taking on more complexity. Reasonable - and wrong.

While you're cleaning up, the next deal lands. Now you're integrating onto a foundation that's still half-built.

**The fix:** Accept that the current estate is just the first entity in a series. Design for scale from Day 1.

### **Mistake 2: Building bespoke integrations because "this one's different"**

Every acquisition is different. But your response can't be.

The moment you start customising integrations to fit each acquisition, you've lost. Now you're running bespoke infrastructure for every entity.

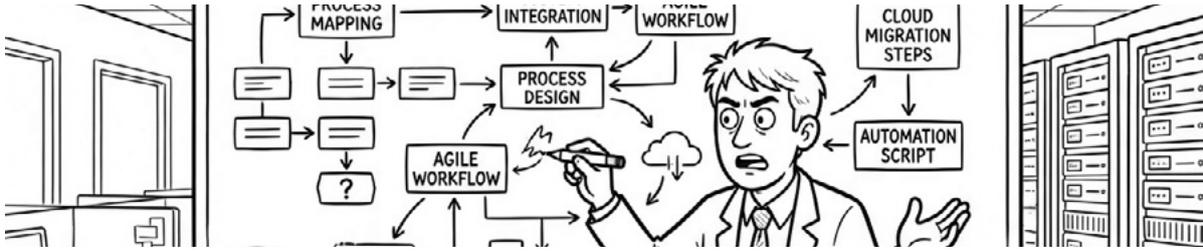
The fix: Define the target state architecture once. Make it non-negotiable. Every acquisition migrates to it.

### **Mistake 3: Treating integration as a project, not a product**

Projects end. Products evolve.

If you treat each integration as a unique project, you rebuild the wheel every time.

The fix: Build integration as a product. Standing team. Standardised tooling. Each deal makes the next one faster.



## WHAT “REPEATABLE” ACTUALLY MEANS

Most IT leaders say they have “repeatable processes.” What they mean is “we documented what we did last time.”

### Repeatable means:

- The 47-step checklist is now 12 automated workflows
- Your cloud landing zone provisions in 4 hours, not 4 weeks
- Finance can run consolidated reports on Day 1, not Month 3
- The integration team is the same four people every time
- You can quote an integration timeline before you’ve even seen their systems

Real repeatability looks boring. No heroics. Just the same machine, running faster with each iteration.

## THE ONE DECISION THAT CHANGES EVERYTHING

Stop treating each acquisition as a unique snowflake that deserves a bespoke integration approach.

Start treating your core platform as the non-negotiable destination that every acquisition migrates to.

### The best IT leaders in PE make this call in Month 1:

“Here’s the target architecture. Everything lands here. Non-negotiable. We’ll make the migration smooth, but we’re not preserving legacy systems to make people comfortable.”

When the target platform is fixed, integration becomes a production process. When it’s up for debate every time, you’re back in consulting mode.



## WHERE TO START (IF YOU'RE ALREADY BEHIND)

### **Instead, start building the machine for the next one:**

1. Define your target state architecture now - What does the end-state platform look like? Write it down. Make it policy.
2. Automate the top 5 repetitive tasks - User provisioning, data migration, security config. Build the automation now.
3. Document what slowed you down, then build the tooling that eliminates it - Did compliance reviews add weeks? Did data quality stall reporting? Fix the root cause.
4. Make decisions policy - "Can we keep our legacy CRM?" "Do we migrate all users at once?" These shouldn't be open questions.
5. Keep your integration team together - Don't disband when this deal closes.

The next deal announcement will come faster than you expect.

Don't try to fix the current integration. Firefight enough to get it over the line, but don't gold-plate it.

# Chapter 2: The £10M Decision Nobody Wants to Make

Three months in, the IT leader presents two options:

Option A: Migrate to the standard platform. 18 months, £8M, significant disruption, but you end up with one clean estate.

Option B: Build integration middleware. 6 months, £2M, minimal disruption, everyone keeps their systems.

Guess which gets picked?

Fast forward 24 months. Option B has now cost £12M. There are still two ERPs. Month-end takes three weeks. The next acquisition can't integrate because the middleware is now a tangled mess.

## WHY INTEGRATION MIDDLEWARE IS A TRAP

**It sounds reasonable:** “Rather than ripping and replacing, let’s connect the systems.”

The first integration works. Finance can consolidate reports. Board is happy. Then the second acquisition arrives. Now you’re integrating three systems. Complexity compounds. Two consultants become six. What was “temporary” is now “the architecture.”

By the third acquisition, you’re spending more on middleware than standardisation would have cost. And you’re slower than before.

**The trap:** Middleware preserves the illusion of progress while deferring the actual decision. It kicks the can down the road - at compounding cost.

## THE HIDDEN TAX



<b>Year 1:</b> £2M implementation + £300K licensing = £2.3M
<b>Year 2:</b> £500K enhancements + £350K licensing + 2 FTE = £1.2M
<b>Year 3:</b> £800K fixing broken integrations + £400K licensing + 3 FTE = £1.7M
<b>Year 4:</b> £1.2M extending for new acquisitions + £500K licensing + 4 FTE = £2.5M

Total four-year cost: £7.7M - and you still have two ERPs.

Option A would have cost £8M and been done in 18 months.

## THE REAL COST OF “LET’S KEEP BOTH FOR NOW”

Dual systems cost more than licensing:

### Indirect Costs:

- Training and onboarding on two systems
- Reporting complexity requiring manual reconciliation
- Decision latency when teams argue about “which system is the source of truth”
- Next acquisition now integrates into a mess, not a platform
- Your best people leave because this environment is exhausting

The first-year saving becomes a five-year tax.

## THE OPPORTUNITY COST

One portfolio company deferred ERP standardisation for three years to “avoid disruption.” At exit, the acquirer discounted the offer by £15M to account for integration costs.

The three-year deferral cost more than standardisation would have.

## WHEN “LATER” NEVER COMES

- **Year 1:** “We’ll standardise after the integration settles”
- **Year 2:** “We’ll do it before the next acquisition”
- **Year 3:** “We’ll include it in exit preparation”
- **Exit year:** “The new owner can deal with it”

Standardisation keeps getting deferred because it’s never the most urgent thing. Until it’s blocking the exit.

## THE COMPOUNDING COST OF DELAY

**Every quarter you defer, the cost increases:**

- Q1:** Clean migration possible. Users can guide the transition.
- Q2:** Key users have left. Undocumented customisations discovered.
- Q3:** New processes built on old systems. More dependencies.
- Q4:** Another acquisition lands. Now integrating three entities across two platforms.
- Year 2:** “Too complex to change now.”

The window for painless standardisation closes faster than people think.

## THE 90-DAY WINDOW

There’s a brief period after acquisition when painful changes are expected and accepted. People understand things will be different.

That window closes fast.

- Month 1-3:** “Of course things are changing, we just got acquired”
- Month 4-6:** “Things are settling down, let’s not rock the boat”
- Month 7+:** “This is how we do things now, why are we changing again?”
- By Month 6,** the acquired team has settled. Systems are familiar. “Temporary” workarounds are now “how we do things.”
- The rule:** Make the hard platform decisions in the first 90 days.



## WHAT GOOD LOOKS LIKE

The best IT leaders don't present options. They present the decision:

"We're migrating to Platform X. Here's the timeline. Here's the risk mitigation. Here's what stays and what goes. We start Monday."

The decision on whether to standardise is non-negotiable. It's how to execute it safely.

### What IS negotiable:

- Timing the cutover to avoid your busiest periods
- Training schedules that fit your team
- Data migration support and validation

## THE QUESTION THAT CUTS THROUGH POLITICS

When the business pushes back:

***"If we were designing this company from scratch today, would we choose to run two ERPs / CRMs / HR systems?"***

The answer is always no.

So why are you defending the accident of history that left you with two?

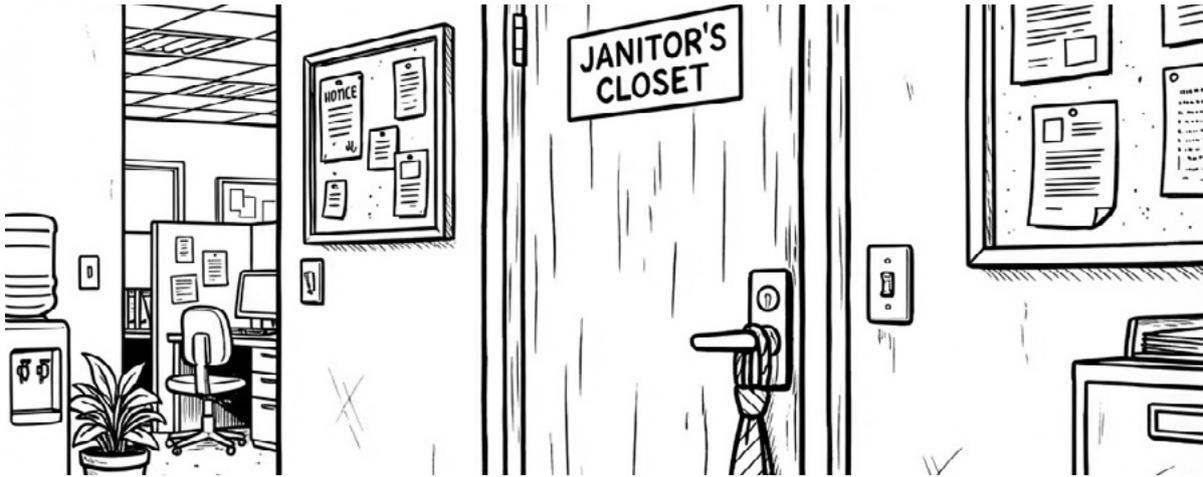
# Chapter 3:

# Why Your “Single Source of Truth” Is Neither

Every IT leader inherits the same promise: “We’ll create a single source of truth.”

Six months later, finance is still reconciling three versions of revenue in Excel. Product team can’t get accurate customer data. The board asks “which regions are growing?” and it takes a week to answer with confidence.

What happened? Declaring a single source of truth isn’t the same as creating one.



## THE DIRTY SECRET ABOUT DATA QUALITY

Whether you've acquired companies, grown organically, or both – often your data is a mess.

### The “revenue reporting system” is actually:

- Data exported from the ERP
- Manual adjustments in Excel
- “Corrections” from the finance director’s personal spreadsheet
- Historical data that doesn’t reconcile
- Assumptions that were true in 2019 but nobody’s validated since

Multiple product lines have different customer definitions. Geographic expansions created regional data silos. That “temporary” workaround from two years ago is now load-bearing.

This is the data landscape most IT leaders inherit. The good news: it’s solvable with the right approach.

## WHY THIS MATTERS FOR GROWTH

Bad data doesn’t just make reporting slow. It makes scaling dangerous.

**Scenario:** The board asks “Which customer segments should we double down on for the new product launch?”

**With clean data:** You pull the report. Growth is clearly in mid-market enterprise customers in the UK and Germany. You build the product roadmap and go-to-market strategy accordingly.

**With messy data:** You spend a week reconciling customer definitions across regions and product lines. Mid-market means different things in different systems. Customer segments don’t align. By the time you answer, the competitive window has narrowed. The product launch timeline slips.

Bad data is a growth velocity problem that compounds across every strategic decision.

## WHY “SINGLE SOURCE OF TRUTH” PROJECTS FAIL

### Failure Pattern 1: Boiling the Ocean

“We’ll create an enterprise data warehouse that consolidates everything.”

#### **Eighteen months later:**

- £4M spent
- 60% of the warehouse built
- Nobody using it because it doesn’t have the data they need yet
- Project quietly dies

**What works:** Pick one high-value use case. Solve it completely. Then expand.

### Failure Pattern 2: Technology Without Governance

You implement a data platform. Now what?

Who decides which system is the source of truth? When data quality issues get fixed vs. worked around? How to handle conflicting definitions?

Without clear ownership, the new platform becomes another system with its own version of truth.

**What works:** Governance first, technology second.

### Failure Pattern 3: Perfection as the Enemy

“We can’t release this dashboard until every data quality issue is resolved.”

Meanwhile, the board makes decisions on Excel spreadsheets that are definitely wrong.

Perfect data doesn’t exist. Waiting for it means decisions get made with worse data, not better.

**What works:** Good-enough data, clearly caveated, delivered fast.

## THE ONE METRIC THAT MATTERS

Time from “question asked” to “answer delivered” for the top 10 business questions.

**If the CEO asks “How much revenue did we do last month across all entities?” and the answer takes:**

- **15 minutes:** Your data is excellent
- **4 hours:** Your data is good enough
- **2 days:** Your data is broken
- **“We’ll get back to you”:** Your data needs urgent attention



## WHAT ACTUALLY WORKS: THE INCREMENTAL APPROACH

### Step 1: Pick One Use Case That Matters

Don't build a data warehouse. Solve a specific problem.

**Example:** “Finance spends 3 weeks closing books after each acquisition because revenue data doesn't reconcile.”

**Target:** Get that down to 3 days.

### Step 2: Build the Minimum Viable Pipeline

What's the absolute minimum to solve that problem?

- Extract revenue data from each ERP
- Standardise the format
- Load into a single reporting database
- Build a reconciliation dashboard

### Step 3: Prove the Value

Get finance using it for one month-end close. Measure the time saving.

If it worked, you've earned the right to expand.

### Step 4: Expand Incrementally

Apply the same pattern to the next problem:

- “Sales can't see pipeline across entities” > Build cross-entity CRM reporting
- “HR doesn't know accurate headcount” > Standardise HR data feeds

Each success builds momentum. Each use case reuses infrastructure from the last.

Within 18 months, you have an enterprise data platform - not because you set out to build one, but because you solved 12 business problems that needed similar infrastructure.

## THE GOVERNANCE MODEL THAT WORKS

Most data governance programmes fail because they’re too heavy. Endless committees. Approval processes that take weeks.

Here’s what works: Lightweight governance with teeth.

### The Rules

- 1. Every dataset has one owner** - A real person. They’re accountable for accuracy and access.
- 2. Conflicts get escalated fast** - Two systems claim to be source of truth? Goes to CFO. Decided within 48 hours.
- 3. Quality issues block releases** - If critical data quality drops, dashboard shows a warning.
- 4. Exceptions require a business case** - Want to keep your legacy system? Write down cost vs. benefit. Get executive approval.
- 5. Migration is mandatory** - Once target platform is chosen, migration isn’t optional.

### The Structure

- **One data lead** (full time) - Sets standards, breaks ties
- **Dataset owners** (part of day job) - Finance director owns financial data, sales director owns CRM data
- **Monthly review** (30 minutes) - Top issues, upcoming migrations, policy decisions

## HOW TO MAKE REPORTING BORING

Nobody should say “Wow, amazing dashboards!”

They should say “I got the answer I needed” and move on.

Boring reporting is the goal. It means the data is reliable, the answer is obvious, no one questions the numbers.

## THE BORING DASHBOARD CHECKLIST

- 1. Answer one question per dashboard** - Don’t create a kitchen sink
- 2. Default to the right view** - Show the most relevant data immediately
- 3. Make anomalies obvious** - Red means problem. Green means good
- 4. Update automatically** - No “refresh data” buttons
- 5. Load in under 3 seconds** - If users wait, they’ll go back to Excel
- 6. Work on mobile** - The CEO’s checking revenue from their phone

# Chapter 4: Why Your IT Team Is the Bottleneck

(And How to Fix It)

Every growth initiative follows the similar pattern:

Week 1: "We need infrastructure for the new product launch / market expansion / acquisition."

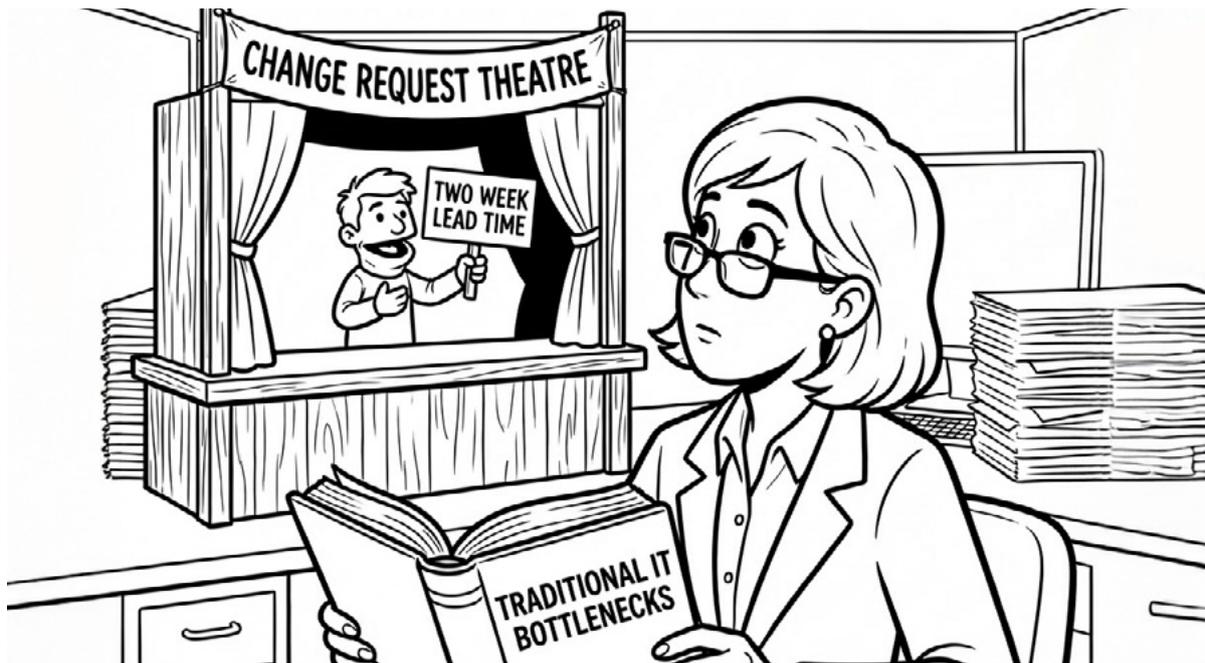
Week 3: Still waiting on the change request to be approved.

Week 5: Environment finally provisioned. Missing half the required access.

Week 7: Second change request to fix the first one.

Week 9: Infrastructure ready. Timeline already shot. Competitive window narrowing.

The technology isn't the problem. Your team's operating model is.



## THE HIDDEN BRAKE ON SPEED

You can have the best target architecture in the world. But if provisioning an environment takes 6 weeks, configuring access takes 3 change requests, and deploying code requires a 40-person CAB meeting, you're still slow.

PE-backed businesses don't have time for processes designed for enterprises that change quarterly. You need a team that moves at growth speed - whether that's launching new products, entering new markets, scaling operations, or integrating acquisitions.

## THE TRADITIONAL IT BOTTLENECKS

**Manual provisioning:** Every new environment is hand-built. Takes 2 weeks even though it's been done 47 times before.

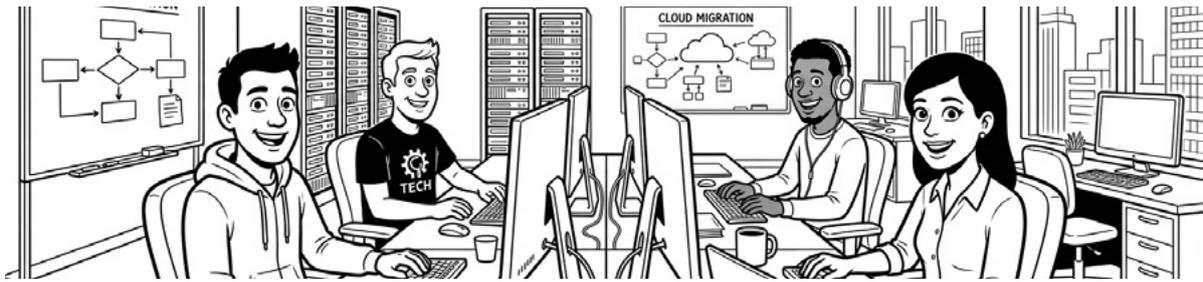
**Change request theatre:** A 12-page form to add a user. Approval chain that includes people who don't understand the request. Two-week lead time for "standard changes."

**Security as a gate:** InfoSec reviews happen at the end. They find problems. Everything goes back to the start. Launch delays by months.

**Siloed teams:** Infrastructure team doesn't talk to app team. App team doesn't talk to security. Everyone blames everyone else when product launches miss their window.

**Tribal knowledge:** The one person who knows how the system works is on holiday. Everything stops.

This worked fine when IT was about "keeping the lights on." It's a disaster when you need to launch three products, expand to two new markets, and integrate an acquisition - all in 18 months.



## WHAT MODERN IT TEAMS LOOK LIKE

**The IT leaders, who enable fast growth, have fundamentally different operating models:**

### 1. Infrastructure as Code (Not as Ticket)

**Old way:** Submit a ticket for new infrastructure. Wait for someone to manually configure. Hope they don't miss something. Two weeks later, maybe you have what you need.

**New way:** Define infrastructure in code. Run a script. Environment provisions in 4 hours, identical every time, with security and compliance built in.

This isn't about being trendy. It's about repeatability at speed.

When product team needs infrastructure for the new SaaS offering, you don't start from scratch. You run the same automation that's been proven on the last three launches. Cloud resources, network config, security baseline, monitoring - all provisioned before lunch.

When you're expanding to a new geographic market, the infrastructure pattern is the same. Just deployed to a different region.

When an acquisition lands, same foundation - just configured for their specific requirements.

### 2. Platform Engineering Over Ticket Engineering

**Old way:** Development teams submit tickets to IT for every environment, deployment, or config change. IT becomes a queue. Product launches wait in line behind BAU work.

**New way:** Build self-service platforms. Development teams provision what they need within guardrails. IT builds the rails, teams run on them - fast.

**Example:** Instead of "submit a ticket for a dev environment for the new product," it's "log into the portal, click provision, select template, environment ready in 20 minutes."

This doesn't mean chaos. The platform enforces standards - security baselines, network configs, compliance policies, cost controls. But it removes IT as the bottleneck for every small decision that would otherwise slow product development or market expansion.

### 3. Security Built In, Not Bolted On

**Old way:** Build first, security review at the end. InfoSec finds problems three weeks before launch. Everything redone. Product launch misses market window. Competitor launches first.

**New way:** Security policies embedded in the platform. Automated compliance checks. Security involved from Day 1, not Week 12. Launch on time, securely.

When security is baked into your infrastructure-as-code templates and deployment pipelines, it scales with growth. Every new product inherits the same security baseline. Every deployment runs through the same automated checks. Every market expansion starts secure.

This makes growth faster, not slower. Because you're not discovering security issues three weeks before go-live and scrambling to fix them.

### 4. DevOps Pipelines Replace Change Committees

**Old way:** CAB meeting every two weeks. 40 people reviewing changes they don't understand. Product deployments wait the same duration as infrastructure changes. Time-to-market measured in months.

**New way:** Automated deployment pipelines with appropriate approval gates. Low-risk changes deploy continuously. High-risk changes get human review. Product releases go out weekly, not quarterly.

**The shift:** From "prevent all change" to "make change safe and fast."

In a PE environment, waiting two weeks to deploy a product enhancement is unacceptable. Your competitor isn't waiting. But deploying without appropriate testing and rollback capability is also unacceptable.

The answer isn't more process. It's better automation with built-in safety.

### 5. Modular, Repeatable Patterns

**Old way:** Every initiative is bespoke. Every product launch needs custom infrastructure. Every market expansion is unique. Nothing is reusable. Teams start from zero every time.

**New way:** Build once, reuse everywhere. Standard templates for common workloads. Modular components that snap together.

When you're launching a new product, you're not designing cloud architecture from scratch. You're deploying the same proven patterns you've used before - configured for this product's specific requirements.

When you're expanding to APAC, you're using the same infrastructure patterns that work in EMEA and North America - just deployed to Singapore instead of Frankfurt.

When you acquire a company, same patterns - but tailored to their scale and requirements.



## WHY THIS MATTERS FOR MODERN CLOUD PLATFORMS

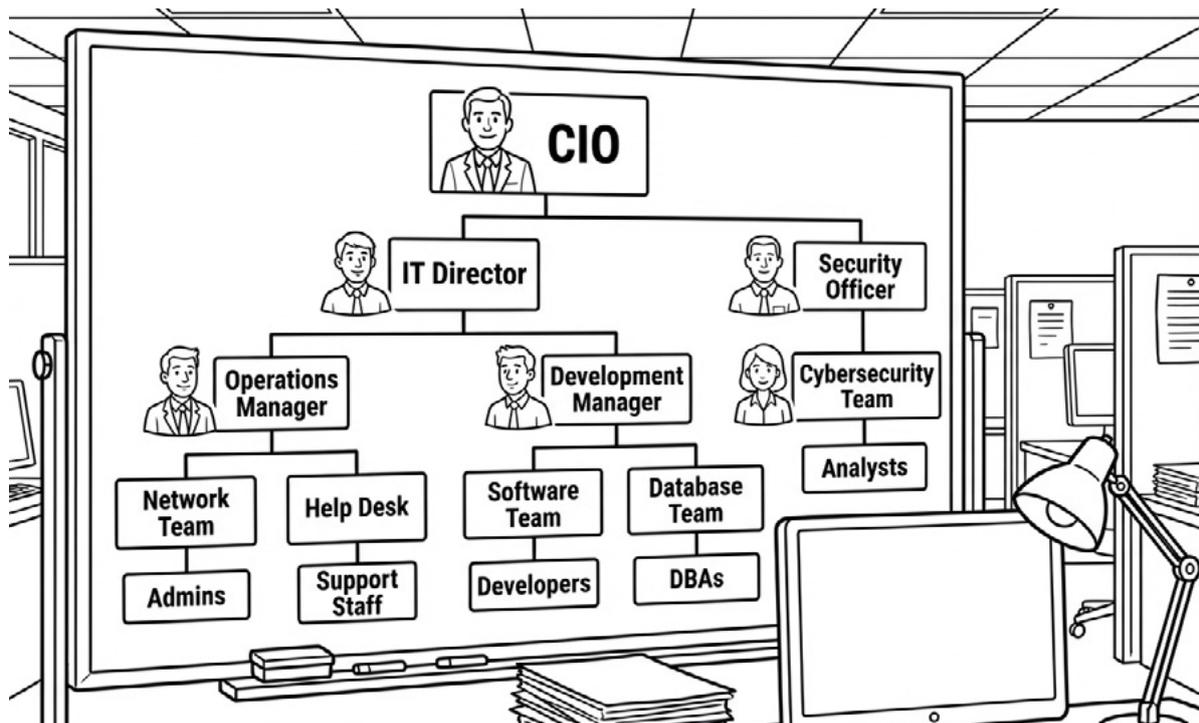
Modern cloud platforms enable this transformation. But only if you use them properly.

**The trap:** Treating Azure (or AWS, or GCP) like an expensive data centre. Lifting and shifting VMs. Manually configuring resources. Not using automation, PaaS services, or platform capabilities. Still slow.

**The opportunity:** Using cloud as a platform for growth:

- **Infrastructure as Code:** ARM templates, Bicep, Terraform - pick one, use it consistently across all growth initiatives
- **CI/CD Pipelines:** Azure DevOps or GitHub Actions - automated pipelines from commit to production, enabling fast product releases
- **Azure Policy:** Security and compliance enforced automatically, not through tickets that slow launches
- **PaaS services:** Don't build and manage databases for every new product, use Azure SQL. Don't patch servers, use App Service. Focus team effort on differentiation, not undifferentiated heavy lifting.
- **Landing zones:** Microsoft's Cloud Adoption Framework provides the blueprint for repeatable deployment patterns - whether you're launching products, expanding geographically, or integrating acquisitions

The companies that launch fastest, scale smoothest, and integrate most efficiently aren't necessarily more skilled. They've just automated what everyone else does manually.



## THE TEAM STRUCTURE THAT ENABLES GROWTH

Traditional IT org charts don't work at PE speed.

### What Doesn't Work

**Siloed by technology:** Server team, network team, database team, app team. Every product launch or market expansion requires coordination across four teams with different priorities and timelines. Everything takes weeks longer than it should.

**Ticket-driven:** Every request becomes a queue. IT team spends their day processing tickets instead of building capability that accelerates growth.

**Project-based:** Spin up a team for an initiative. Disband after go-live. Next product launch or acquisition starts from zero again. Never getting faster.

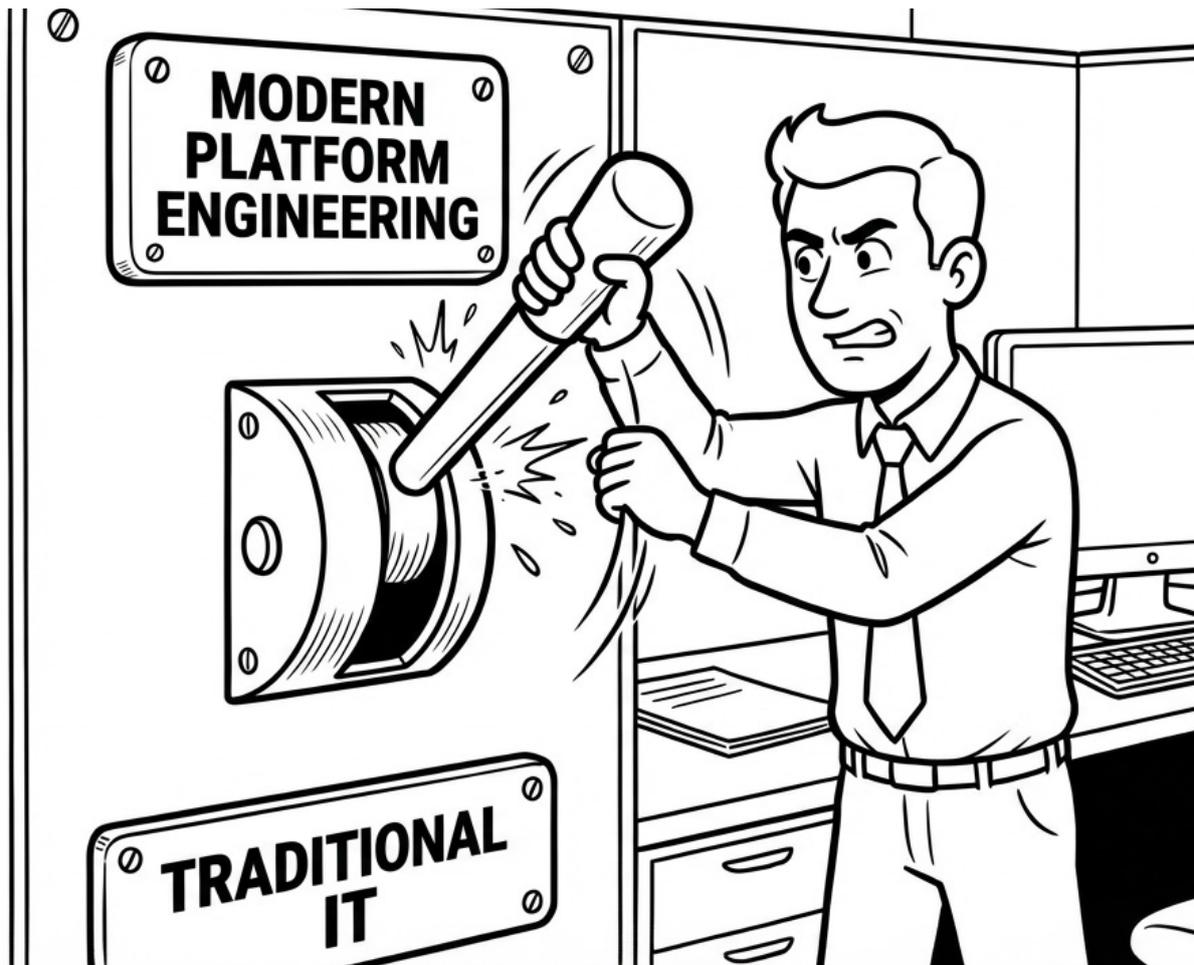
### What Works

**Product teams:** Small, cross-functional squads owning specific platforms (e-commerce platform, data platform, identity platform, customer portal). They build it, they run it, they improve it. When product teams need to launch fast, these platforms are ready.

**Platform team:** Builds self-service capabilities that other teams consume. They create the rails everyone else runs on to move fast.

**Growth squad:** Whether it's integrations, product launches, or market expansions – it's a permanent team that gets better every time.

**The goal:** Remove dependencies. Teams should be able to move fast without waiting on other teams or lengthy approval chains.



## THE TRANSFORMATION ROADMAP

You can't flip a switch from traditional IT to modern platform engineering. But you can start.

### **Month 1-3:** Automate the Painful Stuff

Pick the three most repetitive, time-consuming manual processes that slow every growth initiative. Automate them.

#### **Candidates:**

- Environment provisioning (from 2 weeks to 4 hours)
- User account creation and access management
- Standard application deployments
- Security baseline configuration

Don't try to automate everything. Just the things that slow every product launch, market expansion, or acquisition.

### **Month 4-6:** Build the Self-Service Platform

Create a portal or process where teams can provision common resources without submitting tickets.

**Start simple:**

- Dev/test environments for product teams
- Standard application templates
- Database provisioning
- Network access requests

Embed security policies and compliance checks. If it goes through the platform, it's secure by default. Product teams move fast without compromising security.

**Month 7-9: Deployment Pipelines**

Move from manual deployments to automated pipelines.

Start with less critical applications. Build confidence. Expand to customer-facing products and revenue-critical systems.

Include automated testing, security scanning, and rollback capability. The goal is safe, frequent deployments - enabling product velocity without sacrificing stability.

**Month 10-12: Kill the Change Committee**

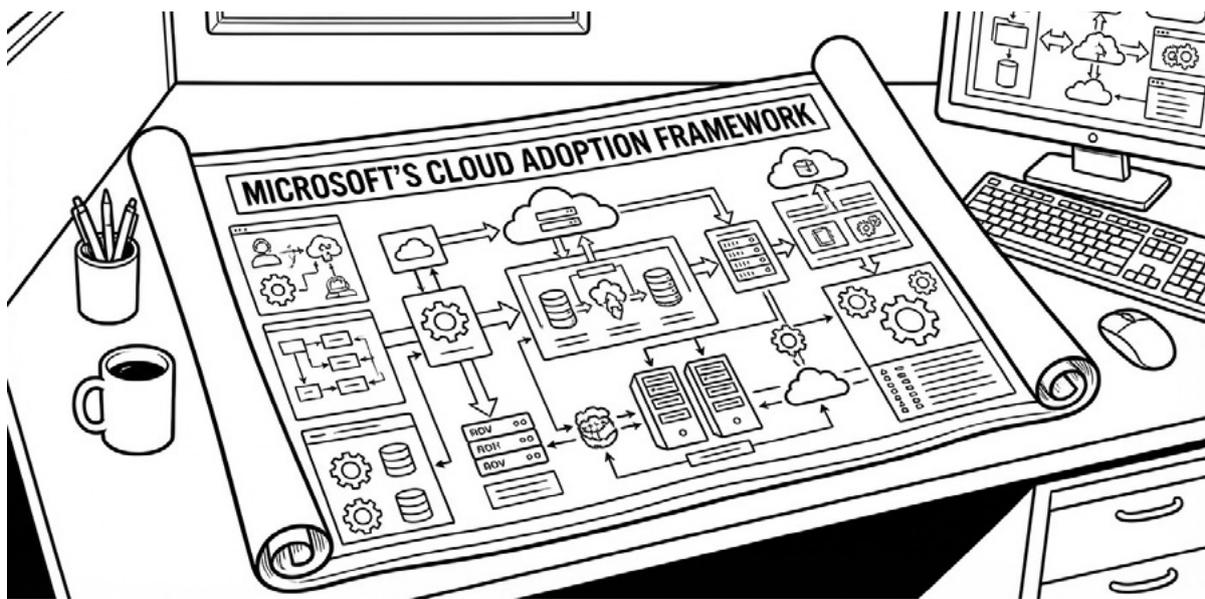
Once you have automated pipelines with appropriate controls, you don't need a weekly CAB meeting for standard changes.

**Keep human approval for:**

- Changes to production databases with customer data
- Network architecture changes
- Security policy modifications

**Automate everything else:**

- Product deployments and updates
- Configuration changes
- Scaling operations
- User access (within policy)



## THE CLOUD LANDING ZONE PATTERN

If you're modernising on Azure, Microsoft's Cloud Adoption Framework provides the blueprint.

The landing zone concept is simple: build the foundation once, deploy new workloads repeatedly - whether that's new products, new markets, or new acquisitions.

### What a landing zone includes:

- Network architecture (hub-spoke, connectivity to on-prem if needed)
- Identity and access management (Azure AD integration, RBAC)
- Security baseline (Azure Policy, encryption, monitoring)
- Logging and monitoring (centralized, consistent)
- Governance (tagging, cost management, compliance)

### Why it matters for PE-backed growth:

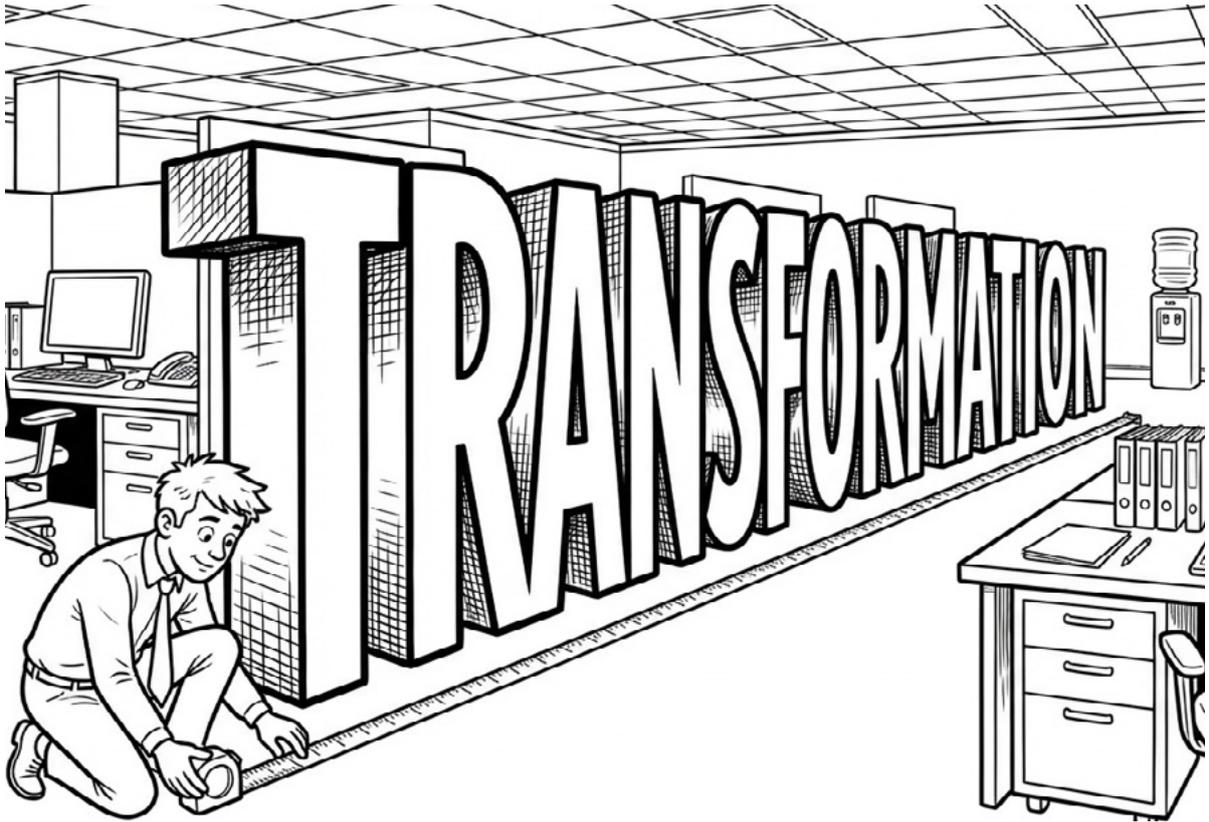
When product team needs to launch the new SaaS offering, you're not designing network architecture from scratch. You're deploying into an existing landing zone that's proven, secure, and well-understood.

When you're expanding to a new geographic market, same pattern - just deployed to the appropriate Azure region with any required data residency controls.

When an acquisition lands, you're onboarding them into the same foundation that's already supporting your existing products and operations.

Provisioning time drops from weeks to hours. Because the foundation exists. You're just adding new workloads or entities to proven infrastructure patterns.

This is the difference between "let's scope this launch" and "run the deployment script, we go live Monday."



## MEASURING THE TRANSFORMATION

**Track these to know if modernisation is working:**

### **Time to provision new environment**

**Target:** <4 hours (down from 2-4 weeks)

### **Time to market for product enhancements**

**Target:** Weekly releases (up from quarterly)

### **Change request volume**

**Target:** 60% reduction (automated what used to need tickets)

### **Growth initiative velocity**

**Target:** Each product launch / market expansion / acquisition 20% faster than the last

### **Team efficiency**

**Target:** Same team size supports 3x more growth initiatives

If these numbers aren't improving, your modernisation is theatre. You've adopted new tools without changing the operating model that kills speed.



## THE UNCOMFORTABLE TRUTH

Modernising your IT team's operating model is harder than modernising technology.

Technology is easy. Buy Azure. Deploy some automation.

**The hard part:** Convincing people to work differently.

- The infrastructure team who've manually built servers for 15 years
- The change manager who believes every change needs a 12-page form
- The security team who wants to review everything manually at the end
- The leadership who equates "control" with "process"

This transition requires clear communication about why these changes matter and support for people as they adapt. Some will embrace it quickly. Others will need more time and support. A few may decide this isn't the environment for them.

The IT leaders who succeed focus on making the case for change, providing the training and support people need, and being clear about expectations. Because it's the operating model that determines whether IT accelerates growth or slows it.

# Chapter 5: Why Change Programmes Fail (Even When Technology Works)

The new ERP is live. Training complete. Technology works perfectly.

Six months later, 30% of users still using the old system "just in case."

Another 40% working around the new one with Excel.

This is the adoption crisis. And it kills more integrations than technology failures.



## THE PROBLEM NOBODY BUDGETS FOR

Most integration budgets include software, infrastructure, consultants, data migration, training.

### What they don't include:

- Six months of productivity loss while people learn
- Shadow IT that springs up when new platform doesn't do what the old one did
- Decision paralysis when different entities use different processes
- Quiet rebellion of teams who refuse to adopt

## THE MATH OF LOW ADOPTION

100 users × £80K salary × 20% productivity loss × 6 months = £800K

Add cost of parallel systems, data quality issues from dual entry, delayed synergies.

That "low disruption" integration just cost £2M+ in hidden costs.

## WHY TRAINING DOESN'T CREATE ADOPTION

Every integration includes training. But why does adoption still fail?

Because training teaches people how to use the system. It doesn't make them want to.

### People adopt new systems when:

- 1. They understand why it matters** - Not "because head office said so"
- 2. They see people like them succeeding** - Not consultants, but actual colleagues
- 3. The new system makes something genuinely better** - Faster, easier, less painful
- 4. Resistance has consequences** - Clear expectations and follow-through
- 5. They have support when they struggle** - Real support, not "submit a ticket"

Notice what's missing? Training sessions.



## THE LOCAL HERO STRATEGY

Find the heroes in each entity and amplify them.

### Week 1-2: Identify the Heroes

**In every acquired entity, find people who:**

- Learn new systems quickly
- Help colleagues without being asked
- Have credibility with the team
- Are genuinely excited about improving things

Usually 2-3 people per entity of 50-100 staff.

### Week 3-4: Equip Them

- Early access to new system
- Direct line to your integration team
- Authority to make local decisions
- Recognition and air cover from leadership

### Week 5+: Deploy Them

They're not trainers. They're champions.

When someone struggles, they ask their local hero. When there's resistance, the hero explains why it matters. When there's a problem, the hero escalates with credibility.

**This scales.** Two heroes supporting 50 users is sustainable. A central team supporting 500 users is not.



## CHANGE COMES FROM "SOMEONE LIKE ME WHO SAYS IT'S WORTH IT."

### The First-Month Playbook

#### Week 1: Intensive Support

- Integration team in full support mode
- Respond to issues within 30 minutes
- Leadership visibility - they should be using it too

#### Week 2: Quick Fixes

- Deploy fixes for most common issues
- Celebrate early adopters publicly
- Address major concerns

#### Week 3: Gentle Pressure

- "When are you switching?" becomes "Why haven't you?"
- Local heroes check in with their teams
- Managers expect new-system outputs

#### Week 4: Hard Cutoff

- Old system access removed
- Reports only run from new system
- No more "temporary" exceptions



## HOW TO MAKE PEOPLE WANT TO SWITCH

Make the new system noticeably better at something people care about.

### Examples That Work

#### For sales reps:

“The new CRM auto-logs calls and pre-fills customer data. Five minutes saved per prospect.”

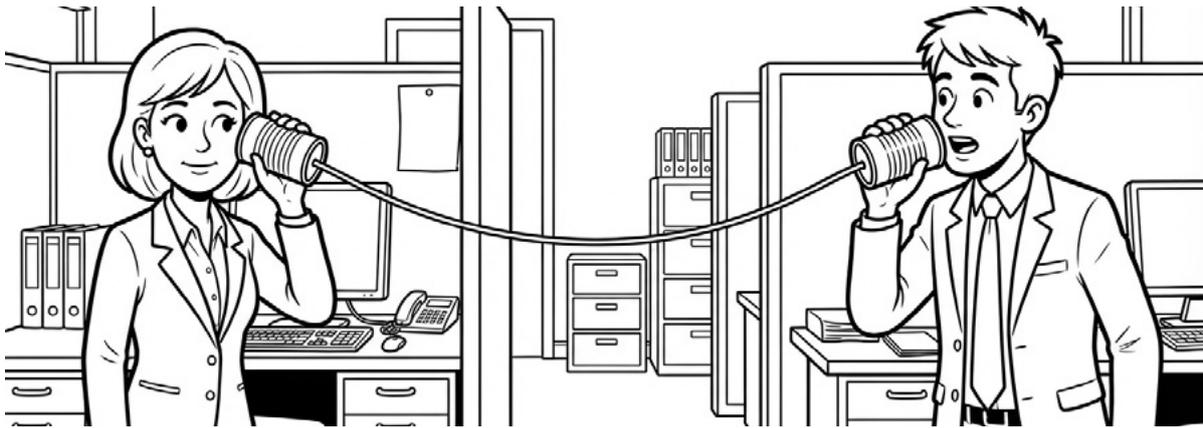
#### For finance:

“Month-end close runs in 3 days instead of 2 weeks. No more weekend work.”

#### For operations:

“Approval workflows automated. No more chasing signatures across three systems.”

Find the thing that makes people say “Oh, that’s why we’re doing this.”



## COMMUNICATION THAT WORKS

Most M&A programmes communicate too much and say too little.

### What Works

1. **“What’s Changing This Week”** - Three bullets. One email. Every Monday.
2. **Before/After Visuals** - Show, don’t tell.
3. **Video Walkthroughs (30 seconds)** - “Here’s how to log an expense.”
4. **Weekly Quick Win Spotlight** - “Shoutout to Sarah who closed books 3 days faster.”
5. **Transparent Blocker Reports** - “Here’s what slowed us down and how we’re fixing it.”

### What to Stop

- Monthly town halls where leadership reads slides
- “Change champion” newsletters nobody asked for
- Celebration emails about “milestones” that don’t matter
- Surveys that ask for feedback but don’t act on it
- Measuring Change That Actually Helps

### Track these:

1. **Active Usage** - % of users who logged in and completed real work  
**Target:** 90% within 30 days
2. **Old System Usage** - How many still accessing legacy  
**Target:** <5% by Day 60
3. **Productivity Recovery** - How long to return to pre-migration levels  
**Target:** Full recovery within 90 days
4. **Support Ticket Trend** - Are tickets decreasing or steady  
**Target:** 50% reduction by Week 8

# Chapter 6: The Only KPIs That Matter

Your monthly IT report shows:

- 99.7% uptime
- 2,847 tickets closed
- 12 projects in flight
- £3.2M spent YTD

The board nods politely and moves on.

Why? Because none of those numbers answer: "Is IT accelerating value creation or slowing it down?"



## WHY INVESTORS DON'T CARE ABOUT YOUR SLAS

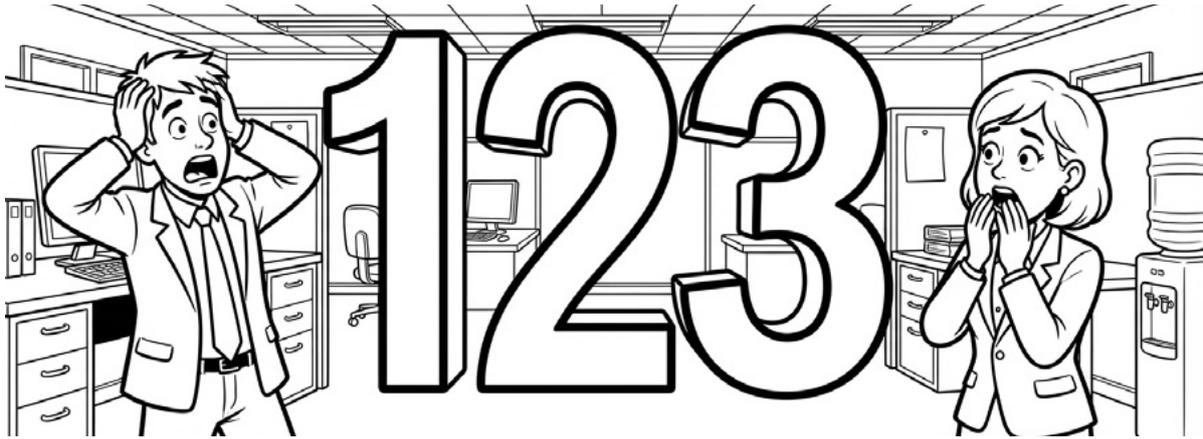
Traditional IT metrics track operational health: uptime, tickets, projects, security incidents.

These are necessary. But they're table stakes, not value drivers.

Hitting 99.9% uptime doesn't create value. It just avoids destroying it.

### **What investors care about:**

- Are we integrating acquisitions faster?
- Are we capturing synergies?
- Are we ready for the next deal?
- Will IT slow down the exit?



## THE THREE NUMBERS THAT PREDICT EXIT VALUATION

When a potential acquirer does tech due diligence 3-5 years from now:

### 1. Technical Debt Load

How much legacy complexity will they inherit?

#### How to measure:

- % of estate on standard platforms
- Number of legacy systems still running
- Critical systems without succession plans

**Target:** <10% of estate on non-standard platforms by exit

### 2. Integration Readiness

How fast can they absorb this business?

#### How to measure:

- Time to integrate a new acquisition (trending down)
- API coverage for core systems
- Data extraction capability

**Target:** Full entity integration in <90 days

### 3. Run Cost Efficiency

Is IT spend reasonable or bloated?

#### How to measure:

- IT spend per employee
- IT spend as % of revenue
- Cost reduction trend since acquisition

**Target:** IT spend at or below industry benchmark

## THE OPERATIONAL KPIs THAT DRIVE VALUE

### Revenue & Growth Enablement

#### Speed of New Product/Market Launch

**Target:** <30 days from decision to technical readiness

#### Integration Velocity

**Target:** Day 1 for basic reporting, Week 4 for full integration

### Cost & Margin Efficiency

#### IT Spend vs. Budget

**Target:** Within 5% of budget

#### SaaS/Cloud Optimisation

**Target:** 15%+ reduction in first year, ongoing 5% annually

#### Cost Per Employee

**Target:** Flat or declining as headcount grows

### M&A Velocity

#### Time to Integrate

**Target:** <90 days (and trending down)

#### Percentage of Estate Migrated

**Target:** 80% by Month 6, 95% by Month 12

#### Integration Cost

**Target:** 20% cost reduction from Deal 1 to Deal 3

### Operational Stability

#### Business-Critical System Uptime

**Target:** 99.9% for revenue-critical systems

#### Mean Time to Resolution (Critical)

**Target:** <4 hours

### Security & Compliance

#### Material Security Incidents

**Target:** Zero incidents that impact customers or expose data

#### Audit Pass Rate

**Target:** 100% pass with zero critical findings



## THE DASHBOARD THAT KEEPS YOU FUNDED

Bring this to your monthly board meeting:

Metric	Target	Current	Trend
Days to integrate last acquisition	<90	78	↓
% of estate on standard platforms	>80%	73%	↑
IT cost per employee	<£2,500	£2,380	→
Integration cost (last deal)	-20% vs Deal 1	-18%	↑
Critical system uptime	99.9%	99.94%	→
Material security incidents	0	0	→

**One slide. Six metrics. Traffic lights.**

**Green** = no discussion needed

**Amber** = watching this

**Red** = here's the fix

**How to Report Without Creating Theatre**

**Monthly:** The One-Pager

Six KPIs. What changed. What you're fixing.

Takes 2 minutes to review.



## QUARTERLY: THE VALUE STORY

### How IT contributed to business outcomes:

- “Integration completed 2 weeks ahead, bringing synergies forward by £400K”
- “SaaS rationalisation saved £800K annually while improving security”
- “Platform standardisation reduced next integration timeline from 120 to 75 days”

### Annually: The Exit Readiness Check

Where are we on technical debt, integration readiness, and cost efficiency?

#### The Exit Readiness Audit

Three years from now, tech due diligence will examine your estate.

#### Red Flags That Cost Millions

1. Legacy systems on end-of-life technology
2. Significant undocumented customisations
3. Each entity still on separate instances
4. No single source of truth for critical metrics
5. Material security or compliance risks
6. Critical systems with single-person dependency
7. IT costs significantly above industry benchmark

#### Green Lights That Protect Valuation

1. Modern, well-supported technology
2. Documented, minimal customisation
3. All entities on common platforms
4. Single source of truth for critical metrics
5. No material security or compliance findings
6. Well-documented architecture and processes
7. IT spend at or below industry benchmarks

Run an internal version every 6 months. Find problems early. Fix them before they become valuation risks.



## THE ROAD AHEAD

The IT leaders who succeed in PE don't work harder than the ones who struggle. They make different decisions earlier.

- They standardise when it's uncomfortable.
- They cut off old systems when people resist.
- They build repeatability when it feels premature.
- They track value creation when process metrics would be easier.

## WHAT GOOD LOOKS LIKE (18 MONTHS IN)

### The CEO's phone buzzes. She steps out. Returns:

"We're acquiring MarketLeader. £200M deal. 600 employees. Close in 45 days. Can we integrate in 10 weeks?"

**You:** "Yes. We'll have them on our platforms, reporting consolidated, and operationally integrated by Week 10. The team knows what to do."

No panic. No scoping. No heroics. Just execution.

That's the machine working.

## THE LONG GAME

**Year 1:** Stabilise, standardise, prove you can deliver

**Year 2:** Scale the machine, make integrations repeatable

**Year 3:** Optimise, reduce cost, prepare for exit

**By Year 3,** your job should be easier than Year 1 - even though you've integrated 4 companies.

That's how you know the machine is working.

Now go build it.

# Looking to scale without the chaos?

GET IN TOUCH

[sales@synextra.co.uk](mailto:sales@synextra.co.uk)

0161 883 1383

Or explore our website:

