FROM THE LEADERSHIP ARCHIVE

Staying Agile

Learn how the agile software development process is being used as a management strategy, with lessons from ING bank, Toyota, Spotify, and others.

CONTENTS

Staying Agile

INTRODUCTION

Agile practices emphasize process flexibility and the quick delivery of value. This collection of articles from *MIT Sloan Management Review* looks at how the agile software development process is being used as a management strategy, with lessons from ING bank, Toyota, Spotify, and others.

From "Consider Culture When Implementing Agile Practices":

- Agile practices help organizations bring products and services to market quickly and enable them to adapt nimbly to changes in the technology landscape.

- A challenge to implementing agile practices globally is accommodating cultural differences. For example, openly expressing thoughts and opinions to authority figures and publicly discussing successes and failures are accepted practices in agile teams, but they may not be consistent with cultural norms in all parts of the world.

- Understanding the unique words used to describe particular behaviors, or the "cultural script" around an action, is an important first step when implementing agile projects outside the U.S.

- To deal with areas where agile practices interact with cultural scripts in atypical ways, companies should maintain flexibility and speed, build an effective agile team, and create accessible communication channels.

From "What to Expect From Agile":

- Agile emerged during the 1990s as a software development methodology, gathering momentum as a more responsive and collaborative approach to development than the traditional "waterfall" methodology.

- In recent years, agile has moved into mainstream thinking as a management practice.

- A detailed case study looked at the operations of ING bank in the Netherlands, which has adopted agile throughout its headquarters in Amsterdam.

- Five lessons emerged from interviews with 15 executives of the internet banking company. Among them: For agile to work, executives need to figure out how much power they were willing to give up. They also have to give employees the right balance of oversight and autonomy.

From "A New Approach to Designing Work":

- In applying agile methods from the field of software development to other domains, managers often confuse practices and principles.

- When agile methods work for software development, they do so because the associated practices manifest key behavioral principles — in the context of software development.

- The key to transferring a set of practices from one domain to another is to first understand why they work and then to modify them in ways that both match the new context and preserve the underlying principles.

- By using a dynamic work design, organizations can create their own work processes that are both more flexible and more efficient. An example from a Toyota supplier shows how.

From "Goodbye Structure; Hello Accountability":

- Well-designed systems can help make a company more agile. But even great processes are responsive to change only if the people who use them recognize what needs to be done and how to do it.

- Organizational structures are designed to clarify how a company will meet stated objectives. They're not necessarily good at helping people adapt to changing objectives.

- Managers will be able to operate as true digital leaders only when they shake their reliance on structure as the primary tool of organizational design. They instead need to assign accountabilities in ways that instigate focused responses to opportunities.

- For instance, the Swedish streaming entertainment company Spotify supports its customer offerings through the efforts of small autonomous teams. Called squads, these teams define their own missions and develop their own goals, testing and adjusting as they go along.

From "Architect Your Company for Agility":

- Great strategy is valuable only if a company is capable of executing that strategy, and whether a company can execute its strategy depends on whether it is designed to do so.

- What matters is business architecture — the way a company's people, processes, systems, and data interact to deliver goods and services to customers.

- To keep pace with customer demand and competitor activity, companies must be able to quickly experiment with a potential offering and, depending on customer response, enrich and scale that offering, or discard it and move on to the next experiment.

- To that end, growing numbers of companies are creating small, cross-functional, agile teams. The concept of continuous release is essential to the effectiveness of these empowered teams. Critically, these teams don't fear failure.

From "What the Military Can Teach Organizations About Agility":

- U.S. armed services are experimenting with ways to make faster — and smarter — decisions, and business should take note.

- Once bastions of command-and-control management style, modern military institutions such as the U.S. Marine Corps are at the forefront of thinking about organizational agility. That has been essential in an era of digitally enabled terrorism.

- Military thinking now emphasizes the kind of innovation necessary to move leaders efficiently through the four decision cycles of Observe, Orient, Decide, and Act in order to ensure that forces are best able to respond quickly to sudden external threats.

- For instance, military leaders focus only on decisions they can make, letting teams closest to the problems make decisions and take action decisively and independently.

Consider Culture When Implementing Agile Practices

BALASUBRAMANIAM RAMESH, LAN CAO, JONGWOO KIM, KANNAN MOHAN, TABITHA L. JAMES

Teams can create balance between agile practices and local culture.

Agile practices help organizations bring products and services to market quickly and adapt nimbly to customer and market changes and innovations in the technology landscape. In today's globalized economy, agile methods pioneered in the United States are being adopted in organizations worldwide.

One challenge to implementing agile practices globally is accommodating cultural differences. Because the agile approach started in the United States, American cultural norms may play an outsized role in how agile methods are prescribed and carried out — and that could create problems when teams in other countries adopt agile methodologies. For example, openly expressing thoughts and opinions to authority figures, publicly discussing successes and failures, and assigning credit or blame are often accepted practices in agile teams, but such open interactions may not be consistent with cultural norms in all parts of the world.

Recognizing the unique cultural characteristics of employees taking part in agile projects outside the United States may be critical to project success. When agile practices clash with local culture, it's important for organizations to recognize the conflict and develop solutions sensitive to the societal norms without impeding agile practices.

We interviewed employees of eight software companies in China, India, and South Korea that had adopted agile software development practices to find an answer to this key question: How do the cultural scripts common in your country work with or against the tenets of agile methods? A population may have a unique word to describe particular behaviors; that named phenomenon is the "cultural script." [1] We focused on a small set of cultural scripts in each of the three countries. (See "Three Countries, Many Cultural Scripts.")

The employees we talked to described situations in which complications arose because agile practices differed from behaviors that were considered acceptable according to cultural scripts. When faced with such discrepancies, companies developed culturally sensitive ways to encourage employees to adopt agile practices. In many instances, organizations were able to create harmony between being agile and abiding by the local culture's accepted behavioral norms. We've distilled their experiences into practical recommendations your organization can use to address the challenge of being both agile and culturally sensitive.

Three Countries, Many Cultural Scripts

A population may have unique words to describe particular behaviors, a phenomenon known as the "cultural script." The authors identified a small set of cultural scripts for three countries in which their data was collected.

CHINA	*Guanxi:* a reciprocal exchange relationship where one person's favor to another is eventually repaid. These exchanges build and strengthen relationships. [i]
	Mianzi: the desire to maintain social standing. Individuals perceive their standing in a social context and behave in ways that preserve it (that is, save face). Causing others to lose face in social situations is undesirable. [ii]
	Doctrine of the mean and harmony: a desire to maintain a harmonious social environment. It stresses avoidance of confrontation and the value of compromise. [iii]
SOUTH KOREA	*Palli palli:* a tendency to value quick and quality task completion. It emphasizes the importance of performing tasks quickly, accurately, and diligently — and doing so as efficiently as possible by eliminating unnecessary work. [iv]
	Jeong: the process of forming emotional bonds through repeated social interaction. It stresses the idea that people can forge strong social relationships through shared experiences, regardless of personal opinions of each other. [v]
INDIA	*Jugaad:* a practice of solving problems through improvisation. It originated in a milieu of societal constraints and scarce resources and therefore stresses using what is available to create solutions. [vi]
	Social hierarchy: a viewpoint that outlines a well-defined organizational structure, with clear leaders and subordinates, that drives decision-making. It may also give leaders the latitude to guide the work and careers of subordinates. [vii]
	Apane log: the concept of being part of the in-group. Being considered part of the in-group (or the out-group) can define one's relationships within an organization. [viii]

Lessons From Global Agile Implementations

We interviewed agile team members who held roles such as senior manager, project lead, developer, scrum master, agile coach, and customer representative. They worked on the development of a diverse collection of products, from customer relationship management systems, supply chain tools, and inventory management systems to game platforms and mobile applications. As we learned about the cultural challenges they faced in agile projects and how they dealt with them, we found three areas where agile practices interacted with cultural scripts in atypical ways and created challenges: maintaining flexibility and speed, building an effective agile team, and creating accessible communication channels. In many cases, teams can resolve those types of challenges with flexible solutions that preserve key agile practices but accommodate a culture's distinct norms.

Maintaining Flexibility and Speed

Agile practices emphasize process flexibility and the quick delivery of value. Some cultural scripts — such as *jugaad*, an Indian concept that stresses improvisation as a way to solve problems, and South Korea's *palli palli*, which emphasizes the importance of completing tasks quickly— align well with this focus, but they may also cause project teams to overreach. We uncovered two solutions for avoiding unmanageable situations in teams whose members may embrace either jugaad or palli palli:

streamlining improvisation and adjusting client expectations using a hybrid approach.

Streamline Improvisation

In India, one benefit of jugaad is that it empowers employees to work well in resource-limited environments on tight schedules. Both of those characteristics align well with the flexible and fast-paced nature of agile teams. Employees who can find ways to do more with less can help speed development and reduce waste. The downside is that unorthodox problem-solving approaches may be inconsistent with formal organizational processes, so they may not be reproducible and they may not have undergone the vetting necessary to ensure that they don't result in unexpected or negative consequences.

This trade-off is illustrated by an example from our interviews. A developer in India needed access to new mobile devices for developing and testing an application. As a solution, he arranged to borrow mobile devices from a retail store at night. This novel approach became an embarrassing problem for both companies when a customer of the retail store found out that her new phone had been used already. The incident caused the senior managers of the software development company to realize that clear organizational guidelines were needed to manage improvisation encouraged by jugaad. The two companies set up a formal arrangement whereby the retail store made mobile devices available to the software company in exchange for software development services. By streamlining improvisation encouraged by jugaad, organizations can leverage the ingenuity of improvised solutions and at the same time bring organizational legitimacy to them.

Adjust Expectations With a Hybrid Approach

South Koreans see palli palli as a dominant part of their culture. It emphasizes getting things done quickly, and that mindset is ideally suited to agile's principle of delivering value early in the development process. In fact, the two concepts are so well-aligned that some South Koreans refer to agile practices as palli palli practices. Employees in a culture that values speed will strive to deliver results quickly, but an overemphasis on quick results can encourage a continual tightening of already-aggressive schedules, create unrealistic client expectations, and orient work toward short-term results.

When agile practices were put in place in South Korea, employees felt significant pressure from clients to deliver products faster than before. To counter this pressure, some project teams have adopted a hybrid approach wherein a contract is drawn up and the project team and the client agree to terms that specify the overall scope and schedule for the project and identify a set of initial requirements. Contractually limiting a project's scope and timeline allows companies to implement agile practices only after major requirements are specified and better manage fast-delivery expectations. While such approaches go beyond what is advocated by agile methods, they are essential because they rein in the hyper-agility that would have been culturally expected otherwise. Hybrid solutions like that can help a team shift the baseline for agility so expectations are realistic.

Building an Effective Agile Team

Agile practices are people-oriented, and agile teams are self-organizing and cross-functional. Cultural scripts such as the doctrine of the mean and harmony (DOM) in China and India's principles of social hierarchy and *apane log* can contribute to team building, but they also emphasize communication patterns that could restrict the free flow of information between team members. We recommend avoiding such issues by adding pragmatic structure to teams while being sensitive to cultural impacts on team cohesiveness.

Add Pragmatic Structure

The hierarchical structures encouraged by DOM in China and social hierarchy in India mean employees expect decisions to be made by superiors and are accustomed to following instructions. This dependence on structure is counterproductive in agile environments that rely on dynamic self-organization and empower employees to determine the best way to get their work done.

We found that software development companies in both China and India took steps to reorganize their teams in ways designed to discourage employees from falling back on traditional hierarchical structures to manage their projects. For example, in India, one company formed multiple subteams and each member of a subteam was designated the leader for a module. The leaders could recruit other members of the subteam to perform tasks for their modules. Similarly, in an agile project at a company in China, instead of one person acting as product owner, a group of stakeholders collectively served as product owner. In both examples, authority was dispersed across multiple individuals and reporting structures were rearranged, so the companies were able to encourage individual decision-making and communication between parties — practices that would have been discouraged in traditional structural hierarchies.

Balance Skill Sets and Group Cohesiveness

In India, the cultural script of apane log leads to the formation of teams made up of people with shared backgrounds and experiences. While such teams may be highly cohesive and therefore able to pursue projects with a sense of common purpose, they may lack diversity and members may demonstrate a tendency to prioritize team goals over the project.

We found that agile teams in India may lack the diversity of skills that is so important to the agile process because team members are often drawn from a pool of people with similar backgrounds. That's because staffs in India tend to be made up of people with similar education and training because social ties generally play a more important role than skill sets in decisions about who to hire and who to promote. The end result is an undesirable uniformity of skills on agile teams. Moreover, the strong social relationships typical among people who embrace the notion of apane log also led to team members covering for one another's weaknesses. To reduce the impact of apane log, managers adapted hiring practices to ensure that teams possessed the range of competencies and skill sets that were essential for particular projects.

Creating Accessible Communication Channels

Open and ongoing communication is critical to agile practices. But interpersonal communication is often deeply affected by cultural scripts that can inhibit open dialogue and interaction essential in agile teams. We suggest overcoming this obstacle by building an environment in which people can engage in open communication without running afoul of cultural norms that may discourage it.

Create Context for Openness

Open communication was a challenge for agile teams in all of the countries we studied. For example, in China, *mianzi*, which emphasizes the importance of maintaining social standing, caused employees to be reluctant to share information they believed might damage their own reputations or the reputations of others. And in South Korea and India, *jeong* and apane log, respectively, which both stress the importance of forging strong social relationships, can have both positive and negative repercussions in agile projects. On one hand, the group cohesiveness that results from such relationships can be beneficial, but it can also lead individuals to engage in behaviors that favor the team over the project.

Employees who embrace mianzi, jeong, and apane log may take criticism too personally or cover for the weaknesses of fellow team members. Those three cultural scripts inhibit the open communication that agile methods rely on to keep the development process transparent and ensure that project information is available to all members of the team.

Managers in China and South Korea dealt with their employees' hesitancy to express themselves openly by creating environments in which open dialogue was acceptable. In South Korea, one organization held informal meetings after regular work hours in bars or other locations outside of the office, creating an opportunity for team members to discuss difficult topics in a comfortable atmosphere. At some of those off-site meetings, project members informally briefed their managers and customers about the difficulties of adopting certain agile practices. In China, the managers' approach was to explain to employees how mianzi could be used to assist the agile process. They emphasized that mianzi is a

conflict management tool that could help employees avoid responding to one another in irrational ways. They also noted that embracing mianzi could help colleagues and customers "save face" by adopting a positive attitude when communicating with one another.

Create New Communication Channels

In China and India, DOM and the institution of social hierarchy, respectively, instill people with a sense of respect for authority figures and, as a result, interpersonal communication is often very formal and less open. This less relaxed style of interaction can hinder the free exchange of ideas and make it harder for employees lower in the organizational hierarchy to develop a deep understanding of customer needs.

In both countries, we found that employees would remain silent when they disagreed with their superiors, because they believed they were not in a position to make decisions. In China, only some employees would talk to customers, and in India, communication occurred mainly between people who were at the same level of the organizational hierarchy. On an agile team, a lack of open communication with the customer may lead to incorrect assumptions about customer needs, and therefore significant rework may be required when the end result is not what the customer expected. To address that problem, companies created wikis to support collaborative decision-making. Through the wiki, all team members could share their thoughts openly but also anonymously, so that they didn't appear disrespectful. Such solutions can help employees feel they are preserving their respect

for authority, and at the same time provide a channel for openly sharing concerns.

Consider Culture in an Agile Implementation

Introducing practices that bring agility to a team or organization can be done with cultural sensitivity. Successful agile companies adapt their practices to suit project and organizational characteristics. [2] Adapting your agile practices to be considerate of cultural scripts will demonstrate to global teams that agile methods are not a one-size-fits-all approach and that localized nuances *do* matter. As you implement agile practices pioneered in one culture in countries that have different cultural norms, it's important to consider how the requisite agile behaviors may mesh with the cultural norms of the employees being asked to behave that way.

While the specific solutions you create to resolve conflicts between agile practices and cultural scripts will likely vary depending on local contexts and cultural norms at play, the broader and more compelling storyline here is that agile product development teams can find a workable balance between standard agile practices and behaviors that would be typical in the local culture. Striking such a balance is important not only because it helps ensure that agile teams will be able to accomplish their goals, but also because it can create interesting opportunities to leverage the unique perspectives and solutions different cultures may offer.

About the Authors

Balasubramaniam Ramesh is Distinguished University Professor and chair of the Computer Information Systems Department in the J. Mack Robinson College of Business at Georgia State University. Lan Cao is a professor of information technology in the Strome College of Business at Old Dominion University. Jongwoo Kim is an associate professor of management science and information systems in the College of Management at the University of Massachusetts, Boston. Kannan Mohan is a professor of information systems at the Zicklin School of Business, Baruch College. Tabitha L. James is an associate professor of business information technology in the Pamplin College of Business at Virginia Tech.

References

1. H.C. Triandis, G. Marin, J. Lisansky, and H. Betancourt, "Simpatía as a Cultural Script of Hispanics," Journal of Personality and Social Psychology 47, no. 6 (1984): 1363-75.

2. D.K. Rigby, J. Sutherland, and H. Takeuchi, "Embracing Agile," Harvard Business Review 94, no. 5 (2016): 40-50.

i. C. Xiaojuan Ou, P.A. Pavlou, and R.M. Davison, "Swift Guanxi in Online Marketplaces: The Role of Computer-Mediated Communication Technologies," MIS Quarterly 38, no. 1 (2014): 209-30.

ii.   S. Ho Park and Y. Luo, "Guanxi and Organizational Dynamics: Organizational Networking in Chinese Firms," Strategic Management Journal 27, no. 5 (2001): 455-77.

iii. D. Tjosvold and H. Sun, "Understanding Conflict Avoidance: Relationship, Motivations, Actions, and Consequences," International Journal of Conflict Management 13, no. 2 (2002): 142-64.

iv. C. Rowley and J. Bae, "Human Resource Management in South Korea After the Asian Financial Crisis: Emerging Patterns From the Labyrinth," International Studies of Management and Organization 34, no. 1 (2004): 52-82.

v. I. Yang, "Jeong Exchange and Collective Leadership in Korean Organizations," Asia Pacific Journal of Management 23, no. 3 (2006): 283-98.

vi. M.G. Singh, A. Gambhir, and J. Dasgupta, "Innovation in India: Affordable Innovations," The Global Innovation Index (2011): 77-86.

vii.   A. Ardichvili, D. Jondle, B. Kowske, E. Cornachione, J. Li, and T. Thakadipuram, "Ethical Cultures in Large Business Organizations in Brazil, Russia, India, and China," Journal of Business Ethics 105, no. 4 (2012): 415-28.

viii.   A. Varma, E.S. Srinivas, and L.K. Stroh, "A Comparative Study of the Impact of Leader-Member Exchange in US and Indian Samples," Cross Cultural Management 12, no. 1 (2005): 84-95.

This document is authorized for use only in Prof. V. S. Prakash Attili's MDP/207/21/EPITM/508 at Indian Institute of Management - Lucknow from Nov 2021 to Jan 2022.

SPECIAL COLLECTION • "STAYING AGILE" • **MIT** SLOAN MANAGEMENT REVIEW   7

What to Expect From Agile

What happens when a company whose roots go back over a century — a bank, no less — decides to adopt agile management methods developed in the software industry?

BY JULIAN BIRKINSHAW

? **THE LEADING QUESTION**
What happens when an established company decides to adopt agile management methods?

FINDINGS

▶ Executives need to be willing to share power and embrace new ways of working.

▶ It's important to pre-pare stakeholders for the change.

▶ Build the organiza-tional structure around customers — and keep it fluid.

HOW IS TECHNOLOGY transforming the practice of management? As everyone knows, tech-nological innovation *enables* changes in how we work, for example, helping people collaborate, access information more quickly, and make smarter decisions. Less obvious, but no less important, is the observation that technological innovation *inspires* new approaches to management. For example, the shift from mainframe computers to personal computers gave impetus to the empowerment trend of the 1980s, and the emergence of collaborative software tools shaped the knowledge management movement in the 1990s. In such cases, new technologies expand our capabilities *and* broaden our horizons, and it is this combination that enables management to evolve.

A case in point is agile. This emerged during the 1990s as a software methodology, made possible by new programming languages that made it much easier for developers to build prototypes and gain rapid user feedback. The concept of agile software development was formally defined in 2001,[1] and over the next decade it gathered momentum as a more responsive and collaborative approach to software development than the traditional "waterfall" methodology.[2] In recent years, agile has started to move into mainstream management thinking, with some observers proclaiming it the

next big thing. Forbes.com contributor Steve Denning calls it a "vast global movement that is transforming the world of work." In a 2016 *Harvard Business Review* article, Darrell Rigby, Jeff Sutherland, and Hirotaka Takeuchi wrote that "agile innovation has revolutionized the software industry. … Now it is poised to transform nearly every other function in every industry."[3]

The purpose of this article is to shed light on agile as a management practice. To do this, I report on a detailed case study of the operations of ING bank in the Netherlands, which has adopted agile across its headquarters in Amsterdam. Though ING's Dutch operations are less than three years into the process — and it's therefore premature to declare the initiative a success — taking a deep dive into the organization's early experience with adopting agile is nonetheless instructive.

Most IT departments in large companies today are adopting agile techniques to some extent, al-though with varying degrees of success.[4] And many fast-growing technology companies, such as London-based Spotify Ltd. and Los Angeles-based

Riot Games Inc., have embraced agile not just as an IT methodology but as a way of working.[5] By contrast, ING is a bank whose roots go back more than a century. It is the first case I know of in which an established company in a traditional industry is reinventing its management model throughout its operations in a particular country — not just its IT or software development management model — using agile principles. By studying the experience of ING's operation in the Netherlands, leaders at other established companies should be able to make more informed decisions about whether pursuing agile is right for them.

In this article, I highlight key learnings at ING in the Netherlands, largely from the point of view of the senior executives of the bank during this transition period, and I reflect on some of the broader implications. I don't spend much time on the internal workings of the agile teams, or *squads* as they are known within ING, because others have written extensively on those.[6] Instead, my focus is on implementing agile on an organization-wide basis. This is where the ING experience is unique — and hopefully most useful to other established companies that are seeking to embrace agile working.

My research is based on in-depth interviews with 15 ING executives and many front-line employees. (See "About the Research.") In addition, I spoke to leaders tackling similar issues about new ways of working at other large companies, including Barclays, Roche, Bayer, Unilever, and BMW. Tellingly, one of the ING leaders I interviewed, Bart Schlatmann, left ING early in 2017 when another large global bank recruited him to help them implement agile methods. (He had spent 22 years with ING, the last 10 as COO of ING Netherlands.)

Why ING Adopted Agile

ING has always been open to new ways of working. It was an early mover in internet banking, creating ING Direct in the late 1990s as a nonbranch offering. In 2007, ING merged its two Netherlands-based businesses, Postbank (a savings-only bank with no branches) and ING (a traditional retail bank). The transformation process was called TANGO (together achieving new growth opportunities), and it achieved annual savings of 280 million euros (roughly $330 million at today's exchange rates). In 2014, with the emergence of mobile banking, ING began rethinking

its entire model through a process called RIO (redesign into omnichannel). ING quickly realized that it needed to look beyond the banking industry for guidance. Specifically, ING found inspiration from Amazon, Spotify, and Zappos, where agile methods had demonstrably improved customer orientation and employee engagement.

ING also conducted an internal study, which highlighted how bureaucracy, silos, and risk aversion were cultural problems. Based on this analysis, ING decided on a top-to-bottom restructuring of its operations in the Netherlands, based mostly on Spotify's model but also on practices from Google, Netflix, and Zappos. The plan was to organize the 3,500 employees in Amsterdam into squads: teams of up to nine people with end-to-end responsibility for a specific customer-related activity. The squads would then work according to agile principles: a series of short "sprints" with frequent user feedback and daily progress updates.

ING went live with the new structure for its Netherlands operations on June 15, 2015. Eighteen months later, employee engagement was up (according to an internal survey to which I had access). In addition, ING's Net Promoter Score for its business in the Netherlands rose from –21% in 2015 to –7% in 2017, and its cost-to-income ratio in that business dropped over the same period from 65% to 51%. While the transformation is not finished, it is still fruitful to reflect on what ING has learned so far. I've grouped ING's lessons into five points.

1. Decide how much power you are willing to give up. In December 2014, ING executives flew to meet executives at Spotify. At that meeting, a Spotify executive said: "I can see you are fascinated by our way of working, but it's not that easy. You need to ask yourself honestly, how much are you willing to give up?" His point was that agile shifts power away from those at the top and puts ownership in the hands of those closest to the action. That is a difficult shift for executives at established companies.

How did ING handle this shift? The "big bang" approach meant that senior managers in the Netherlands had to embrace the new way or leave the company. Those who stayed had to reapply for the newly categorized jobs. This led to major personnel changes and a significant downsizing in the organization (a net reduction of about 1,500 employees in

This document is authorized for use only in Prof. V. S. Prakash Attili's MDP/207/21/EPITM/508 at Indian Institute of Management - Lucknow from Nov 2021 to Jan 2022.

SPECIAL COLLECTION • "STAYING AGILE" • **MIT SLOAN MANAGEMENT REVIEW 9**

the Netherlands from 2014 to 2016). All told, about one-third of the senior managers left.

The overarching lesson is that you cannot implement agile unless top executives accept that they are surrendering some status and power. "It requires sacrifices and a willingness to give up fundamental parts of your current way of working," said Schlatmann.

2. Prepare stakeholders for the leap. Some ING stakeholders were "completely freaked out by our proposals — they thought it would be complete chaos," recalled Schlatmann. It was one thing for Spotify and Netflix to adopt agile; it was quite another for a large bank to do so in the post-financial-crisis era.

How did ING's executives sell agile to nervous stakeholders? To the board, the executive team cited its track record with TANGO and RIO. They also argued that ING needed a new way of working to stay competitive in the lightning-fast digital marketplace. In addition, ING executives sought early buy-in from the works council representing employees, explaining how agile would engage and empower rank-and-file employees. "They accepted the notion that this was a one-time chance to really change the organization, and they ended up supporting us in a very positive way," said Schlatmann.

Meanwhile, bank regulators were concerned: They had never seen this structure in the industry. So ING's executives invited regulators to headquarters, where they could observe how agile brought to operations a habit of daily communication about progress and customer solutions. Most important, ING's executives assured regulators that finance, compliance, and legal functions would continue to be managed in their traditional way.

The lesson here is to assess — as early as you can — how stakeholders will react to a major change. Then find the right arguments to allay their concerns.

3. Build the structure around customers — and keep it fluid. The notion that work should be focused on customers is as old as the hills. Management thinkers such as Peter Drucker, W. Edwards Deming, Philip Kotler, and Theodore Levitt have all espoused their own variants of customer orientation. But agile goes a step further, forging a *structure* around customer needs. The basic building block for ING was a self-managed team, or squad, of up to nine people focused on a particular customer group. These squads were then clustered into larger *tribes* working on related activities. ING distinguished between *experience tribes* that attract new customers and *service tribes* that take care of existing customers. ING also defined two *enabling tribes* to serve these two customer-facing tribes. For example, one of the enabling tribes built black-box technical solutions for customer identification.

ING also seeks to keep its structure fluid so that it can evolve to do what's best for customers. For example, the experience tribe in charge of daily banking was, for a time, handling some customer communications duties. But eventually they shifted these duties to the tribe that specialized in communications. Likewise, ING has created the concept of "pop-up squads" to manage one-off, short-term projects.

The lesson here — regardless of any decision you make about agile — is to revisit your organizational structure to make sure it maps to the real needs of customers.

4. Give employees the right balance of oversight and autonomy. How do you ensure that squads prioritize important work? ING in the Netherlands moved to a quarterly business review (QBR) process adapted from Google LLC and Netflix Inc. Four times a year, each tribe lead writes a maximum six-page summary of what the tribe achieved, what they did not achieve (and why not), what they are going to achieve next quarter, and any dependencies outside the tribe's control.

These summaries are then discussed in a big meeting (the QBR Market) attended by tribe leads and other relevant leaders — about 20 people overall. They challenge one another's achievements and plans, and in doing so often resolve tensions or overlaps. Each tribe lead emerges with a set of objectives and key results (OKRs) for the following quarter. The OKRs then get translated into tasks for the individual squads within the tribe.

All of that has been a learning process for ING employees accustomed to a traditional goal-setting process. At first, tribe leads defined quarterly goals that were comfortably achievable. ING's executives had to urge them toward more ambitious targets — since the whole point was setting stretch goals, not erring on the safe side.

ING's experience is a reminder that you still need top-level oversight in an agile organization — to continually tweak the framework for goals and reporting, and to keep the level of ambition high.

5. Provide employees with development and growth opportunities. Squad-based structures can be scary for employees used to having their personal development and career progression mapped out by HR departments or the mainstream career trajectories of a given industry. Indeed, one risk of agile is that employees become *too* task-focused and results-oriented. They burn out and neglect to think about their careers over the long term. Having discovered this risk in its advance research, ING has taken steps to attend to employee development.

For example, ING in the Netherlands instituted weekly POCLAC meetings for each squad, where the activities of the squad and the development needs of individuals are discussed in tandem. POCLAC stands for product owner, chapter lead, agile coach — the three people responsible for empowering a squad. Though these meetings are undoubtedly a sound idea, they remain a work in progress. In fact, one area in which ING in the Netherlands has struggled in its agile transition is that the POCLAC meetings do not always happen on a weekly basis. Perhaps this isn't surprising: In most organizations, long-term individual goals easily get subsumed by and subordinated to short-term urgencies. What's more, in ING's old way of working, a manager was responsible for everything: the product, the process, and the people. With agile, each element is the responsibility of a different person. It's been an adjustment for chapter leads, product owners, and agile coaches to grow comfortable with all of their new responsibilities, let alone nonurgent matters like career development.

The lesson? Finding proper coaching and support for agile — and the new, long-term responsibilities employees must embrace — is one of the hardest parts of the transformation.

Lessons From ING

ING's experiences are a reminder that implementing new practices is much more difficult than suggesting them. The key challenges — shifting power from the top, getting buy-in from stakeholders, and changing employees' views about professional development — are operational concerns, rather than big-picture ones. No wonder new management practices often work better at young companies than they do at old ones, where the employees have entrenched expectations and habits.

But agile does have one advantage for established companies: It is now a bona fide way of working, with its own set of principles and a track record of success in certain sectors (mostly tech) and functions (mostly IT). In discussions with stakeholders, leaders can say that they are exploring a tested management model, rather than reinventing the wheel. Moreover, agile is starting to migrate into mainstream business — and ING in the Netherlands is at the forefront of this movement. By discussing the details of its experiences, I hope others can be encouraged toward comparable experiments and explorations.

Julian Birkinshaw (@JBirkinshaw) is a professor of strategy and entrepreneurship, academic director of the Institute of Innovation and Entrepreneurship, and deputy dean (programs) at London Business School. Comment on this article at http://sloanreview.mit .edu/59201.

REFERENCES

1. The original "Agile Manifesto" is available at http://agilemanifesto.org.

2. As part of this gathering momentum, a wide range of industry associations and training providers have emerged, such as Scrum.org, the Scrum Alliance, the Agile Business Consortium, the Agile Centre, and Lean Kanban.

3. S. Denning, "Explaining Agile," www.forbes.com, Sept. 8, 2016; and D.K. Rigby, J. Sutherland, and H. Takeuchi, "Embracing Agile," Harvard Business Review 94, no. 5 (May 2016): 40-50.

4. Challenges with implementing agile have been reported in several places. See, for example, J. Birkinshaw and M. Guest, "Digital Transformation in Practice," London Business School and Deloitte Institute of Innovation and Entrepreneurship, October 2016, www.london.edu; and B. Boehm and R. Turner, "Management Challenges to Implementing Agile Processes in Traditional Development Organizations," IEEE Software 22, no. 5 (September-October 2005): 30-39.

5. S. Denning, "Can Big Organizations Be Agile?" www.forbes.com, Nov. 26, 2016; and H. Kniberg, "Spotify Engineering Culture (Part 1)," labs.spotify.com, posted March 27, 2014.

6. J. Sutherland and J.J. Sutherland, "Scrum: The Art of Doing Twice the Work in Half the Time" (New York: Crown Business, 2014); S. McChrystal, with T. Collins, D. Silverman, and C. Fussell, "Team of Teams: New Rules of Engagement for a Complex World" (New York: Penguin, 2015); and K.S. Rubin, "Essential Scrum: A Practical Guide to the Most Popular Agile Process" (Upper Saddle River, N.J.: Pearson Education, 2013).

A New Approach to Designing Work

For years, management thinkers assumed that there were inevitable trade-offs between efficiency and flexibility — and that the right organizational design for each was different. But it's possible to design an organization's work in ways that simultaneously offer agility and efficiency — if you know how.

BY NELSON P. REPENNING, DON KIEFFER, AND JAMES REPENNING

?

THE LEADING QUESTION
How can companies achieve both agility and efficiency in their work?

FINDINGS
▶ **Make a distinction between well-defined and ambiguous tasks.**
▶ **Break processes into smaller units of work that are more frequently checked.**
▶ **Identify points at which collaboration is needed.**

YOU CAN HARDLY pick up a business publication without reading about the ever-increasing pace of change in technologies and markets and the consequent need for more adaptable organizations. Given the imperative of adaptability, it is not surprising that few words have received more attention in recent conversations about management and leadership than "agile."[1] Organizations ranging from large corporations like General Electric Co. to tiny startups are trying to be both flexible and fast in the ways that they react to new technology and changing market conditions.[2]

The word "agile" appears to have been first applied to thinking about software by 17 developers in 2001.[3] Having experimented with more iterative, less process-laden approaches to developing new applications for several decades, the group codified its experience in an agile manifesto. "We are uncovering better ways of developing software by doing it and helping others do it," they wrote. In software development, agile now has a variety of manifestations, including scrum, extreme programming, and feature-driven development.[4] The results have been significant. A variety of studies show that agile software development methods can generate a significant improvement over their more traditional predecessors.[5]

But what does this mean outside of software? Can agile methods be successfully applied to other types of work? Many proponents (a number of whom started in the software industry) argue that the answer is yes, and a growing collection of books, papers, and blog posts suggests how it might be done.[6] The evidence, however, remains limited to date, and a recent article by two

of agile's founders cautions against applying agile indiscriminately.[7] The blogosphere is also replete with discussions of an ongoing agile backlash.

To provide some practical advice to business leaders trying to understand what agile might mean for their organizations, we take a different approach. Our research suggests that in applying agile methods from the software industry to other domains, managers often confuse *practices* and *principles*. When agile methods work, they do so because the associated practices manifest key behavioral principles in the context of software development. But, successful as those practices can be when developing software, there is no guarantee that they will work in other contexts. The key to transferring a set of practices from one domain to another is to first understand *why* they work and then to modify them in ways that both match the new context and preserve the underlying principles.

The goal of this article is to help you understand several key work design principles that undergird not only agile practices in software but also Toyota Motor Corp.'s well-known production system in manufacturing. Once you understand these underlying work design principles — through a framework we call *dynamic work design* — you can create work processes in your own organization that are both more flexible *and* more efficient. (See "About the Research.")

Stability Vs. Uncertainty

Academics and managers alike long believed that organizations had to make trade-offs between flexibility and efficiency. A central notion in the academic theory on organizational design is *contingency*, the idea that organizations and their associated processes need to be designed to match the nature of the work they do. One of the most common variables in contingency theory is the degree of uncertainty in the surrounding environment (often also conceptualized as the need for innovation). When both the competitive environment and the associated work are stable and well understood, contingency theory suggests that organizations will do best with highly structured, *mechanistic* designs. In contrast, when facing highly uncertain situations that require ongoing adaptation, the theory suggests that organizations will do better with more flexible, *organic* designs.[8]

An early advocate of the mechanistic approach to work design was Frederick Winslow Taylor,

author of the 1911 book *The Principles of Scientific Management*.[9] Taylor's essential insight was simply that if work is regularly repeated, it can also be studied and improved. In stable, well-understood environments, it is thus often best to organize work in ways that leverage the efficiency that comes with repetition. For example, in a modern factory, well-defined tasks are specified, and the work proceeds serially, moving from one carefully constructed and defined set of activities to the next. There is little need for collaboration in these settings, and the organizational structure that surrounds stable and repeatable work tends to be hierarchical to ensure that everybody follows the prescribed work design. The cost of such efficiency is adaptability. Due to the high degree of routinization and formalization, mechanistic process designs are difficult to change in response to new requirements. Though efficient, a mechanistic design is not agile.

When, however, the environment is unstable and uncertain, discrete tasks are harder to define, and therefore organizations cannot rely on a sequence of clearly defined steps. For example, product development teams often face challenges for which there is little precedent. Contingency theory holds that in unpredictable environments like new product development, organizations rely more on things like training and collaboration and less on routinization and careful specification. Developing a breakthrough product or service usually can't be organized like a factory assembly line. Marketing experts may develop a set of initial requirements, which are then passed on to designers and engineers, but the requirements often evolve through multiple iterations as designers and engineers determine what is technically feasible. Consequently, effective development processes often require ongoing real-time collaboration, rather than rote adherence to a set of sequentially organized steps.

Though the contingency theory was first developed more than 50 years ago, its basic insights reappear frequently in contemporary management thinking. Many flavors of process-focused improvement, such as total quality management, Six Sigma, and business process reengineering, are extensions to Taylor's fundamental insight that work that is repeated can also be improved. Recently, the increasingly popular design thinking approach can be thought of as a charge to tackle ambiguous,

uncertain tasks with a more collaborative, less hierarchical work design.[10] In general, contingency theory gives managers a straightforward approach to designing work: Assess the stability of the competitive environment and the resulting work, and then pick the best mix of defined tasks and collaboration to fit the challenge at hand. (See "A Traditional Approach to Work Design.") If the work being designed consists of well-defined tasks (for example, assembling components), then it is best to organize it serially, or, as we label the cell on the bottom left, using the "factory" mode. Conversely, if the work is highly ambiguous and requires ongoing interaction (for example, designing new products), then the work is best organized collaboratively, or, as we label the cell on the top right, in "studio" mode.

Though powerful, this approach to work design is not entirely satisfying for two reasons. First, it describes an unpalatable trade-off: Work done using the serial factory design isn't very flexible, making it hard to adapt to changes in external conditions, and work done using the collaborative studio approach often isn't very efficient. Second, few types of work perfectly fit the archetype of well-defined or ambiguous work. Even the most routine work has the occasional moment of surprise, and conversely, even the most novel work, such as designing a new product or service, often requires executing routine analysis and testing activities that support each creative iteration. Academic theory notwithstanding, real work is a constantly evolving mix of routine and uncertainty.

At first glance, agile methods appear to fall more toward the collaborative side of the work spectrum. However, our research suggests a different interpretation. The conventional approach to process and organizational design is almost entirely static, implicitly presuming that once a piece of work has been designed, everything will go as planned. In contrast, a *dynamic* approach to work design suggests viewing

A TRADITIONAL APPROACH TO WORK DESIGN

In a traditional approach to work design, if the work being designed consists of well-defined tasks (for example, assembling components), then it should be organized serially, in what we call the "factory" mode. Conversely, if the work is highly ambiguous and requires ongoing interaction (for example, designing new products), then the work should be organized collaboratively, in what we call the "studio" mode.
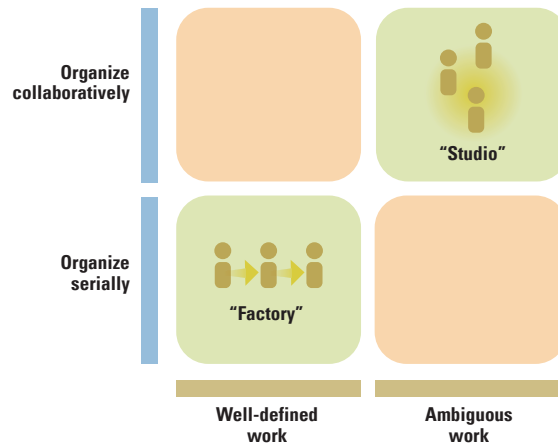
work as an ever-evolving response to the hiccups and shortfalls that are inevitable in real organizations. As we will describe later in this article, agile methods actually transcend the traditional serial vs. collaborative work framework by creating better mechanisms for moving *between* the two basic ways of organizing work. By identifying mechanisms to cycle back and forth between well-defined factory-style tasks and collaborative studio modes when appropriate, an agile approach can considerably reduce the trade-off between efficiency and adaptability.

Dynamic Work Design at Toyota

What does this look like in practice? Consider a well-known example of work and organizational design, Toyota's Andon cord. Work on Toyota assembly lines is the epitome of the serial, mechanistic design. Tasks are precisely specified, often detailing specific arm and hand movements and the time

> A *dynamic* approach to work design suggests viewing work as an ever-evolving response to the hiccups and shortfalls that are inevitable in real organizations.

that each action should take. In a plant we visited recently, training for a specific role began with the trainee learning to pick up four bolts at a time — not three and not five. Only when the trainee could pick up four bolts regularly was she allowed to learn the next motion. But, despite an attention to detail that would have made Taylor proud, sometimes things go awry. In the Toyota scheme, a worker noticing such an issue is supposed to pull what's known as the Andon cord (or push a button) to stop the production line and fix the problem.

While the management literature has correctly highlighted the importance of allowing employees to stop the line,[11] what happens *after* the cord is pulled might be more important. During a recent visit we took to a Toyota supplier in Toyota City, Japan, we observed that one operator on the factory floor was struggling to complete her task in the allotted time, and so she hit a yellow button, causing an alarm to sound and a light to flash. (This factory has replaced the Andon cord with a yellow button at each operator's station.) Within seconds, the line's supervisor arrived and assisted the operator in resolving the issue that was preventing her from following the prescribed process. In less than a minute, the operator, now able to hit her target,

returned to her normal routine, and the supervisor went back to other activities.

What, from a work design perspective, happened in this short episode? Initially, the operator was working in the "factory" mode, executing well-defined work to a clearly specified time target. (See the box on the lower left in the exhibit "Dynamic Work Design at a Toyota Supplier.") But when something in that careful design broke down, the operator couldn't complete her task in the allotted time. Once the problem occurred, the operator had two options for responding. She could have found an ad hoc adjustment, a workaround or shortcut that would allow her to keep working. But this choice often leads to highly dysfunctional outcomes.[12] Alternatively, as we observed, she could push the button, stop the work, and ask for help. By summoning the supervisor to help, pushing the button temporarily *changed* the work design. The system briefly left the mechanistic, serial mode in favor of a more organic, collaborative approach focused on problem resolution. Once the problem was resolved, the operator returned to her normal task and to the serial work design.

The Toyota production system might at first appear to be the ultimate in mechanistic design, but a closer look suggests something far more dynamic. When a worker pulls the Andon cord, the system actually moves between two modes based on the state of the work. Though the nature of the work couldn't be more different, such movement between the two modes is also the key to understanding the success of agile software development.

DYNAMIC WORK DESIGN AT A TOYOTA SUPPLIER

At a Toyota supplier, a worker on an assembly line can press a button if he or she faces a problem. A manager then helps solve the problem through collaboration; once the problem is solved, the worker returns to his or her task. Pushing the button thus initiates a temporary shift in the work design — from serial to collaborative work and then back again — that increases agility.
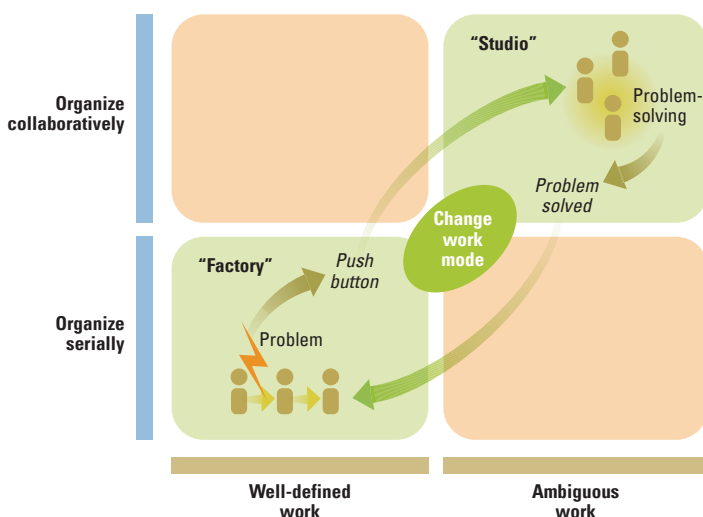
Agile as Dynamic Work Design

As we discussed earlier, the last two decades have witnessed a significant change in the conduct of software development. Whereas software was once largely developed using what is known as the *waterfall* approach, agile methods have become increasingly popular. From a dynamic work design perspective, the waterfall and agile approaches differ significantly.

In the waterfall approach, the software development cycle is typically divided into a few major phases. A project might include a requirements phase, an architecture development phase, a detailed coding phase, and a testing and installation phase. A waterfall project typically cycles between three basic

> Checking in with more senior leadership only in the form of periodic phase-gate reviews means that the entire team could work for months before realizing they are not meeting management's expectations.

modes of work. First, the bulk of the time is spent by software architects and engineers working individually or in small groups, completing whatever the specific phase requires. Second, typically on a weekly basis, those people leave their individual work to come together for a project meeting, where they report on their progress, check to ensure mutual compatibility, and adapt to any changes in direction provided by leadership. Third, at the end of each phase, there is a more significant review, often known as a "phase-gate review," in which senior leaders do a detailed check to determine whether the project is ready to exit that phase and move to the next. Development cycles for other types of non-software projects often work similarly.[13]

Agile development processes organize the work differently. For example, in the scrum approach[14] (one version of agile), the work is not divided into a few major phases but rather into multiple short "sprints" (often one to two weeks in length) focused on completing all of the work necessary to deliver a small but working piece of software. At the end of each sprint, the end user tests the new functionality to determine whether or not it meets the specified need.

Like the waterfall method, the agile approach to software development also has three basic work modes — individual work, team meetings, and customer reviews — but it cycles among them very differently. First, proponents of agile suggest meeting *daily* — thus moving from individual work to teamwork and back every day — in the form of a stand-up or scrum meeting, where team members report on the day's progress, their plans for the next day, and perceived impediments to progress. Second, agile recommends that at the end of each sprint, the team lets the customer test the newly added functionality. Finally, in something akin to the Andon cord, some versions of agile also include an immediate escalation to the entire team when a

piece of code does not pass the appropriate automated testing, effectively again moving the system from individual work to the team collaboration mode.

Viewed from a dynamic work design perspective, agile offers two potential benefits over waterfall. First, in waterfall development, the frequency of collaborative episodes is usually too low, both among the team members and between the team and its customers. A developer working for a week or two without a check-in could waste considerable effort before it's clear that he or she has made a mistake or gone off course. In practice, developers often do not wait this long and informally check in with supervisors or teammates. While seemingly functional, these check-ins can lead to a situation in which the entire team is not working from a common base of information about the state of the project. In such cases, the operating mode starts to migrate from the box on the lower left, the "factory" mode, to the one on the lower right, where ambiguous work is organized serially. This results in costly and slow iteration, which we call *ineffective iteration*. (See "Dysfunctional Dynamics," p. 35.) Research suggests that in R&D processes, this mode can be highly inefficient.[15] Similarly, checking in with more senior leadership only in the form of periodic phase-gate reviews means that the entire team could work for months before realizing that it is not meeting management's expectations, thus also potentially causing rework.

The agile approach to software development also improves the quality of the time that developers spend working alone. The focus on developing pieces of functionality means that both the team and the customer are never more than a few weeks away from a piece of software that can be used, making it far easier to assess whether it meets the customer's need. In contrast, in waterfall, the early phases are characterized by long lists of requirements and features, but there is nothing to try or test. It's not surprising that waterfall methods often lead to projects in which major defects

and other shortfalls are discovered very late in the development cycle and require costly rework.[16]

Applying Dynamic Work Design

Both the Toyota production system and agile-based software methods are thus examples of what we call good *dynamic work design*. In contrast to traditional static approaches, dynamic work design recognizes the inevitability of change and builds in mechanisms to respond to that. Once managers recognize the necessity of moving between more individual and more collaborative modes of work, they can build on four principles to create shifting mechanisms that are well matched to the work of their organization.

1. Separate well-defined and ambiguous work. Begin by clearly separating well-defined and ambiguous tasks. Trying to handle both types of work in the same process often leads to trouble. (See "Dysfunctional Dynamics.") Often, the two types can be separated by inspection, but if not, then look for the signature element of ambiguous work, iteration. When work is well defined, it can be moved to the next stage like the baton a relay runner hands off. When done correctly, it doesn't need to come back. In contrast, when work is ambiguous, even the best effort often needs to be revisited. If you find that a particular task often requires multiple iterations through the same set of steps, that's a good sign that you are confronting ambiguity inefficiently.

2. Break processes into smaller units of work that are more frequently checked. If you strip away all the hype, the agility of any work process — meaning its ability to both adjust the work due to changing external conditions and resolve defects — boils down to the frequency and effectiveness with which the output is assessed. In both traditional, pre-Toyota manufacturing and waterfall software development, the assessments are infrequent and not particularly effective. Consequently, both approaches tend to be slow to adjust to changes in the external environment, and quality will be achieved only through slow and costly rework cycles. In contrast, when assessments are frequent and effective, the process will be highly adaptable and quality will improve rapidly. The fundamental recipe for improved process agility is this: *smaller units of work, more frequently checked*.

3. Identify the chain of individuals who support those doing the work. It is also important to identify the *help chain* — the sequence of people who support those doing the work. In manufacturing, the help chain starts with a machine operator and extends from foremen to supervisors all the way up to the plant manager. In software, the help chain often begins with an engineer and moves through the team leader to more senior managers, ultimately ending with the customer. It is critical, in our experience, that you identify the chains of individuals who do and support the work, not their roles, departments, or functions. Increasing agility requires knowing whom to call when there is a problem or feedback is needed.

4. Introduce triggers and checks that move work into a collaborative mode. Once you understand the help chain, you have two basic mechanisms for activating it: *triggers* and *checks*. A *trigger* is a test that reveals defects or misalignment and then moves the work from a factory mode to a more collaborative mode. In our opening example, the Toyota operator's inability to complete the assembly task on time triggered her pushing a button and then receiving help from a supervisor. A *check* involves a prescheduled point when the work is moved to a more collaborative environment for assessment. In agile software development, this shift happens daily in stand-up meetings where the team quickly assesses the current state of the project. Completing a sprint creates a second opportunity, this time to check in with the customer.

Improving Procurement Performance

Using this dynamic work design framework within a company can lead to significant improvements in both efficiency and adaptability. Consider the case of a company we'll call "RefineCo," which owns several oil refineries and distribution terminals in the United States. The company had a procurement organization that was uncompetitive by almost any benchmark. RefineCo paid more for similar parts and services than its competitors, and the procurement group's overhead costs were higher than the industry average. Even more troubling, when critical parts were not delivered to a refinery, it often turned out that the location was on "credit hold" due to an inability to pay the supplier in a timely fashion. Every participant in the system, from senior management down to the shipping and receiving clerks, was frustrated.

DYSFUNCTIONAL DYNAMICS

What happens when organizations don't do a good job of cycling between factory and studio modes of work? We have observed two related failure modes, *ineffective iteration* and *wasted attention*. When they are combined, they create a truly unproductive work design — one we have dubbed *the axis of frustration*. (See "The Axis of Frustration.")

Ineffective Iteration Consider first what happens when elements of the work in question are highly ambiguous but are nonetheless organized serially (captured in the box in the lower right-hand corner). Relative to a more collaborative design, this approach tends to create slow and costly iteration. The lack of speed comes because the ambiguity must travel among participants to be resolved, thus requiring multiple rounds, each of which takes time. Worse, when knowledge work is designed serially, many of these interactions take place through email or text messaging. Research suggests both that such communication modes are less effective for reducing ambiguity than face-to-face communication *and* that those sending such messages are unaware of those limits.[i] Trying to resolve ambiguity via email or text messaging tends to create more misunderstandings and often necessitates multiple iterations.

Wasted Attention On the flip side, organizing well-defined work in a collaborative fashion also creates inefficiency. If the work is clearly defined, then it doesn't benefit from a collaborative approach, and collaboration just multiplies the cost. Worse, too much collaboration may prevent the efficiencies that come with the learning curve that emerges when people repeat the same task.[ii]

The Axis of Frustration Whereas functional work processes move between the factory and studio modes, our research suggests that absent careful design attention, processes can devolve to the point where they move between the failure modes described above, oscillating between wasted attention and ineffective iteration — the dynamic we call the axis of frustration.

Getting stuck on the axis of frustration typically starts with time pressure — a project is behind schedule or a more repetitive process is not delivering on its targets. When people feel they are behind, they don't want to take the time to shift into collaborative studio mode for problem-solving, preferring to stay in the factory box on the lower left and "just get the work done." The consequence of this decision is to leave one or more problems unresolved, whether it is an element of a product design that doesn't work or a defect in a manufactured product. Eventually, these problems will be discovered, usually by an activity downstream from the one that generated it. And, if this problem is not then solved in collaborative studio mode (again due to time pressure) but instead sent back for rework, then the system has effectively moved from the box on the lower left to the box on the lower right and is now in "ineffective iteration" mode.

The consequence of ineffective iteration is that the process becomes increasingly inefficient and incapable of meeting its targets. Senior leaders are, of course, unlikely to stand idly by and will eventually intervene. Unfortunately, the typical intervention is often to scrutinize the offending process in more detail, usually in the form of more frequent and more detailed review

meetings. (As a manager we once interviewed said, "I knew my project was in trouble when I was required to give hourly updates.") But the form of those reviews makes all the difference. If they are well designed and focus on resolving the key problems that are causing the iteration, then they can move the system back to a more productive cycling between factory and studio modes. Such interventions, however, are the exception rather than the rule.

THE AXIS OF FRUSTRATION
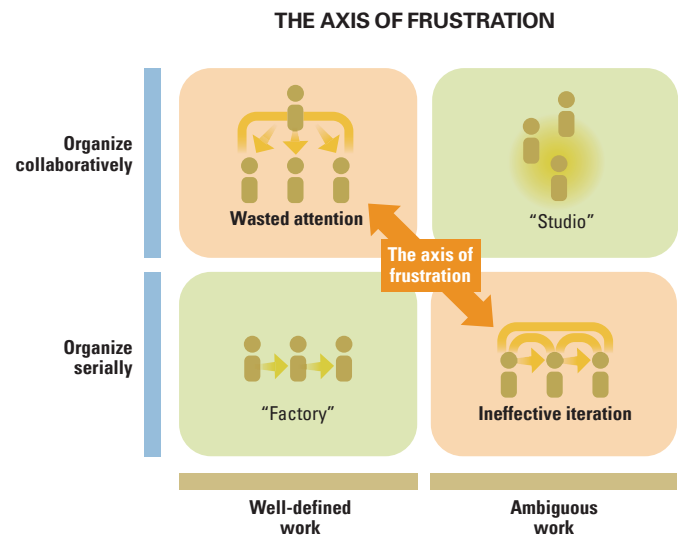
When organizations make the mistake of both structuring well-defined work collaboratively and ambiguous work serially, the result is a highly inefficient process we call *the axis of frustration*. This process oscillates between wasted attention and ineffective iteration.

Most work processes have not been designed with escalation mechanisms in mind. So, when senior managers want to intervene and scrutinize a project, they don't know where to look and want to review *everything*. The result of such scrutiny is long review meetings, the majority of which focus on elements of the process that are just fine, thereby trapping the process in the upper left-hand box, "wasted attention." Worse, long review meetings and the preparation that they require steal time and resources from actual work, thus intensifying the time pressure that prevented a proper shift between work modes in the first place. Without careful attention to the mechanisms that move a process between the individual and collaborative modes, processes can increasingly cycle between ineffective iteration and wasted attention, basically moving between frantically trying to solve (or at least hide) the latest problem before the next review, and endless, soul-destroying review meetings that never get to solving the problems that would really make a difference.

"Long turnaround times created unhappy customers and suppliers who constantly called to complain and ask about their particular order or payment.

The procurement system at each of RefineCo's sites worked roughly as follows. To purchase an item or service from an outside vendor, an employee would enter the requirements into the electronic procurement system, which would then appear as a request to the central procurement function. The staff in the procurement office would then review the request and issue a purchase order. That order would go to the supplier. When the product arrived at the refinery or the service was completed, a packing slip or service order verification slip would be generated, which would also be entered into the procurement system. Later, the supplier would generate an invoice that was also entered into the system. The electronic system would then perform a three-way match to verify that everything was done correctly: The purchase order should match the verification receipt, which, in turn, should match the invoice. If there was not a three-way match, the invoice would be "kicked out" of the system and the supplier would not get paid until the discrepancy was resolved.

The job of resolving those discrepancies fell to the staff in the refinery's purchasing office. Unfortunately, the products and services procured frequently failed the three-way match, leading to both an overburdened purchasing department and frustrated suppliers. Though the refinery was part of a large and successful company, it was frequently on credit hold with its suppliers for failure to pay invoices on time, making it difficult for the staff to do their jobs and run the plant safely. The dedicated procurement staff worked 10-plus hours per day and had hired temporary workers to help manage the backlog, but they were still falling behind.

Most of the members of the procurement team complained bitterly about being "overworked" and how "screwed up the system was." Nobody saw any opportunity for improvement beyond adding what appeared to be much-needed staff. For us, the critical moment in our work with the procurement staff came when one of the longtime team members

explained that a good purchase request contained "all the information I need" and could be turned into an official purchase order in "five to 10 minutes." A difficult one, however, lacked key pieces of information and might require one to two hours to process as the purchasing staff traded emails with both the requesting unit and the supplier. Despite this effort, difficult purchase orders were usually the ones that failed the three-way matching process and got kicked out of the system. Further investigation revealed that the purchase order system was completely gridlocked with the kicked-out orders, and the team spent much of its time trying to clear the backlog. The system had descended into the classic "expediting" or "firefighting" trap: There were so many purchase orders in process that the turnaround time for any given one was very long. But long turnaround times created unhappy customers and suppliers who constantly called to complain and ask about their particular order or payment. Consequently, the procurement team was constantly reprioritizing its work and reacting to whichever customer or supplier was most unhappy.

Our first insight came in recognizing that the procurement team was engaged in two different types of work that corresponded to what we call serial "factory" work and collaborative "studio" work. When the requested item was standard and all the needed information was provided, a single person could easily process the request without collaboration; then, once the purchase order was entered, it would easily flow through the system, just like an item on an assembly line. However, standard requests flowed easily through the system only *if* the request came with the correct information. If it did not, then it could require several rounds of iteration, usually via email, to issue the purchase order. So the purchasing function created a simple checklist that described a good purchase request. The idea was to ensure that standard orders would always arrive with the correct information. To give the various departments an incentive to use the checklist, the

purchasing function promised that any request received by 7 a.m. *with the proper information* would result in a purchase order being issued by 2 p.m. that day. At that time, a one-day turnaround was unheard of because every order simply went into the "to do" pile. The purchasing department also created a simple trigger to improve productivity: Purchase orders that were missing items on the checklist would be immediately returned to the requesting unit.

The second part of the intervention came in recognizing that not every request could be supported in factory mode. In the existing system, neither the requesters nor the purchasing staff distinguished between a standard request and a novel one. Thus, when a request for a new product or service showed up, the agent would do his or her best to process it, typically requiring multiple emails with the requester, often over several days, to nail down all the relevant information. In many cases, when the agents couldn't get the information they needed, they would make their best guess and then submit an incomplete or incorrect purchase order. This, too, created additional iteration, as the supplier, unsure of what was being requested, would call or email the agent. The purchasing process was thus living in the lower right-hand box of our matrix, attempting to accomplish ambiguous work in a serial fashion and thereby creating slow and expensive iteration.

Creating an effective collaborative studio mode to handle the complex purchase orders required two changes in work design. First, the team created a clear trigger: If a request was nonstandard, then it was moved into a separate pile and not dealt with immediately. Second, each day at 2 p.m., the team would work together to process the more complex cases. By working collaboratively (in studio mode), they were able to resolve many of the more complex cases without additional intervention — somebody on the team might have seen a similar order before. Also, having a face-to-face meeting was far more efficient than the endless chain of email that it replaced. And, if additional information was needed, the team could schedule a phone call in the time window after 2 p.m., rather than send an email, again reducing the number of expensive iterations.

The results of these two changes were significant. Creating a factory mode for the standard orders allowed the team to make good on its "in by 7, out by 2" promise almost immediately, generating an immense amount of goodwill with the requesters. Spending the afternoon in studio mode also sped the processing of the complex orders. The two changes created enough space that the team was able to use studio time to not only process the more complex requests but also work through the backlog of unresolved older orders. In the end, due to the efficiency improvements, the procurement team reduced its staff by the equivalent of two full-time staff members, while providing far faster and more reliable service. These process improvement insights were then applied to the company's other U.S. sites and, as of this writing, RefineCo pays more than 90% of its invoices on time, resulting in a far happier collection of suppliers.

Look for Best Principles

Managers and consultants are often obsessed with the search for best practices — those activities that appear to separate leading organizations from the rest of the pack. The idea behind this search is that once identified, best practices can be adopted by other organizations, which will then experience similar gains in performance. While there is certainly *some* truth to this idea, the supporting evidence is decidedly mixed. Organizations frequently struggle to implement new tools and practices and rarely experience similar gains in performance. In many industries, the performance gap between the top and middle performers remains stubbornly difficult to close. A key reason for these failures is simply that organizations are complex configurations of people

Organizations are complex configurations of people and technology, and a set of tools or practices that works well in one context might not be equally effective in a major competitor — even if that competitor is located just down the street.

and technology, and a set of tools or practices that works well in one context might not be equally effective for a major competitor — even if that competitor is located just down the street.

Best practices are "best" when they manifest an underlying behavior principle in a way that is well matched to the organization that uses them. Toyota's famed Andon cord and the localized problem-solving it catalyzes work by capitalizing on the efficiency that comes from individual repetition and the innovation that comes with collaborative problem-solving. Conversely, agile development methods work by channeling the creativity of software engineers through frequent team meetings and customer interactions. More generally, organizations become more adaptable when they find defects and misalignments sooner. A dynamic approach to contingency, supported by triggers and checks, can open the path to creating practices that support increased agility in the work of *your* organization.

Nelson P. Repenning *is the School of Management Distinguished Professor of System Dynamics and Organization Studies at the MIT Sloan School of Management in Cambridge, Massachusetts, where he currently serves as the associate dean for leadership and special projects and the faculty director of MIT's Leadership Center.* **Don Kieffer** *is a senior lecturer in operations management at the MIT Sloan School and a founder of ShiftGear Work Design, a consulting firm based in Cambridge.* **James Repenning** *is the managing partner at ShiftGear Work Design. Comment on this article at http://sloanreview.mit.edu/59234.*

REFERENCES

1. A quick trip to the web reveals multiple manifestations, including "agile principles," "enterprise agile," "agile organizations," and a variety of suggestions, including the charge to "apply agile development to every aspect of business." See, for example, S.D. Goldstein, "Apply Agile Development to Every Aspect of Business," Mint, Dec. 4, 2016, http://www.livemint.com.

2. D. Leonard and R. Clough, "How GE Exorcised the Ghost of Jack Welch to Become a 124-Year-Old Startup," Bloomberg, March 17, 2016, www.bloomberg.com.

3. See their manifesto here: http://agilemanifesto.org.

4. For a summary, see F.S. Glaiel, A. Moulton, and S.E. Madnick, "Agile Project Dynamics: A System Dynamics Investigation of Agile Software Development Methods," Massachusetts Institute of Technology, Engineering Systems Division working paper, Oct. 2014, available at http://dspace.mit.edu.

5. See C.J. Stettina and J. Hörz, "Agile Portfolio Management: An Empirical Perspective on the Practice in Use," International Journal of Project Management 33, no. 1

(January 2015): 140-152; J. Sutherland and J.J. Sutherland, "Scrum: The Art of Doing Twice the Work in Half the Time" (New York: Crown Business, 2014); and D. West and T. Grant, "Agile Development: Mainstream Adoption Has Changed Agility," Forrester, Jan. 20, 2010, www.forrester.com.

6. See, for example, M.E. Moreira, "Being Agile: Your Roadmap to Successful Adoption of Agile" (New York: Apress, 2013).

7. D.K. Rigby, J. Sutherland, and H. Takeuchi, "Embracing Agile," Harvard Business Review 94, no. 5 (May 2016): 40-50.

8. The classic descriptions of contingency theory can be found in T. Burns and G.M. Stalker, "The Management of Innovation," rev. ed. (Oxford, U.K.: Oxford University Press, 1994) and in P.R. Lawrence and J.W. Lorsch, "Organization and Environment: Managing Differentiation and Integration," rev. ed. (Boston: Harvard Business School Press, 1986). An updated summary can be found in several textbooks, including R.M. Burton, B. Eriksen, D.D. Håkonsson, and C.C. Snow "Organization Design: The Evolving State-of-the-Art" (New York: Springer Science+Business Media, 2006).

9. F.W. Taylor, "The Principles of Scientific Management" (New York and London: Harper & Brothers, 1911).

10. T. Brown, "Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation" (New York: HarperBusiness, 2009).

11. J. Liker, M. Hoseus, and the Center for Quality People and Organizations, "Toyota Culture: The Heart and Soul of the Toyota Way" (New York: McGraw-Hill Education, 2008).

12. See, for example, N.R. Repenning and J.D. Sterman, "Capability Traps and Self-Confirming Attribution Errors in the Dynamics of Process Improvement," Administrative Science Quarterly 47, no. 2 (June 2002): 265-295; and N. Leveson, "A Systems Approach to Risk Management Through Leading Safety Indicators," Reliability Engineering and System Safety 136 (April 2015): 17-34.

13. K.T. Ulrich and S.D. Eppinger, "Product Design and Development," 6th ed. (New York: McGraw-Hill Education, 2016).

14. Sutherland and Sutherland, "Scrum"; and Scrum Guides, www.scrumguides.org.

15. L.A. Perlow, "The Time Famine: Toward a Sociology of Work Time," Administrative Science Quarterly 44, no. 1 (March 1999): 57-81.

16. See Sutherland and Sutherland, "Scrum"; and J. Kamensky, "Digging Out of the Digital Stone Age," Government Executive, March 9, 2017, www.govexec.com.

i. N. Epley and J. Kruger, "When What You Type Isn't What They Read: The Perseverance of Stereotypes and Expectancies Over E-Mail," Journal of Experimental Social Psychology 41, no. 4 (July 2005): 414-422.

ii. L. Argote and D. Epple, "Learning Curves in Manufacturing," Science 247, no. 4945 (Feb. 23, 1990): 920-924.

Goodbye Structure; Hello Accountability

JEANNE ROSS

Digital technologies offer ubiquitous data, unlimited connectivity, and massive processing power. These capabilities have created a business environment in which decision-makers can more readily acquire data to inform decisions and then shrink bottlenecks between knowing and doing. In this way, digital technologies are accelerating the pace of business.

All this speed is making business agility *de rigueur.* No matter how big or old they are, companies that hope to avoid a fate like that of the Titanic — sinking because they can't change course fast enough to avoid calamity —

must learn how to respond quickly to unanticipated opportunities and threats.

Well-designed systems and processes can certainly help make a company more agile. But even great systems and processes are responsive to change only if the people who use them recognize what needs to be done and how to do it.

Organizational structures are designed to clarify how a company will meet stated objectives. But they're not necessarily good at helping people adapt to changing objectives. The result: Leaders will be able to operate as true digital leaders only when they shake their reliance on structure as the primary tool of organizational design and instead start assigning accountabilities in ways that instigate focused responses to opportunities. This is unlikely to come naturally.

What's Wrong With Structure?

Traditionally, business leaders have structured companies to allocate responsibilities to distinct functions, product lines, geographies, or other business areas. This divide-and-conquer approach to organizational structure creates business silos, which are usually designed as hierarchies. In these structures, leaders assign increasingly specific responsibilities and tasks at lower levels of the hierarchy.

These siloed, hierarchical structures support operational efficiencies, but they are not effective in supporting digital, integrated services to customers. Integration requires coordination of activities across silos. Recognizing this interdependence, decision-makers refer a growing number of decisions up the hierarchy, where senior people can resolve conflicts across all affected business silos. As a result, key decisions are made far away from the operational reality, and then communicated back to where action will be taken.

Digital companies cannot wait for such elongated decision-making processes. But because they also cannot afford to ignore the need to coordinate across silos, leaders are reluctant to empower teams within silos to take independent action.

Restructuring Is Not the Answer

As business leaders recognize the limitations of business silos and hierarchies, they invariably attempt to add new structures, like matrices or networks, to make their structures more agile.

But adding structures is likely to make a company more complex rather than more agile. There are at least four reasons why executives should not rely on structure to provide improved organizational agility:

- **Restructuring is a bad use of management time (and everyone else's).** Reorganizations are exhausting. Constant introductions of new structures will consume precious management attention that should be focused on meeting new business demands.

- **Structures trap rather than empower employees.** Roles within formal structures rarely encourage people to do what it takes to solve a problem. Rather, they encourage people to do the job they were told to do. It's agile employees who solve problems.

- **Formal organizational structures often limit experimentation.** Risk-averse individuals may be reluctant to try new things or note that an experiment is failing, if it bodes poorly for their organizational unit. Commitment to structure as a key design lever can limit learning.

- **Formal structures don't fully leverage a company's smarts.** Traditional structures come with annual goals, budgets, and performance metrics. This means that the people at the top need to know what resources are necessary — they need to be the smartest people in the room. But when companies are moving fast, business awareness and creativity are highly distributed.

Most leaders recognize these limitations — they just fear the chaos that could result if they rely less on formal structures. Indeed, structures won't go away. But what *can* go away is the reliance on restructuring as a way of introducing important strategic changes.

Lead Change by Assigning Accountabilities

Instead of restructuring, companies can initiate change by assigning accountabilities for specific business outcomes to small teams or individual problem owners.

For example, in our research at MIT's Center for Information Systems Research (CISR), we've seen that at the banking and investment company BNY Mellon, more than 50 service leaders are accountable for creating and maintaining services like opening an account, making a payment, and reconciling a transaction. These service leaders are responsible for cost, reliability, and customer satisfaction (often internal) related to their service, so they coordinate with other parts of the company through collaboration as opposed to working through the hierarchy. This arrangement is allowing BNY Mellon to provide a more integrated face to customers without restructuring the business.

Leading change by assigning accountabilities involves specifying a desired outcome, putting someone in charge, and letting the responsible person decide how to accomplish the objective. Senior leaders need not divvy up necessary tasks; individuals or teams can quickly pivot as they identify what is and isn't working.

For instance, like many digital companies, the Swedish streaming entertainment company Spotify supports its customer offerings through the efforts of small autonomous teams, known as squads. These squads define their own missions and develop their own goals, as well as hypotheses as to how they will meet their goals, testing and adjusting as they go along. Team membership shifts as the company refocuses priorities or as a team grows beyond an optimal size.

Assigning accountabilities like this differs from structuring in several important ways:

Individual flexibility. Companies designed around accountabilities focus on current issues and outcomes. For digital companies, most accountabilities revolve around digital offerings. These are natural because offerings start as minimum viable products that can grow if customers demonstrate enthusiasm. For successful offerings, assignments are likely to become longer term (and could lead to a more formal structure). Other individuals take on accountability for specific process problems that, once resolved, will lead to reassignment. People start to expect (and, in most cases, desire) to move to where they are most needed.

Less budgeting and more market-based resource allocation. Accountability owners take responsibility for making their solutions cost-effective. Owners of accountabilities attempt to meet the needs of their internal or external customers at a price those customers are willing to pay. They are inclined to recruit resources as needed rather than simply accrue a reservoir of talent. In some cases, they'll recruit people to their teams; in others, they will look for collaborators. Thus, an internal market, rather than a budget, will dictate resource levels.

Experimental mindset. Because every accountability owner designs experiments to test hypotheses, it's important to track whether an experiment is meeting with success or failure. Accountability owners accompany their hypotheses with proposed milestones that indicate if an experiment needs scaling up, tweaking, or abandonment. Failure is an option as long as the company has identified how to capture learning.

Coaching rather than managing. Because accountability owners are proposing hypotheses, defining metrics to track progress, and assuming responsibility for recruiting and paying for resources, they will usually know more about their roles than their leaders do. Thus, leaders must rethink their roles. To lead empowered teams, leaders

have two important responsibilities: (1) aligning multiple accountabilities and (2) coaching individual accountability owners.

Assigning accountabilities rather than developing structures will be a radical new way of working for people who have climbed the corporate ladder in traditional ways. The demands for agility, however, make it imperative that leaders learn how to help their people adapt quickly to new demands and opportunities.

Getting Started

To develop accountability at all organizational levels, leaders must learn to empower their people without creating chaos. In turn, accountability owners must learn how to take responsibility for organizational outcomes.

When former BNY Mellon Global CIO Suresh Kumar revamped the IT unit at BNY Mellon around accountabilities, he noted that he was asking people to act like "mini-CEOs." He found that some of the people he placed in this role were natural problem-solvers who seized the opportunity. Others regularly asked their bosses what they should do next. Not all could be trained out of old ways of thinking.

To get started, leaders should solicit the names of people who feel they have far more to add to the company than they are contributing in their current roles. This will help identify problem solvers and risk-takers. Leaders can assign them a single issue or digital offering. As they gradually designate new tasks to new accountability owners, leaders themselves will learn how to coordinate and coach.

There will be mistakes, and people throughout the company should know about those mistakes and learn from them. But as the process unfolds, it will accelerate the company's digital transformation.

About the Author

Jeanne Ross is principal research scientist for MIT's Center for Information Systems Research.

Architect Your Company for Agility

JEANNE ROSS

Recent operational issues at Tesla Inc., the future-forward electric-car maker based in California, serve as a reminder that a great strategy is valuable only if a company is capable of executing that strategy. And whether or not a company can execute its strategy depends largely on whether it is designed to do so. In other words, it depends on *business architecture* — the way a company's people, processes, systems, and data interact to deliver goods and services to customers.

As companies now develop digital strategies, they are invariably promising integrated customer solutions. These strategies will be especially difficult to execute because so many organizational elements must be synchronized to deliver an integrated solution. And it's not just the number of organizational elements that makes digital strategy execution difficult. Speed matters. To keep pace with customer demands and competitor moves, companies must be able to quickly experiment with a potential offering and, depending on customer response, continuously enrich and scale that offering, or discard it and move on to the next experiment.

In other words, business architecture just became more important — and more difficult. In the pre-digital economy, business architecture most often focused on operational efficiency — designing seamless end-to-end business processes. That is no longer sufficient. In the digital economy, business architecture must also focus on agility — designing rapid reuse of individual business components.

Organizational agility won't happen by accident. It must be architected.

Innovating at Speed Means Utilizing Empowered Teams

To facilitate speed, companies must design themselves to minimize the obstacles to getting work done. This requires empowering and supporting problem-solvers.

To this end, growing numbers of companies are creating small, cross-functional, agile teams. Each team owns delivery of a digital offering or a set of services contributing to an offering. Typically, teams clarify their own objectives and define their metrics for success.

Spotify AB, the digital music service, serves as a model for many companies' forays into empowered teams. The Stockholm-based company relies on small teams, called squads, to deliver product features and related business components. Squads are assembled into tribes that provide major offerings and capabilities, such as online music offerings or services in support of artists or advertisers.

The concept of continuous release is essential to the effectiveness of empowered teams. At Spotify, squads release new digital features and offerings as quickly as they become viable rather than conform to scheduled release dates. Rapid innovation depends on teams learning quickly what does and doesn't work. Empowered teams experiment, and the best teams learn how to rapidly respond to the outcomes of their experiments.

These teams don't fear failure. In fact, critical to the concept of empowered teams is that when experiments fail, company managers do not assume responsibility for dictating how to fix things. Instead, they act as coaches, posing questions and eliciting hypotheses and expected outcomes.

Failed experiments are essential to learning. If an entire concept is deemed a failure, the team can be disbanded and reassigned to a new initiative. This helps to keep team formation fluid. Unlike traditional organizational structures, a design built on empowered teams is in a constant state of change.

The Critical Role of Alignment

Empowered teams invariably contribute innovation — and high energy — to companies. The challenge is ensuring that the efforts of individual teams align to achieve company-wide objectives. Our research at the MIT Center for Information Systems Research has identified three mechanisms for achieving alignment: (1) clear missions, (2) common business components, and (3) fruitful knowledge sharing.

Missions provide direction, both at the enterprise and at the individual team level. At the enterprise level, a clear mission or vision statement establishes priorities for the entire organization. It directs teams' innovation efforts by clarifying the objectives of the company's investments in resources. At the team level, mission statements define how the team will contribute to the company's goals.

At Spotify, managers told us that the company defines a small set of "big bets" that establishes enterprise priorities based on beliefs managers derive from their data. Individual teams then state missions to help achieve the big bets. For example, the mission of a music delivery tribe is "providing fast and reliable access to all the world's music." The mission of an infrastructure tribe is "enabling high product development speed while maintaining a highly available service." A clear mission will guide a team's choice of metrics, so that the team can easily track its progress in contributing to the enterprise mission.

To respond to changes in customer needs or market conditions, companies can redefine missions at the enterprise or individual team level. Restating a mission

enables a company to adjust priorities without necessarily requiring changes in organizational structures.

To ensure alignment across teams, companies need to embrace reuse. Reusable business components such as on-boarding processes, dashboarding, and payment systems, facilitate integration and speed. New business capabilities build on existing capabilities and provide consistency across offerings.

For example, the Dutch electronics company Philips is building digital offerings to enable seamless health care for its hospital clients in accordance with a vision to "build a healthier tomorrow." To support this effort, an internal platform team reviews all propositions for new digital offerings to distinguish unique business needs from common needs. The team then establishes a road map to ensure that common business components are available when needed.

Technical standards are essential to business component reuse. Standards for application programming interface (API) development, for instance, ensure that teams approach development of their business components so that they are API-enabled and available through the company's catalogue of internal services.

Finally, empowered teams depend on knowledge sharing to coordinate their activities and share their learning. It appears that when it comes to sharing knowledge, the more mechanisms a company deploys, the better.

Spotify relies on what it calls chapters, guilds, and agile coaches. Every Spotify squad member is assigned to a chapter, which is usually organized around a single competency, such as graphical design or back-end development. Chapter members meet to discuss issues and ideas specific to their roles, which leads to more coherent technical decisions. Guilds bring together people with common interests to share the latest discoveries in their domain and develop specialized skills. Agile coaches, who facilitate team dynamics, can recommend best practices observed in one squad to others.

Other companies rely on weekly stand-up meetings where teams inform each other — and other people in the company — about their deliverables and learning. And many companies provide collaboration tools for communicating within and across companies. These tools are designed to provide transparency and move away from command-and-control approaches that attempt to distribute information only on a need-to-know basis. Although initial attempts to provide transparency can lead to concerns of information overload, people on empowered teams eventually learn which people and tools are the most valuable — and therefore which to pay the most attention to.

Learning How to Architect Your Business

Most companies are still learning how to architect for efficiency. To do so, they must design people, processes, systems, and data that discipline their core operations.

As they learn to architect for efficiency, however, companies must also start learning how to architect for agility. This means designing empowered teams, as well as the systems, data, and processes that ensure the synchronization of individual teams' efforts. These business architecture efforts will ultimately allow for rapid delivery of integrated customer solutions.

About the Author

Jeanne Ross is principal research scientist for MIT's Center for Information Systems Research. Follow CISR on @mit_cisr.

What the Military Can Teach Organizations About Agility

DAVID GILLESPIE

U.S. armed services are experimenting with ways to make faster — and smarter — decisions, and business should take note.

Organizations across the world are facing a crisis of agility. Confronted with a volatile and highly unpredictable external environment, they are being forced to adapt at a speed for which they simply are not built. The larger and more established the company, the more it seems to struggle to escape organizational gravity.

While companies often look to technology companies for inspiration on how to adjust, large organizations might

consider using the U.S. military as an example. Once bastions of command-and-control management style, modern military institutions such as the U.S. Marine Corps are at the forefront of thinking about organizational and leadership agility, mastering how to evolve at "clock speed."

Overcoming the military's traditional hierarchical model has been essential in an era of digitally enabled terrorism. Today's military thinking now emphasizes the kind of innovation necessary to move its leadership efficiently through the four decision cycles of *observe*, *orient*, *decide*, and *act* in order to respond quickly to sudden external threats.

These same kinds of practices can make a big difference in helping any company respond to threats with greater agility, when applied as part of a wider program of change. Here are three specific practices that organizations should borrow from military leadership:

Practice 1

Leaders should focus on decisions only they can make. "I was most effective when I supervised processes," wrote retired U.S. Army Gen. Stanley McChrystal in his book *Team of Teams: New Rules of Engagement for a Complex*

World (Portfolio/Penguin, 2015). "Individuals and teams closest to the problem, armed with unprecedented levels of insights from across the network, offer the best ability to decide and act decisively."

This recommendation might sound obvious, but it makes for a far-reaching and powerful strategy: To be successful, leaders must resist the temptation to dive in to "solve" a problem that should have someone else's name on it. Leaders should be absolutely clear about what decisions only they can make and push all other decisions as far down the organization as possible.

Managers will be surprised at how few decisions only they can make. Those decisions include choosing their direct team, authorizing spending over a certain limit, and updating the Board or regulators. Most other decisions can be made elsewhere in the organization, and preferably as close as possible to the front line — the customer, in the corporate context.

Most companies, if they were to review their decision-making structure, would find too many situations where committees rather than individuals are tasked with gathering information, weighing options, and making decisions. This is a potential blocker of organizational agility. As a general rule, organizations should reduce decisions by committee and increase decisions by the individuals and teams who are closest to a particular issue.

Highlighting opportunities for a broader constituency of people to be decision-makers reduces reliance on bureaucratic hierarchies that depend solely on the most senior leaders for direction. Over time, people and teams will be empowered to be responsible for decisions within

their authority, and will act rather than automatically escalate everything to senior leadership.

Practice 2

Leaders should establish "commander's intent." The origins of this principle of U.S. military thinking stretch back to the 18th century. In its modern incarnation, "commander's intent" is a military leader's clarity of the end goal. It means focusing on the "why" and "what," while leaving the "how" to those closer to the front line. It means not micromanaging. As U.S. Army Gen. George S. Patton is quoted as saying, "Never tell people how to do things. Tell them what to do, and they will surprise you with their ingenuity."

In 2012, Gen. Martin Dempsey, then the chairman of the U.S. Joint Chiefs of Staff, published the Mission Command White Paper that instituted the modern-day commander's intent as a core principle "to empower agile and adaptive leaders." The goal was to infuse a bias for action, recognizing that only commanders could create the culture required for success. The vision was to develop a leadership doctrine that would be principles-based, rather than rules-based.

The six principles of Mission Command are to provide a clear commander's intent; build cohesive teams through mutual trust; create shared understanding; exercise disciplined initiative; use mission orders; and accept prudent risk. But it is commander's intent that is at the heart of this form of leadership, calling for a culture of disciplined initiative and risk-taking.

This idea is a world apart from what exists in many companies. Most frontline employees are instructed to

follow "one best way" rules, with organizations checking to ensure that employees have followed rules to the letter.

Examples of successful principles-based leadership are highly agile organizations such as Amazon.com Inc. and Netflix Inc., which are guided by Amazon's Leadership Principles and Netflix's Principles of Chaos Engineering, respectively. These organizations work with a speed and an experimental agility that traditional command-and-control organizations can only dream of. It's one reason why Amazon could deliver something as significant as Amazon Prime in only 111 days.

These companies recognized early in their development that training people to follow prescribed sets of rules can reduce the speed, dexterity, and adaptability of the organization over time. People in rules-based organizations, when faced with unexpected situations, wait for the chain of command to produce an answer. Those in principles-based companies solve problems themselves. When corporate leaders invest the time to articulate the "why" and "what" of target outcomes and convey a big-picture strategy in simple language, people become empowered to take the initiative to innovate.

Practice 3

Leaders should find a "directed telescope." A directed telescope is a concept that dates back to the 19th century. It means using a small number of trusted officers as the leader's eyes and ears in the field.

"Environmental dexterity" — meaning being actively receptive and responsive to new and changing external conditions — is critical for leaders of any organization. Keeping in close contact with the outside world from the perspective of employees, customers, suppliers, board members, and other stakeholders, becomes increasingly critical in a world where the pace of change seems to be ever-increasing.

In the corporate world, a directed telescope pipeline of information can come in the form of senior advisors, external coaches, or trusted colleagues at any level from inside the organization — anyone who can provide the manager with greater awareness and environmental insight. With candid feedback on what is happening on the ground, managers can improve their own agility and the agility of their organizations as a whole.

The Challenge: Overcoming Deep-Rooted Practices

Increasing leadership agility and, in turn, the agility of the wider organization is difficult. It requires changes to ingrained mindsets and behaviors of senior individuals who have been successful in their careers. Managers who recognize the need for organizational change usually fail to see *themselves* as one of the things that needs changing. They may set the goal of increased agility for the organization, but then neglect to establish a feedback loop for insight into how their own leadership behaviors may be interfering with achieving that goal.

Military leaders would be the first to say that their institutions can represent both the very best and the very worst in terms of agility, and that they don't necessarily have the full answer to the challenge. They do, however, invest significant resources in discussing the question and experimenting with innovative solutions. Business organizations need to adapt a similarly urgent rethink, in particular on the mindset and behaviors required for sustained organizational agility.

About the Author

David Gillespie is a partner in Oliver Wyman's
Organizational Effectiveness practice in London.

MITSloan
Management Review

PDFs • Reprints ▪ Permission to Copy ▪ Back Issues

Articles published in *MIT Sloan Management Review* are copyrighted by the Massachusetts Institute of Technology unless otherwise specified at the end of an article.

MIT Sloan Management Review articles, permissions, and back issues can be purchased on our website: **sloanreview.mit.edu,** or you may order through our Business Service Center (9 a.m. - 5 p.m. ET) at the phone numbers listed below. Paper reprints are available in quantities of 250 or more.

To reproduce or transmit one or more *MIT Sloan Management Review* articles by electronic or mechanical means (including photocopying or archiving in any information storage or retrieval system) **requires written permission.**

To request permission, use our website:
sloanreview.mit.edu
or
Email: **smr-help@mit.edu**
Call (US and International): 617-253-7170
Fax: 617-258-9739

Posting of full-text *MIT SMR* articles on publicly accessible Internet sites is prohibited. To obtain permission to post articles on secure and/or password-protected intranet sites, email your request to **smr-help@mit.edu**.