



Transforming Digital Infrastructure: How Business Leaders Can Manage Disruptive Technology in the AI Era

AI makes it easier than ever to change how work is completed, at a previously unfathomable speed and scale. It also makes it easier for every employee, team, and system to pull the business in a different direction before management can recognize and react to what is happening. The next challenge of digital transformation extends beyond the question of technical feasibility: it's one of adoption and leadership.

Prepared by LoopHole Automation

Written by: Sean Sakaguchi, COO

sean@loopholeautomation.com



EXECUTIVE SUMMARY

Digital transformation is a leadership system

Most companies do not have a technology problem that can be solved by buying more technology. They have a coordination problem. Information lives in different places. Employees have built personal workarounds. Important processes depend on memory, judgment, and informal handoffs that no one has ever written down.

AI did not create those conditions, but it has made them harder to ignore. An employee can now build a personal workflow in an afternoon, generate a finished-looking report in minutes, or hand an agent a task that once required several people. That freedom feels productive to the individual. Across a team, it can create several versions of the same process, several definitions of acceptable quality, and no reliable way to know which one represents the business.

This is the tension at the center of modern transformation. The method that feels fastest for one employee is often not the method that lets a company perform reliably at scale. A business has to deliver consistent work across different people, customers, locations, and periods of pressure. That requires some conformity. It requires shared definitions, shared systems, and limits on how much of the operating model each employee gets to reinvent.

Organizations will not benefit from AI simply by finding more places to use it. They will benefit when leaders decide how work should move, what the business must do consistently, where people should retain ownership, and what boundaries automated systems must follow.

This paper documents the problems we encounter repeatedly in our work at Loophole Automation. We have helped scores of clients redesign workflows, connect systems, automate routine work, and adopt AI. Many first came to us after modernization efforts led internally or by other providers had failed. In those earlier efforts, the same underlying weaknesses had undermined otherwise promising investments. AI is making those weaknesses more consequential, and organizations need to adapt how they coordinate work, set standards, and maintain accountability.

The businesses that benefit most from AI will not be the ones that use it everywhere. They will be the ones that decide where individual judgment creates value and how technology should connect the two.

A business is not a collection of personal workflows. It is a promise that work will be done reliably, regardless of who performs it.

The argument in four parts

- | | |
|----|--|
| 01 | Most leaders aren't ready for new technology |
| 02 | Individual efficiency is not company performance |
| 03 | Leadership is evolving away from management and towards coordination |
| 04 | Faster execution is counterproductive to success |

PART I

Most leaders aren't ready for new technology

Companies rarely waste a technology investment because they chose the second-best product. They waste it because fear, a vague ambition, or a visible symptom was allowed to stand in for a diagnosis.

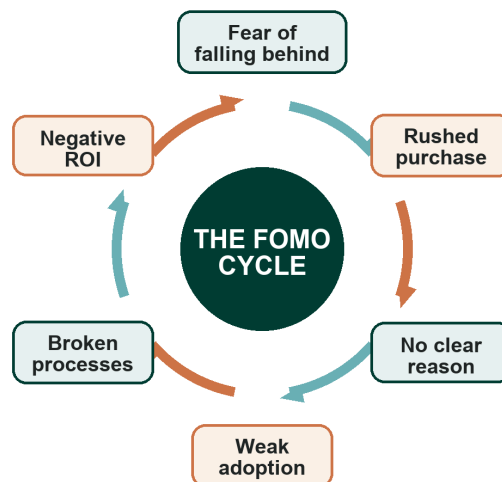
1. How technology FOMO turns into negative ROI

The first meeting often sounds urgent. A competitor has announced an AI initiative. A vendor says an entire category of work is about to change. Someone on the board asks why the company is behind. By the end of the conversation, waiting feels reckless and buying something feels like leadership.

That pressure is effective because it reflects a real concern. Technology is changing quickly. A company that refuses to learn will fall behind. The mistake is assuming that the correct response to urgency is a fast purchasing decision.

Once the fear takes hold, the conversation narrows. People compare features, plans, and demonstrations. They ask how quickly the tool can be deployed. The harder questions begin to sound like delay. Which business result should change? Which employees will work differently? What happens to the old process? What information does the tool need? Who will decide whether the output is good? Those questions are left for implementation, after the contract is signed.

When a tool is purchased without a clear reason, employees receive a new login but no shared answer about what problem it should solve or when it should be used. If the new tool does not clearly improve the process, or leadership does not explain that improvement, adoption becomes inconsistent. A few people experiment enthusiastically. Others ignore it. Some use it for work the sponsor never considered.



Weak ROI renews the fear and makes the next rushed purchase feel urgent.

Inconsistent adoption breaks the process. Some employees complete the task in the new tool while others keep using the old method. Handoffs stop matching, data splits across systems, training has to cover multiple methods, and managers spend time reconciling the differences. The result is negative ROI: the company has added cost, duplicate work, messy data, and new coordination problems without removing the old process.

Then the weak return creates a new sense of danger. The company concludes that it still has not transformed, so it begins searching for the next platform all over again.

Moving quickly still matters, but speed should apply to diagnosis and testing rather than purchasing. A narrow experiment on real work can reveal whether the problem is worth solving, whether the tool changes the result, and what the organization would have to change around it. A demonstration cannot answer those questions. A contract cannot answer them either.

I 2. The problem is usually misdiagnosed before the tool is chosen

A company sees late reports and assumes it has a reporting problem. It sees an overloaded employee and assumes it has a staffing problem. Customers wait for answers, so leadership assumes the team needs a better communication platform.

Those symptoms describe where the pain appears. They do not explain what causes it.

The report may be late because three people enter the same information in different formats and someone has to repair the differences at month-end. The employee may be overloaded because every exception requires approval from the one manager who understands the history. The customer may be waiting because the answer exists, but it is split across email, a project system, a spreadsheet, and the memory of the person who handled the last call.

Software demonstrations and sales people are partly to blame for this behavior. They encourage leaders to match each visible symptom to an obvious product or purchase. Having reporting problems? Build a new dashboard. Don't have enough capacity to handle more work? Hire another employee. Teams are having issues coordinating? Purchase a new messaging tool. Each answer may improve one surface while leaving the process underneath untouched.

This is how companies digitize their work without improving it. The old handoff becomes a workflow notification. The old spreadsheet becomes an online form. The old approval bottleneck receives a mobile app. The process looks more modern, but the same person still waits for the same information and the same decision.

The reverse can also be true. Every business relies on three core types of infrastructure: physical, human, and digital. Many leaders feel the sense that they're 'not getting enough out of their teams' and immediately diagnose the problem as a human infrastructure one. When work is late, unorganized, or taking too long, the default assumption is that something is wrong with the execution. 'We need to hire more people' or 'we need to hire better people'. But in many cases, what's actually happening is that good people are operating inside systems that aren't built to support them.

The problem is that digital infrastructure challenges rarely look like digital problems; they show up as people problems.

Misdiagnosis also makes the technology stack harder to manage. A tool bought for one symptom becomes another place where data lives. Employees copy information into it because the systems do not share a source of truth. When the result is disappointing, the company cannot tell whether the product failed or whether it was asked to solve the wrong problem.

A useful diagnosis stays with the work long enough to see where information changes hands, where people make judgment calls, where exceptions accumulate, and where one person's preferred method creates work for someone else. Only then does it make sense to decide whether the answer is a process change, a clearer rule, a connection between systems, automation, new software, or no new technology at all.

I 3. 'AI-first' is often a slogan covering a missing strategy

'AI-first' sounds like a strategy because it points toward action. In practice, it often means that leadership wants the organization to use more AI but has not decided what AI should be responsible for.

The phrase hides several different objectives; leadership may want lower labor costs, faster decisions, better customer service, stronger analysis, fewer administrative tasks, or a company that appears "current" to investors and employees. Depending on which of those goals are the real priorities, the appropriate technology, process, or risk decisions look very different. Until the company names the result it wants, 'AI-first' gives every participant permission to imagine a different destination.

Moreover, when 'AI-first' is declared without specific instructions on how to proceed, employees hear a broad instruction to experiment. One person uses AI to prepare client emails. Another uses it to summarize meetings. A third builds a private assistant around project files. Someone else refuses to use it because no one has explained what information is safe to share. All four employees are behaving rationally under an ambiguous directive.

The damage from this trial-and-error appears slowly. Personal experiments become habits. Prompts, file formats, review practices, and definitions of acceptable quality begin to differ by employee. Managers cannot compare outputs because they do not know how it was produced. A useful experiment in one team never becomes available to another. A bad experiment survives because no one owns the decision to stop it.

The company may look active while remaining strategically passive. It has allowed vendors and individual employees to decide where AI belongs, one feature and one workaround at a time.

An AI strategy doesn't necessarily need to have an overly controlled plan for every potential AI use-case. But it does need to name the business problems worth pursuing, the information each use depends on, the authority the system may exercise, the quality standard that applies, and the person who remains responsible. That is enough structure to let experimentation teach the organization instead of fragmenting it.

Moreover, leaders are often poorly equipped to develop the strategy themselves. Non-technical Leaders are asked to weigh model quality, token limits and efficiency, security features, integrations, data access, and the amount of review each tool requires. Most leaders have barely encountered several of those variables, much less learned how to balance them. Choosing a provider because “most people I’ve talked to use Claude Co-work and they like it” is not an operating strategy. The tool has to fit the business problem, workflow, guardrails, and quality standard the company has already defined.

I 4. Most businesses are not ready to implement AI

Most established businesses have a layer of hidden knowledge that makes their systems appear more reliable than they are. Employees know that the customer name in accounting is different from the name in the CRM. They know which project status is never updated, which spreadsheet contains the real schedule, and which field must be ignored because everyone has used it differently for years.

The work continues because people carry the translation layer in their heads. A tenured employee recognizes that two records refer to the same customer. A manager remembers why an exception was approved. A coordinator knows which total is trustworthy and which one must be rebuilt before a meeting.

AI does not inherit that context simply because it can access the systems. It sees conflicting names, incomplete fields, stale records, and undocumented exceptions. Worse, it can turn those contradictions into a fluent answer that looks more certain than the underlying information deserves.

The consequences can be substantial. If AI prepares reports, routes work, or recommends actions from inconsistent data, it distributes the inconsistency through the organization. Different teams receive different versions of reality faster than before. Managers spend more time explaining why the system is wrong. Employees learn to ignore it. Trust falls faster than data quality improves.

The unfortunate reality is that most businesses are not ready to implement AI at the level they imagine. They are behind in at least one of three areas: connected and centralized data, a clear source of truth and shared rules for interpreting facts, or redesigned workflows and guardrails that define what an AI agent may do. Until those foundations are in place, adding AI gives the existing disorder more speed and reach.

Becoming data-ready is the unglamorous work of making information understandable and dependable enough that a person or a system can act on it without relying on private memory. AI can expose the need for clean data - it cannot make the business decision to invest in it.

Individual efficiency is not company performance

AI gives each employee more power to optimize work around personal preference. A company, however, succeeds by producing reliable work across many people. Those two goals do not automatically align.

I 5. What works for one employee can fail across a company

An employee finds a faster way to complete a task. The new method uses a personal template, a private set of prompts, a spreadsheet that matches how that employee thinks, or an AI assistant configured around that employee's files. The work gets done sooner. From the employee's seat, the improvement is obvious.

The company sees a different system. The next employee cannot follow the reasoning. A manager cannot tell whether the same checks were performed. A customer receives a different type of answer depending on who handled the request. When the employee is absent, no one knows where the supporting information lives or how an unusual case was resolved.

This is the difference between local efficiency and organizational performance. Local efficiency asks how quickly one person can finish. Organizational performance asks whether the whole process remains reliable when the work moves between people, volume increases, a mistake occurs, or the original employee is no longer available.

Some shared practices feel inefficient at the individual level for a reason. Required fields take time. Standard templates can feel restrictive. Review steps delay completion. A common naming rule may be less natural than the shorthand one employee prefers. Those constraints often exist so the rest of the organization can understand, compare, audit, and continue the work.

AI sharpens the conflict because it makes deviation cheap. Employees no longer need technical help to invent a new process. They can change the instructions, format, tool, and level of detail on their own. By the time leadership notices, the team may already have five functioning methods and five reasons why each person should be allowed to keep theirs.

The costs show up in places that individual productivity measures miss. New employees take longer to learn. Managers spend more time reconciling output. Data cannot be compared. Customers experience uneven quality. Improvements cannot be rolled out cleanly because no common process exists to improve.

A company is not obligated to maximize every employee's freedom. It is obligated to make good promises to customers and keep them. Individual initiative matters, but it has to improve the shared system rather than leave the company with a patchwork of private processes.

The fastest method for one person can be the most expensive method for everyone who depends on that person's work.

I 6. Unlimited employee freedom destroys shared systems

Tech providers and business leaders often love to tout a new tool's ability to give employees 'freedom'. In practice, expanding employee capabilities without critically analyzing the specific process and output a team needs to share can devastate organizational efficiency. When employees are given open-ended tools and the freedom to build a different process around every preference, that freedom can make one person faster, but it prevents the organization from enforcing consistency, maintaining clean data, and improving one shared way of working.

This isn't some new truth that exists because of AI. You only need to look as far as Excel to realize the danger. Time and again, you can look into a business and find that it's being held together by a series of fragile, but critical, spreadsheets. A team begins with one useful workbook. It contains the project list, a few formulas, and some notes. Then someone needs a special column. Another person adds color coding. A third employee duplicates the file for a new project because changing the original feels risky.

None of those choices is irrational. Each one solves an immediate need. Over time, however, the organization accumulates copies with different formulas, categories, and assumptions. There is no central database, no dependable version, and no practical way to analyze all projects together. The tool that helped each employee adapt has prevented the company from learning at scale.

But when leaders propose a structured system, employees often compare it with everything their spreadsheet can do. They point to the note they can place in any cell, the color that means something to their team, or the one exception column needed for a difficult client. The new system appears less capable because it does not reproduce every personal freedom.

But the old freedom was not free. Someone has to combine the files. Someone has to explain the colors. Someone has to discover that a copied formula broke six months ago. The company cannot improve a process it cannot see, and it cannot trust analysis built from records that were never meant to work together. Tools that give employees important flexibility to handle edge cases, can instead creep in to support a core, cross-team process in ways that are institutionally irreversible.

AI adoption is following the same path. Personal prompts become personal procedures. Private agents accumulate context no one else can inspect. Employees create their own categories, formats, and standards for what counts as a finished answer. Each setup may work well for its owner while making the organization less able to train, compare, govern, or improve the work.

This style of adoption comes with the same risk as the spreadsheet if the process of using the tool doesn't start with the necessary structure in place, it will be harder to reverse once employees depend on them. Except that in the case of AI, the tool will be capable of acting across more of the business – the reach of conflicting processes across employees will be seen at a scale not imaginable by current management practices. It is far cheaper to define the shared process and acceptable boundaries before that freedom becomes the default.

Exceptions still matter. A shared system should preserve notes, unusual cases, corrections, approvals, and human judgment. Constrained AI should make those exceptions visible rather than forcing employees to ignore them. In practice, the constraints are specific agent instructions, defined read and write permissions, authoritative sources of truth, and required templates or structured formats for acceptable outputs.

I 7. An agent without boundaries has authority without a job description

The word 'agent' can make a system sound more capable than it is defined. Leaders hear that an AI agent can search files, use software, prepare work, and take actions. Employees hear that they can hand off an annoying task. What often remains unclear is the job itself.

A bounded agent has a specific responsibility. Its instructions identify the authoritative sources it may use, its read and write permissions, the template or structured format it must produce, the quality checks it must perform, when a human must approve the next step, and what evidence should be preserved. Two people can run the same process and expect to get a reasonably consistent result.

An unbounded agent receives whatever objective an employee writes in the moment. One employee tells it to favor concision. Another asks for completeness. A third gives it access to a different set of files. The output varies because the job varies, even though the organization believes it has adopted one technology.

Security is another obvious concern for unbounded agents, but the operating risk is just as important. An agent may create records, change files, communicate with customers, or make recommendations before the company has agreed on the underlying process. It can turn one employee's preference into an automated behavior that other people now have to work around.

The faster the agent works, the less time the organization has to notice that its instructions conflict. A poorly defined manual task creates inconsistent work one case at a time. A poorly defined agent can create it across hundreds of cases before the first review meeting.

Imagine an agent that reschedules customer appointments. The task sounds narrow until a high-value customer has a special service agreement, two calendars disagree, or moving one appointment changes a technician's travel route for the rest of the day. A person familiar with the operation sees a related set of decisions. An undefined agent just sees an available time slot. The set of rules explaining how to evaluate multiple different input sources and how to balance competing priorities is some of the most valuable part of people's work – to trust an agent to intuit those answers without the same level of instruction and training you'd give to a human is no better than going to a roulette table and 'putting it all on green'.

Leaders need to define the task, information, quality standard, approval points, correction path, and stopping conditions before an agent receives lasting authority. That is not bureaucracy around AI - it is the same management discipline the company would apply before giving a person access to important systems and the freedom to act on its behalf.

I 8. AI slop does not save time. It transfers work upward

Generative AI produces finished-looking work at a speed that organizations have never had to absorb. An employee can create a report, presentation, policy draft, meeting summary, or market analysis before the person receiving it has time to ask whether the document should exist.

The output is often competent – not high-quality, but good enough. That is precisely why it is dangerous. It may be accurate enough to avoid immediate criticism, polished enough to look complete, and average enough that no single flaw appears worth sending back. Yet it often lacks the context, judgment, or attention to detail that an experienced employee would normally bring.

The employee who generated it experiences a large productivity gain. The reviewer experiences a new job. Facts must be checked. Assumptions have to be uncovered. The useful points must be separated from filler. The recipient has to reconstruct enough of the reasoning to decide whether the conclusion can be trusted.

The work did not disappear. It moved downstream, usually toward a manager or subject-matter expert whose time is more constrained and more expensive. The person closest to the generation step saved thirty minutes. Three people later in the process each spent twenty minutes deciding what was missing. The company recorded one faster task and missed the larger loss.

Volume makes the problem worse. When producing a document was expensive, employees had a reason to decide whether it was necessary and to improve it before sending it. When production is nearly free, the organization can create more material than it has the capacity or judgment to review. Important work competes with plausible noise. Managers either become bottlenecks or they lower the amount of scrutiny they apply.

For strong businesses, there is another cost. While AI, almost by statistical definition, regularly produces average-quality responses, experienced employees typically produce work that is meaningfully better than average because they understand the customer, the company's voice, and the subtle reasons a standard answer does not suffice. If AI increases volume while pulling quality toward the middle, the company can become less distinctive at the same time it appears more productive.

Research on the 'jagged technological frontier' helps explain the uneven result. The same AI can improve speed and quality on one task and reduce performance on another task that looks similar. The dividing line is not obvious enough to manage through trust or enthusiasm alone.

This changes the employee's core responsibility in an organization: generating the first draft is no longer the main contribution in many knowledge roles. The valuable work is choosing the right inputs, testing the reasoning, recognizing when the result falls outside AI's strengths, and refusing to pass a merely acceptable burden to the next person.

I 9. Do you actually even need that document anymore?

Historically, a lot of office work was designed around a physical limitation: one person knew something that the next person needed to know, so the first person assembled the context into a document. The document was not always the main objective, it was just a container that helped information survive the handoff.

Consider a monthly operations report. An analyst exports information from several systems, fixes inconsistent labels, calculates totals in Excel, writes a narrative, and emails a PDF to a manager. The manager reads ten pages to find three unusual results and decide which one needs attention. Almost every step exists because the systems cannot deliver the relevant information in a form the next person can use.

A superficial AI implementation produces the same report faster. The company celebrates time saved while preserving the exports, repairs, duplicate calculations, PDF, email, and manual search for exceptions.

Most organizations define efficiency as getting each existing step done faster. That directs investment toward the spaces between current checkpoints: prepare the report faster, send the recap sooner, update the tracker automatically. The process moves more quickly, but it still contains every handoff and review that existed before.

But if you step back, it becomes clear that many of those checkpoints exist only because information once had to move through people in sequence. One employee collected the data, another reformatted it, another summarized it, and another routed it to the person who could decide. Properly structured agents can now do much of that collating work directly, moving information from its source to the required end state while preserving the underlying record and flagging exceptions.

That can change the shape of the workflow from an assembly line to a hub-and-spoke system. The automated center gathers, reconciles, and routes information. Subject-matter experts focus on quality assurance and unusual cases. Employees who face customers, vendors, or partners spend more time interpreting the output and having the human conversation that follows. The opportunity is not to accelerate every existing checkpoint. It is to ask which checkpoints still protect judgment or relationships, and which existed only to move information from one person to the next.

Leadership is evolving away from management and towards coordination

Technology is evolving the role of management – a position that used to be about delegation and follow-ups is now far more centered on bi-directional communication and intra-team alignment. New technology gives employees more ways to diverge, which makes a clear vision, shared standards, and disciplined communication more important than before.

10. Leadership now means enforcing a shared way of working

Leadership has always required coordination. A company cannot deliver a dependable customer experience if every employee decides what quality means, which information matters, and how a promise should be fulfilled. AI has not created that responsibility, but it has made avoiding it much more costly.

In the past, changing a process often required a new system, a consultant, or help from IT. That friction kept most employees inside the established method, even when the method was imperfect. Now an employee can use AI to redesign part of the job alone. The new approach may be faster, more thoughtful, and completely incompatible with the way the rest of the team works.

This places leaders in an uncomfortable position. They want initiative, but they also need conformity. They want employees to improve the work, but customers should not receive a different level of care based on which employee discovered which tool. They want experimentation, but the business cannot support a permanent collection of experiments.

Conformity has a negative reputation because it can protect weak practices and suppress useful judgment. In an operating system, however, some conformity is what makes trust possible. A customer expects the company to remember the same commitments. Finance expects the same event to be recorded the same way. A new employee needs a process that can be learned without inheriting one person's habits.

In a world where AI can automatically handle grunt tasks that used to take employees days, the leader's job is less about assigning every task and more about deciding where variation is valuable or destructive. Employees may choose different ways to explore an idea, draft an early concept, or investigate a problem. They should not invent different definitions of a completed order, a qualified lead, an approved expense, or an acceptable client deliverable.

AI also changes the direction of communication. Employees can now solve more problems without waiting for instructions, which means leaders need regular ways to hear what they are learning from customers, vendors, and the tools themselves. That input should influence the shared process. Modern management, therefore, transitions away from a paradigm where the manager is telling the employee the vision and instructions to follow, to one where a primary job function is to become a listener for valuable information employees can surface to them. Like a spider at the center of a web, waiting for the faintest vibration to signal what type of action to take next, leadership needs to rely on employees to be filterers and conveyers of information and vision rather than implementors for limited tasks.

A strong leader turns useful experiments and employee feedback into team capability. They actively review and retires inferior methods, and explain why one standard was chosen over another. Without that work, the organization becomes harder to manage, harder to train, and less predictable for the people it serves.

11. Adoption fails when communication ends at launch

Companies spend millions on a new piece of technology, only to attempt to change years of employee behavior with a simple announcement email. The rollout includes an FAQ, two or three group training sessions, and a vendor representative who will answer questions for the first two weeks. Then the project team moves on.

From the implementation team's perspective, the system is live. From an employee's perspective, a familiar way of working has been removed and replaced with something that is slower precisely because it is unfamiliar. The employee still has deadlines, customers, and a backlog. Returning to the old spreadsheet or sending a private email feels like the responsible way to keep work moving.

This is why adoption cannot be treated as a communications task that happens at launch. People need to understand what problem the company is solving and why the disruption is worth it. They need to hear that explanation before decisions feel final, have a chance to provide real feedback, and test real cases that will reveal problems no demonstration could show.

They also need evidence that leadership is paying attention. If employees report a broken handoff and receive no response, the message is clear even if no one says it aloud. The new process is mandatory, but the pain it creates is theirs to absorb. Employees will either create workarounds or stop reporting what they see.

Weak communication produces two out-of-sync organizations. Some people will follow the new process when it is convenient or visible. The rest will continue through old files, private messages, and side conversations. Data splits. Managers cannot tell which record is current. New employees learn both methods. The company may now spend more time coordinating work than it did before the investment.

That is how a technology change becomes value destructive. The software cost remains, the old labor remains, and the disruption adds another layer of effort. Leadership may conclude that employees resisted change or that the platform failed. The more accurate conclusion is that the company never invested in a robust adoption plan.

A serious communication plan begins early, repeats the reason for the change, shares progress and setbacks, gives employees a clear place to surface exceptions, and responds quickly when the first cases fail. Managers observe actual use instead of relying on training attendance. Wins are made visible so the organization can feel progress. The old process receives an end date. Communication is not decoration around implementation. It is part of the infrastructure that makes implementation real.

A lazy rollout can waste the technology investment and leave the organization worse than it was before the purchase.

12. Employee feedback is essential, conflicting, and not democratic

Employees know things that no executive team can discover from a process map. They know the question customers always ask after receiving the standard email. They know which field is impossible to complete at the time the system requires it. They know which exception appears once a week and which one appears once a year but carries enormous risk.

A transformation designed without that knowledge will fail in everyday use. The formal process will look clean because the messy cases were never represented. Employees will repair the difference through workarounds, and leadership will lose visibility into the real process again. Employee feedback is essential to any effective process change or technology rollout.

Sometimes leaders give lip-service to employee input, only to collect feedback for appearance and ignore it once the launch date approaches. Employees learn that the listening sessions were ceremonial. The next time leadership asks for input, people either stay quiet or arrive ready to defend their existing process.

However, while gathering feedback is essential, that does not mean it should always be strictly heeded. Sales may want speed and freedom to adapt. Finance may want stronger controls and complete records. Operations may want one repeatable method. A tenured employee may defend an exception because it helped one important customer. A newer employee may want to remove it because every routine case is now harder.

All of those perspectives can be legitimate. They can also be impossible to satisfy at the same time.

The actual work of an executive is balancing those interests and deciding who will lose something in the transition. A cleaner process may reduce one team's flexibility. Better controls may slow another team's work. A shared source of truth may take ownership away from the employee who maintained the old spreadsheet. Real transformation creates tradeoffs.

A robust system for incorporating feedback should separate the observed problem from the employee's preferred solution. It should record how often the reported issue occurs, who is affected, what risk it creates, and which business priority is in conflict. Leadership can then decide which needs to honor, which requests to defer, and which practices to overrule.

Overruling a preference is not a failure of collaboration - the final decision will disappoint someone. Leadership's job is not to avoid that outcome; it is to make the tradeoff deliberately and explain it plainly. Employees are more likely to accept a standard they did not choose when they can see the tradeoff, the evidence, and the reason the company needs one method rather than several.

13. Different employees need different support, not different standards

A group training session creates the impression that everyone received the same opportunity to learn. It does not mean that everyone is ready to use the new process.

One employee understands after seeing the complete workflow once. Another needs written instructions that can be revisited without asking for help. Someone who knows the current system deeply may struggle because the new design removes shortcuts that once made that person effective. A newer employee may adapt quickly because there is less to unlearn. A manager may resist a workflow that appears to reduce control, while a frontline employee welcomes the same change because it removes a daily frustration.

Age and generation often influence how employees approach technology. Role, confidence, learning style, risk exposure, prior experience, and the amount of change to an employee's identity and a host of other considerations impact the type of support needed in order to get teams to adopt organizational and technological change.

When training assumes one type of learner, non-adoption still happens behind the scenes. Employees attend the session, understand the demonstration, and still cannot complete a real case. They avoid admitting the gap because everyone else appears to have moved on. They continue to use the old method because it feels more dependable than a new system they don't understand. Managers discover the problem only after data is missing or a customer receives the wrong result.

The organization then faces a false choice. It can allow different employees to use different processes, or it can label struggling employees as resistant. Both choices confuse the standard with the support needed to reach it.

The path to competence can vary. Some employees need guided practice on real work. Some need short follow-up sessions, direct observation, peer support, or a written reference. Everyone needs a safe way to report a mistake before it becomes a hidden workaround. A fair rollout gives each employee a realistic way to reach the same operating standard and gives managers enough visibility to know whether that has happened.

14. People need to know that someone cared - AI can't replicate that

People can accept answers they do not like. What is harder to accept is the sense that nobody understood the question, weighed their circumstances, or cared what happened to them. That distinction can sound emotional and difficult to measure. But in practice, it changes whether employees adopt a process, whether customers accept being turned down, and whether anyone continues to trust an organization.

Businesses often miss this because they judge a process by its measurable result. The new workflow launched. The policy was applied. The customer received an answer. But each decision produces another outcome that affects what the person believes about the organization, and what they might do in response. They may comply, call again, escalate, build a workaround, or leave. Those reactions are business consequences of the original process, even when they appear later.

The first place this problem often appears is inside the company, when leaders ask employees to change how they work. AI is harder to introduce than ordinary software because employees do not encounter it as a neutral tool. Their opinions about AI often reach beyond the workplace and into beliefs about labor, politics, privacy, religion, education, and personal identity. An employee can welcome AI for one task and object to it in another. A rollout of new AI tools or processes can be perfectly executed technically, and still be utterly unworkable socially.

Those broader concerns arrive alongside a practical disruption. A process an employee spent years learning may disappear. Judgment that once made them valuable may become reduced to a single line in a prompt. A familiar routine may be replaced with a system they do not understand or trust. The employee is not evaluating only whether the new workflow is better. They may also be asking what the change says about the value of their experience, the security of their role, or the kind of work the company expects people to do.

Showing care is how a manager makes that complexity discussable. It does not require the manager to agree with every concern or account for every preference. It means asking what the employee believes will be lost, distinguishing operating risks from personal concerns, explaining which tradeoffs leadership has chosen, and providing support for the parts of the change that are genuinely difficult. That conversation may not make the employee like the decision, but it shows that a person with authority understood the concern before making it. It also gives leadership information the rollout plan may have missed. Without that exchange, employees are more likely to comply during training while silently fuming or ignoring the new procedures in private.

The same human need appears in customer service or vendor management. Customers rarely call because the ordinary process worked. They call because a charge looks wrong, a delivery does not match the record, or their circumstances do not fit the available options. An automated system sends them through the same menu, transfers them to another number, or repeats a standard answer to a nonstandard problem. They go around and around while screaming "speak to a representative!" into the phone.

By contrast, a capable customer service employee can recognize that a situation is unusual and needs some form of exception. That person can call another department, ask a manager whether an exception is possible, investigate a system error, or look for an option the standard script does not cover. The final answer may still be no. But saying, "I checked A, B, C, and I am sorry, but there is nothing else we can do" produces a different result from an automated denial. The request was rejected, but the customer knows that someone understood the problem and tried to help - and that sole difference is what determines whether the interaction was a success or failure.

An automated system may eventually reproduce the same investigative steps and say the same words. It may reach exactly the same decision. But no matter how procedurally identical, no automated process can give the customer the *feeling* of being seen and cared for. In order to feel cared for, an action requires effort - requires cost - and an automation will, definitionally, always represent an attempt to minimize cost, and therefore care.

This does not mean every interaction requires a person. Routine requests may justify complete automation. Leaders do, however, need to protect the moments when human recognition is part of the value being delivered - both internally and externally. This role is a critical piece of the evolving responsibility of modern leaders. Significant changes to an employee's work, disputed decisions, or unusual customer exceptions need a route to someone with enough context, authority, and empathetic bandwidth to respond.

Leaders cannot judge processes only by whether they reached the right answer quickly. A fast and technically correct process has not succeeded if it causes resistance, repeat work, or destroys trust. In consequential moments, 'human care' is part of operational performance.

Faster execution is counterproductive to success

The point of automation is not to remove people from work that deserves human responsibility. It is to remove the administrative burden that keeps people from applying judgment, building trust, and improving the system.

I 15. Some work should stay slow

Most transformation plans identify what should become faster. Few identify what should remain careful and deliberate, or intrinsically requires a human touch.

That bias matters because speed is not a neutral improvement. A faster approval can help a customer, but anyone who's sat on a customer service call knows that an artificially faster conversation can feel dismissive. A faster recommendation can reduce analysis. A faster escalation can prevent the employee closest to the problem from using judgment. The value of speed depends on what the step is trying to accomplish.

Software is good at moving information, checking required fields, applying defined rules, preparing routine records, and making sure the next person knows that work is ready. People are comparatively expensive at those tasks. In contrast, people are far better at noticing discomfort, earning trust, persuading someone who is uncertain, understanding an unusual context, and deciding when the formal rule does not fit.

A company should use each form of infrastructure for what it does well. That sounds obvious, but many automation programs measure success through the amount of labor removed rather than the quality of human attention created. The process becomes faster on paper while employees gain no additional time for customers, coaching, analysis, or improvement.

Sometimes the saved capacity is simply filled with more administration. A manager receives an automated report and is asked to attend another status meeting. A sales representative saves time on data entry and is assigned a larger volume of low-quality outreach. An event professional gets a faster schedule but no more time to notice what would make the experience memorable.

At that point, the business has automated the surface-level task without changing what employees are able to contribute. It may even make the customer experience worse by replacing a thoughtful moment with a technically correct response that arrives sooner.

The important part of automation is not identifying where a person can be removed, it's about what valuable human behavior the company wants to buy with the time automation creates.

Automation should remove the work that prevents people from being useful, not the moments when people are most useful.

16. AI turns ordinary mistakes into large, fast decisions

People and software both make mistakes. The distinctive risk of AI is not that it can be wrong. It is that a plausible answer can move from question to action with almost no friction, and the scale of the action may be much larger than the evidence behind it.

A reported case involving a 67-year-old sesame farmer in China shows how trust can accumulate. The farmer began skeptically, used an AI assistant for routine agricultural questions, and received useful advice over many months. Each successful answer made the next answer easier to trust. When weeds and pests threatened the crop, he reportedly followed a pesticide recommendation across almost 25 acres without consulting an agricultural technician or independently checking how the herbicide should be used.

The treatment killed the weeds and the sesame crop. One product was intended for more limited use and affected a plant biologically similar to the crop itself. The failure wasn't as simple as hallucinated answer: it was a chain in which prior success reduced verification, confidence exceeded context, and one recommendation was given authority over a large, difficult-to-reverse action.

This pattern appears in business with less dramatic symptoms. An AI system prepares a customer message, changes hundreds of records, applies a classification rule, or recommends a financial action. The first cases work. Review becomes lighter. Then an unusual case reaches the same automated path, and the error is distributed before anyone recognizes that the context changed.

Hiroki Tomiyasu offers a useful contrast. A former public servant who became a farmer in Hokkaido, Japan, he has used ChatGPT and Codex to learn farming techniques, connect sensors, build a group-chat bot, and remotely control greenhouse vents. AI expands what he can understand and build even though he began without farming or software experience.

The important difference is not that his systems never fail. He uses simple controls, protection against accidental use, explicit safety considerations, and additional verification with human experts when a decision could affect the farm. AI helps him gather information and create tools. It does not receive responsibility for consequences it cannot understand.

This is the authority question every organization has to answer. AI can recommend, draft, classify, compare, and flag. Whether it should act depends on the cost of an error, the strength of the evidence, the ability to reverse the action, and the clarity of human responsibility. Low-risk work may justify broad automation. High-consequence actions need narrower permissions, explicit approval, and a record that makes review and correction possible.

A helpful reminder from the past still rings true today: in 1979, an IBM training manual stated that “a computer can never be held accountable, therefore a computer must never make a management decision.” That principle matters even more now, as AI moves from producing answers to taking action. Execution has never been the hardest part of running a business; there is a reason routine execution has traditionally been assigned to a business' most junior employees. The harder work is judgment: deciding whether the output is right, whether the risk is acceptable, and who will answer for the result. That responsibility has always belonged to an organization's most experienced and capable people. AI can accelerate execution. It cannot assume accountability.

Reliability is not enough. Authority also depends on reversibility and accountability.

17. Transformation is now a permanent operating discipline

AI is accelerating change across the wider technology landscape. Alongside advances in AI itself, its use in software development is changing how quickly other tools can be built and improved. Vendors add features, change prices, and alter what their products can do. Employees discover new methods before leadership has evaluated the old ones. Custom-built tools are becoming cheaper and more practical, opening up options that were difficult to justify even a year ago. Security creates another reason to keep adapting. A system can still perform its intended job and require urgent changes because a new vulnerability or threat has emerged.

Leaders are already accustomed to managing a succession of projects. What is changing is how much those projects overlap. A launch date traditionally allowed for a period of stabilization, training, and adoption before the next major change. Now, the next change may need to begin the day after launch. The assumptions behind an implementation can shift while employees are still learning how to use it.

Managing that pace requires a more conversational approach to change. Training materials need to stay current, and employees need clear explanations of what is changing and why. But their involvement needs to begin well before those instructions arrive. Employees should help shape changes during design and testing, with opportunities to influence further improvements once they begin using the tools. Leadership cannot grant every request. What matters is that people can see where their feedback made a difference. When a problem they raised gets fixed or a capability they requested becomes available, they have a reason to participate in the next change.

Small, targeted rollouts can help that participation spread. A team that receives a useful tool it asked for and helped design has something concrete to share with colleagues. These "moments of delight" give other teams a chance to see the benefits through people they know, doing work they recognize. Teams that had little interest in the project may start asking how they can get involved or suggesting improvements of their own. Frequent change creates more opportunities to build that enthusiasm. Employees can become advocates for adoption because they have experienced how helping shape a change makes their work easier.

Budgets also need to accommodate technological instability. When funding is tied to a fixed project scope, a useful improvement or necessary security change can require a new approval process before work can begin. Teams face pressure to deliver the original plan even when a better option has become available. By contrast, a flexible annual budget for ongoing technology improvement gives leaders room to redirect spending as needs and capabilities change. That flexibility needs clear spending limits and accountability, but it allows project scope to evolve without reopening the entire investment decision each time. Leaders need to control the total investment while allowing the work itself to change.

The companies that thrive will plan for continual adjustment, with the flexibility to make many small changes instead of relying on occasional transformations with fixed scopes and end dates.

Selected sources

Harvard Business School. Navigating the Jagged Technological Frontier: field evidence on when AI improves or worsens knowledge-work performance. [Open source](#)

NIST. AI Risk Management Framework guidance on human roles, oversight, and risk-proportionate controls. [Open source](#)

Tom's Hardware. Reported case of AI-generated herbicide guidance destroying a sesame crop in China. [Open source](#)

Hiroki Tomiyasu. First-person account of using Codex with sensors, motors, and LINE to build practical farm tools. [Open source](#)

ChatGPT Pro Community. Profile of Tomiyasu's use of AI for farming, troubleshooting, and automation. [Open source](#)

IBM. AI decision-making: Where do businesses draw the line? Reproduces the 1979 IBM training-manual quotation on accountability. [Open source](#)