

FIVE SECRETS TO SCALING UP AGILE

By Kaj Burchardi, Peter Hildebrandt, Erik Lenhard, Jérôme Moreau, and Benjamin Rehberg

COMPANIES ACROSS MANY INDUSTRIES are struggling with the transition to Agile, a fast, iterative software-development method. Too many companies have the appearance of Agile—with, for example, hastily converted, brightly colored meeting rooms and daily stand-up meetings—but they achieve little of tangible impact.

At software start-ups in which Agile is commonly used, the development team is at the heart of the business, so buy-in, sustained commitment, and collaboration come fairly naturally. It is not easy, however, to integrate self-directed, cross-functional Agile teams into the existing hierarchy of large companies.

Some large companies, however, are figuring out how to make Agile work. Rather than impose its specific methodologies, they apply its general principles, paying special attention to the integration of Agile teams into the rest of the organization.

When large companies get Agile right, the results can be stunning. Productivity can

improve by a factor of three. Employee engagement, measured in quantitative surveys, increases dramatically too. New product features can be released within weeks or months rather than quarters or years. Rates of innovation rise, while the number of defects and do-overs declines. In the first year after going Agile, one bank's development team increased the value delivered per dollar spent by 50%, simultaneously cutting development time in half and improving employee engagement by one-third.

As the quality of software rises and the responsiveness of processes improves, some companies are applying Agile principles to activities other than software development. For such companies, Agile can become a journey of continuous improvement.

Why Agile?

Agile grew out of a desire to improve traditional methods of software development. Customarily, software has been developed sequentially, with the waterfall serving as a rough metaphor for its progression. Sepa-

rate groups conceive, design, build, test, put into operation, and maintain software, each group waiting for the preceding group to complete its work. The method is inefficient. In many cases, participants spend more time sitting in meetings and managing handoffs across organizational boundaries than writing and testing code. According to the often-cited *The Chaos Report*, less than 10% of large software projects come in on time and on budget.¹

The waterfall method comes from engineering, but writing many types of software is different from building a bridge. A river doesn't change its course, but software users have frequently changing and unpredictable needs. Consequently, Agile relies on bringing together many different points of view and supporting back-and-forth dialogue between developers and business executives.

Many forms of Agile have been developed, but at its heart, Agile is a set of beliefs. It is *iterative, empirical, cross-functional, focused, and continually improving*.

- **Iterative.** Agile is based on doing things repeatedly until you get them right. Short iterations mean that teams can change direction and react quickly. Progress remains visible and predictable because development happens in short sprints. Delivery risk declines progressively.
- **Empirical.** Agile teams rely less on the plans, estimates, and assumptions common to waterfall methods and more on A/B testing and other real-time metrics generated by end users. One of the many virtues of sprints is that they produce empirical feedback quickly, allowing teams to self-correct. Agile teams also measure and track their activities closely.
- **Cross-functional.** Agile teams have members of such relevant functions as business, marketing, development, and, in some industries, risk management working closely together in order to facilitate early and frequent feedback

from business executives and customers. All the members of the team have specific roles and responsibilities.

- **Focused.** Agile teams are fully accountable. They do not work on several projects simultaneously; nor do they leave a project once their specific duty is done. In for the duration, they develop a sense of accountability.
- **Continually Improving.** Agile software is a work in progress, with constant updates and experimentation aimed at satisfying customers.

Putting the Agile set of beliefs into practice can be difficult in large companies, given layers of processes and structures, such as HR, finance, and legal functions. Rather than viewing Agile as yet another new process, companies should integrate Agile values into their own software-development organization and culture, making reasonable modifications when necessary. (See the sidebar, “The Secret Sauce: Making Agile Work.”)

There are five secrets of success for large-scale Agile transformations.

It Starts at the Top

Transformative change requires support from the top. Senior leaders need to be actively involved in fundamental decisions about the business purpose of going Agile and the cultural barriers and root causes that might stand in the way of success. Without this commitment, legacy approaches to, for example, capital allocation, HR processes, and portfolio management will doom Agile. That's why business—not just tech—leaders must be accountable.

Agile transformations are different from other transformations: leaders must mobilize management to march in an unfamiliar new direction. The fast pace and cross-functionality of Agile can put many executives out of their comfort zone. Without strong and steady support from the top, many executives and team members revert to the norm. The CEO of a large European

THE SECRET SAUCE

Making Agile Work

There are several best practices that help activate the Agile set of beliefs at large companies. These practices—which embrace iterative, empirical, cross-functional, focused, and continually improving approaches—accommodate the realities of large organizations while staying true to Agile principles.

- **Iterative.** Agile teams complete manageable chunks of work—and produce a prototype—within fixed time periods. On the basis of feedback on the prototype, the team moves forward to a new set of tasks. The technical environments of large companies may not easily permit teams to operate in the two-week sprints customarily used in Agile, so many of them have stretched the sprints to intervals of four to six weeks.
- **Empirical.** Testing, a cornerstone of the Agile approach, ensures that software quality remains high and development activities are run efficiently. Large companies, especially those new to Agile, may not have invested heavily in testing tools. But as long as they are simultaneously building the business case for making these investments, corporations can forego some of the rigorous testing conducted by true Agile organizations.
- **Cross-functional.** Ideally, teams should not violate the “pizza box rule,” which restricts team membership to the number of people who can eat a single pie. The idea is to limit membership to those individuals who possess essential and complementary skills so that the team can accomplish real work. However, this rule can limit the ability of large companies to have the right experts on a team. The rule may, therefore, be loosened as long as all members are fully on board—part-timers need not apply—and contributing, not delegating.
- **Focused.** The single most important element of a functional Agile team is the “product owner,” a single executive who is empowered to make decisions about scope, timing, allocation of budget, and product features. In a pure form of Agile, the owner does need to consult a steering group or governance body. In large companies, however, this focus may be shared by two or three executives, such as a product manager and a business analyst or expert and possibly a “product executive.”
- **Continually Improving.** Agile teams rely on retrospectives, obstacle removal processes, and scrum masters to continually identify opportunities to enhance productivity by tweaking and tuning their environment and way of working. The specific methodologies are less important than the commitment to view the creation of software as an ongoing—not fixed—and organic process.

bank told us that he wants his organization to operate as a technology company that deals with financial services products.

To Fly, You Need Pilots

In a large organization, Agile pilots are necessary in order to determine whether Agile

will work there and whether the organization will accept Agile principles. Pilots are critical to a company’s making the necessary adaptations to Agile.

For example, in a scrum, a single product owner takes responsibility for managing the relationship and interactions between

developers and customers. This role requires a careful mix of technical and business skills. Companies may need to have even two or three people collectively serving in that role until the organization develops people who have the required multi-functional skills.

Likewise, it might be difficult to fully implement iterative development in all instances, but frequent feedback between developers and business executives ought to be the norm.

Staged rollouts in waves create momentum by building relevant capabilities and ensure that Agile principles and culture are embedded across the organization. (See Exhibit 1.)

Managing the Tipping Point

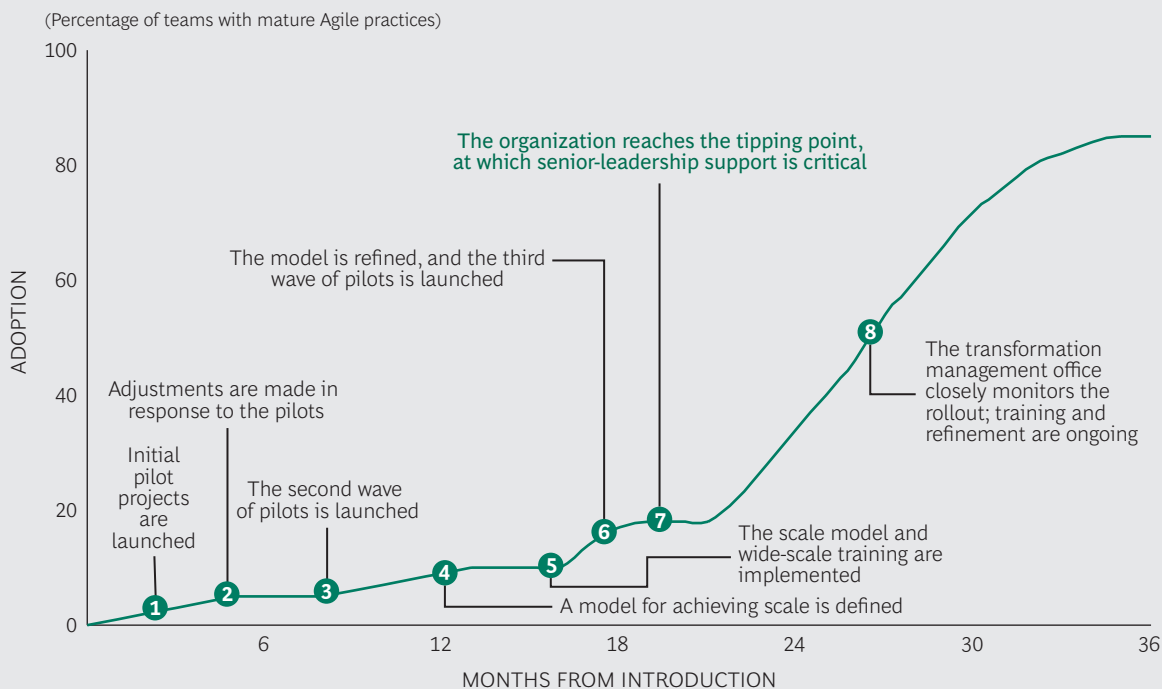
The pilot phase is followed by steps that must be executed with some delicacy to avoid unnecessary tension: it's time to scale up Agile in an organization that may be theoretically willing to accept it but, practically, is challenged to do so.

HR processes, such as performance management, may not be set up to handle fully dedicated cross-functional teams where team—not individual—results matter most. Agile's flexibility will almost certainly strain budgeting processes even if Agile is ultimately less costly than traditional development activities. An organization's IT infrastructure may not be set up to accommodate continual integration and deployment because of lengthy provisioning times. Furthermore, traditional development teams may be resentful, and certain activities may be outsourced.

These are all real technology and organizational concerns that will not resolve themselves on their own. Executives must actively manage the integration, and the enterprise almost certainly will have to invest in training and development to encourage the right culture and behaviors.

Several successful approaches exist for scaling up Agile within organizations. At one extreme, the music-streaming service Spotify has fundamentally changed its organization structure. The company's product-

EXHIBIT 1 | Agile Transformations Take Time



Sources: BCG case experience and analysis.

delivery organization is made up of squads, tribes, chapters, and guilds. The primary unit is the squad, a multidisciplinary team that works toward a shared purpose and is run by a product owner. Tribes are groups of squads that work on related areas. Chapters are groups of people with similar expertise across squads, and they form the line organization. Guilds are interest groups that anyone can join. (See Exhibit 2.) Other companies have simply overlaid cross-functional teams above existing hierarchy.

Measure, but Measure the Right Things

The ultimate goal of Agile is to improve the business. Therefore, the ultimate measurement should relate to business performance. If the goal of a bank’s Agile project is to reduce the dropout rate in credit card applications, then the dropout rate should be the most important metric. But in order to improve the business, companies also need to track software reliability, security, complexity, and size.

That’s where software measurement tools enter the picture. These tools allow companies to demonstrate empirically the pro-

ductivity and quality improvement of Agile development and the overall performance of Agile teams.

Never Stop

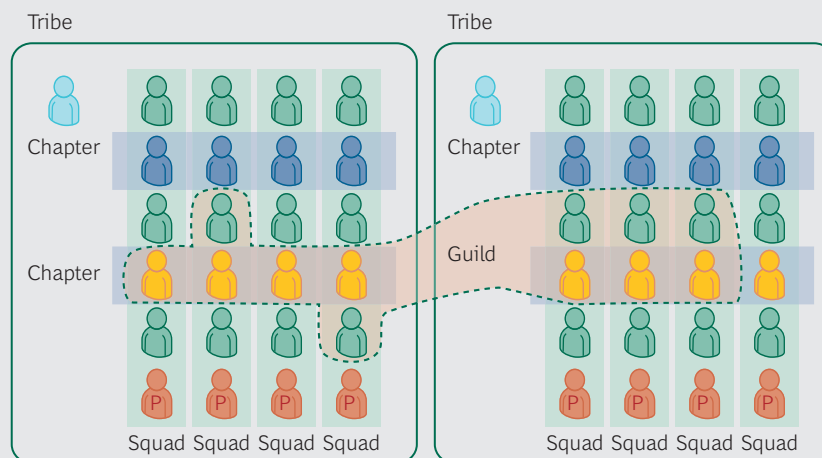
Agile development is an exercise in continuous improvement. It is not a one-off exercise. Agile requires constant monitoring to ensure proper functioning. Companies need to take steps to bake the Agile principles into the organization. There are many ways to ensure that Agile endures. Many companies, for example, create teams consisting of the leaders of each Agile project, and they share best practices.

AT ITS HEART, Agile is about creating the right context in which your people—specifically your developers—can do their best work. It is often thought of as a method for writing software, but ultimately, it is a way to run and continually improve your business.

NOTE

1. The Standish Group International, *The Chaos Report*, 1995, <http://www.csus.edu/indiv/v/velianitis/161/ChaosReport.pdf>.

EXHIBIT 2 | Organizing the Spotify Way



SQUAD

- The primary working unit
- Self-organizing and autonomous
- Long-term mission

TRIBE

- A collection of squads that work on related areas to solve a specific business problem
- Tribes support squads

CHAPTER

- A group of people with similar competencies across squads
- Responsibility for personnel development and training

GUILD

- An interest group that anyone can join

Source: Spotify.

About the Authors

Kaj Burchardi is a managing director in the New York office of Platinion, a subsidiary of The Boston Consulting Group that provides IT consulting services. He specializes in the development of strategic IT concepts and solutions, IT architecture, and digital transformation, especially in the financial services industry. You may contact him by e-mail at burchardi.kaj@bcgplatinion.com.

Peter Hildebrandt is a principal in the firm's Los Angeles office who specializes in technology and operational issues in the financial services industry. You may contact him by e-mail at hildebrandt.peter@bcg.com.

Erik Lenhard is a principal in BCG's Munich office and a core member of the Technology, Media & Telecommunications practice and Technology Advantage practice. He is a topic expert in digital transformation, strategy information management, big data, and Agile software development. You may contact him by e-mail at lenhard.erik@bcg.com.

Jérôme Moreau is a principal in the firm's London office and a core member of the Technology Advantage practice who specializes in digitization, technology transformation, and Agile development. Before joining BCG, he worked for eight years in the automotive sector as the leader of an Agile task force. You may contact him by e-mail at moreau.jerome@bcg.com.

Benjamin Rehberg is a partner and managing director in the firm's New York office and the global leader of the software and Agile transformation topic. He specializes in technology and operational issues in the financial services industry. You may contact him by e-mail at rehberg.benjamin@bcg.com.

Acknowledgments

The authors would like to acknowledge the insights and assistance of their BCG colleague Paul McNamara.

The Boston Consulting Group (BCG) is a global management consulting firm and the world's leading advisor on business strategy. We partner with clients from the private, public, and not-for-profit sectors in all regions to identify their highest-value opportunities, address their most critical challenges, and transform their enterprises. Our customized approach combines deep insight into the dynamics of companies and markets with close collaboration at all levels of the client organization. This ensures that our clients achieve sustainable competitive advantage, build more capable organizations, and secure lasting results. Founded in 1963, BCG is a private company with 82 offices in 46 countries. For more information, please visit bcg.com.

© The Boston Consulting Group, Inc. 2016.

All rights reserved.

2/16