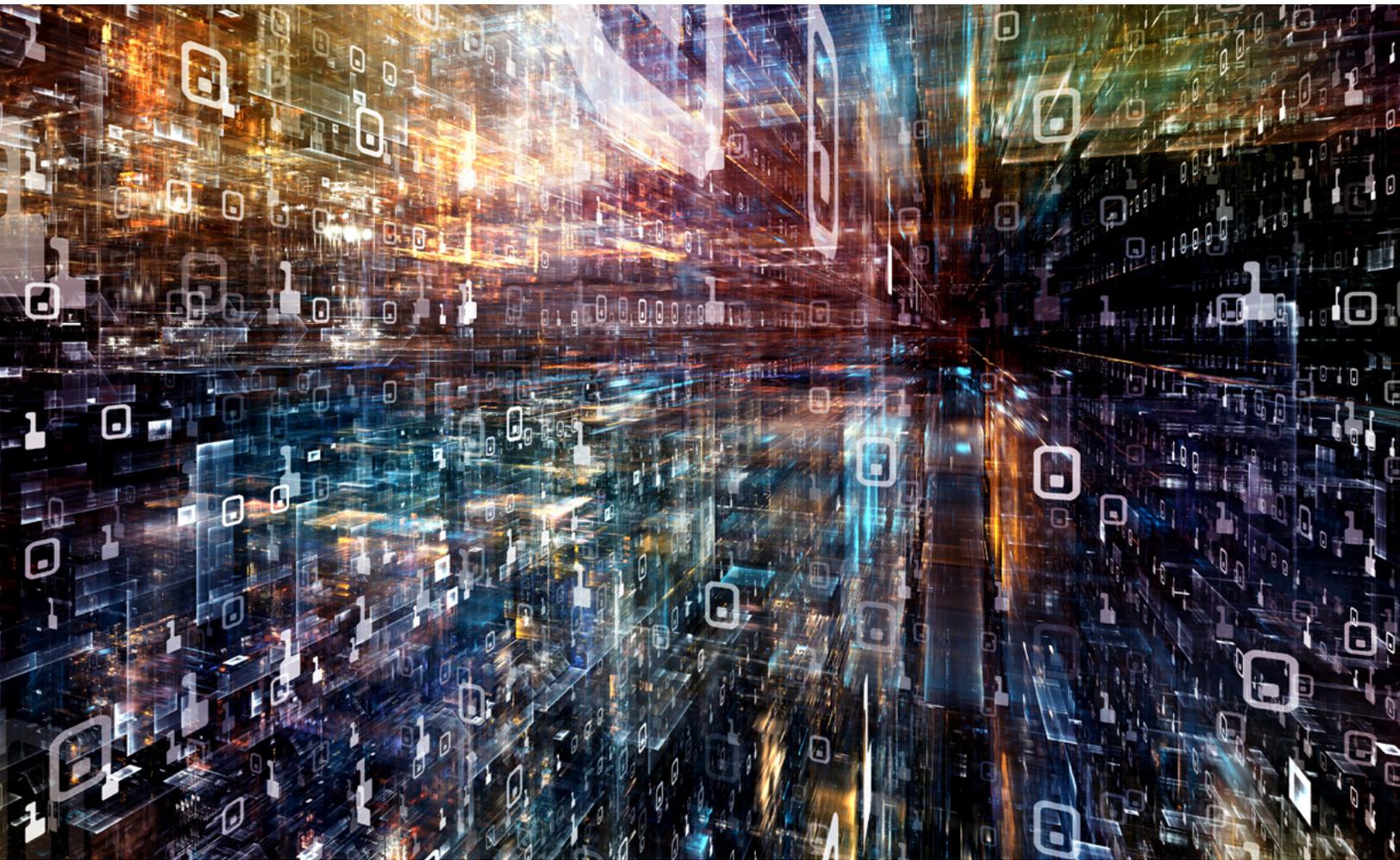


# The new frontier: Agile automation at scale

Large-scale automation of business processes requires a new development approach.

Federico Berruti, Geet Chandratre, and Zaid Rab



Across sectors, business processes are undergoing the most profound transformation since companies replaced paper files with electronic records. A new suite of technologies, including robotic process automation (RPA), smart workflows, and artificial-intelligence techniques such as machine learning, natural language tools, and cognitive agents, promises to radically improve efficiency while eliminating errors and reducing operational risk. Research by our colleagues at the McKinsey Global Institute suggests that, across industries, there is already the potential to automate more than 30 percent of the tasks that make up 60 percent of today's jobs. In finance and insurance, for example, workers spend more than half their time collecting and processing data, tasks that are eminently suitable for automation using techniques that are already available today.

Many companies have identified significant opportunities to apply automation, and the results of pilot projects and technology demonstrators have been encouraging. So far, however, most have struggled to capture the full potential these new approaches by applying them at scale across their operations.

There are multiple reasons why implementing automation is challenging. Some of the technologies involved are still relatively immature, for example. Applying them outside a carefully controlled test environment can reveal unforeseen weaknesses and limitations. And with thousands of processes involving tens of thousands of employees, organizations find it difficult to build workable road maps for large-scale automation.

#### **The devil in the development detail**

Then there's the challenge of software development and implementation. Companies need to tailor and customize their chosen technologies so they work in the context of the

wider organization. And because automation involves significant changes to existing roles and tasks, they need to coordinate technology development within a wider change-management process.

As many organizations have already discovered, established software-development methodologies do not work well in this complex environment. The first to fail has been the traditional "waterfall" approach, in which analysis, specification, design, coding, and testing are conducted sequentially. Automation projects organized this way have been plagued by delays and cost overruns, as companies discover unexpected issues or limitations late in the project-development lifecycle. That can be a particular problem when efforts are centralized at the enterprise level. After a successful proof-of-concept project, for example, one mining company used the waterfall approach to automate an important back-office process. The company was ten weeks into implementation when it discovered that its infrastructure design couldn't be scaled up to handle the work. By the time it identified and fixed the problem, the project was already delayed by more than four months, causing costs to spiral.

Experiences like this are encouraging more companies to pursue agile development approaches in their automation projects. With its emphasis on tight-knit cross-functional teams, focused development efforts, and continual testing, agile has proved highly successful in addressing similar challenges in other areas of software development.

Yet applying agile to automation projects has brought its own challenges. That's because process automation differs from the development of a conventional software product in a number of significant ways.

Scrum, an agile methodology that leverages quick iterations to develop features, works by breaking a complex problem or feature down into discrete chunks or “stories.” Teams work in these chunks one at a time, focusing on quality and releasing software frequently as opposed to at the end of the project. In a conventional software product, that usually means that products start by offering a limited range of features, with new ones added over time. In process automation, however, it can be difficult to break a feature down in this way. The individual components within a process are often tightly coupled: it either works end-to-end or it doesn’t work at all.

In addition, the disruptive nature of process automation, which may involve significant changes to the roles and responsibilities of hundreds of employees, can make frequent release cycles unfeasible. Sometimes the incremental value captured by a single component is not enough to justify a release.

Then there is the issue of ownership. In scrum, there is a dedicated “product owner” who acts as the representative of the end customer, working closely with development teams to answer questions, prioritize work, and give feedback on prototypes. Process automation may span multiple business functions, units, and geographies, making it difficult to find an individual with the requisite knowledge and connections. And because automation is new, the most appropriate “process owner” within the organization may have little or no experience of working on software-development projects, let alone the fast-moving, intensely iterative agile environment.

### Agile automation at scale

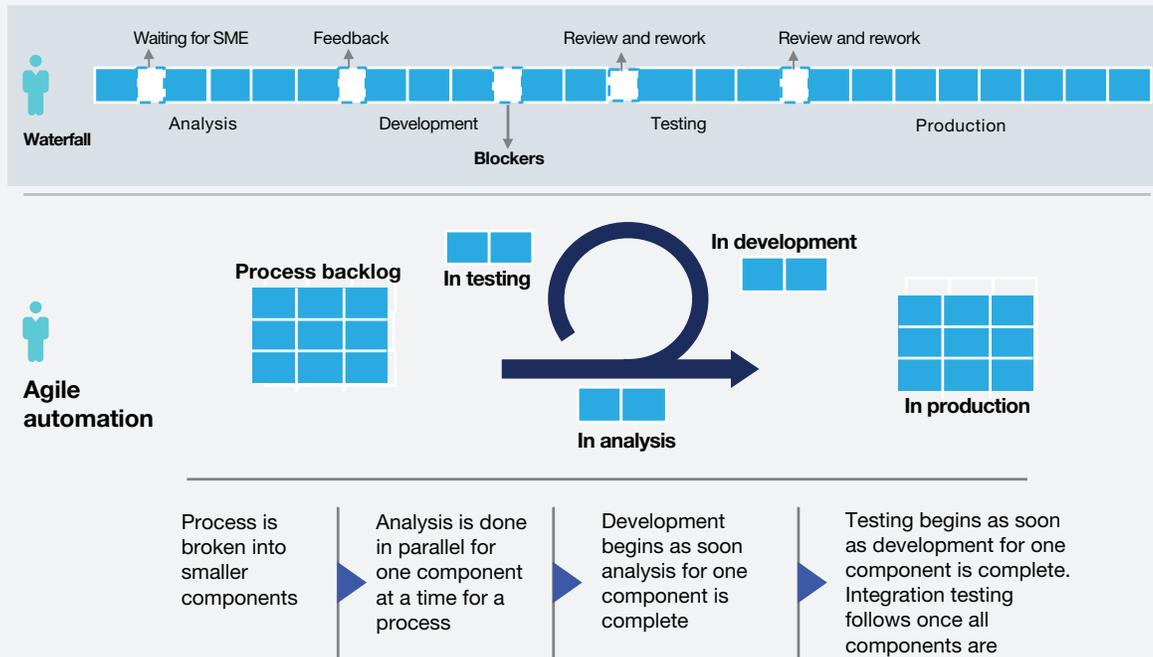
In response to these limitations, some companies are adapting and evolving the scrum framework for process automation. This “agile automation”

approach operates as a variant of scrum, with a few distinctive characteristics (exhibit).

- **Team structure.** Agile automation uses a flexible team or “pod” structure, which includes developers, testers, IT staff, and business stakeholders. Each pod is jointly led by a product owner, with expertise in the specific automation technology, and a subject-matter expert from the business, who provides essential business and domain knowledge.
- **Upfront design.** Agile automation involves an upfront effort that fully defines the process before development work begins in earnest. This work ensures that the automation project will integrate with the wider business and comply with regulatory requirements and other constraints. It also allows stakeholders in affected parts of the organization to prepare their people for coming change.
- **Trigger-driven stories.** To break the project down into addressable chunks, agile automation replaces conventional user stories with the concept of “trigger-driven stories.” This process identifies a “trigger event,” such as the availability of certain data or a user action; it then defines the actions required in response to that event, and the output to be produced. Using this approach allows teams to separate processes into manageable parts. Moreover, because the inputs and outcomes of each chunk are clearly defined, teams can work in parallel, accelerating development work.
- **Release management.** Agile automation decouples releases of prototype and production software. To minimize disruption to the wider organization, production releases are carried out to a controlled schedule that is tightly coordinated with the affected parts of the business. Prototypes are released more frequently into a dedicated

Exhibit

## Agile automation breaks analysis, development, and testing into parallel phases.



test environment where their performance is evaluated on representative data sets.

- **Program support.** Agile automation requires deep organizational change, as it requires companies to subject business-critical activities to unfamiliar technologies and new working methods—all at the same time. Especially in early stages, such efforts require significant support. Most organizations find it useful to establish a dedicated program office to provide expertise, establish good practices, and monitor the progress of the overall automation effort.

It is still early days for agile automation at scale, but the approach is already delivering encouraging results. After its early stumbles, the mining

company we described above rebuilt its automation efforts using agile principles. Its second attempt to roll out the project went twice as fast as the first and saved around 5,000 employee hours in its first year, thereby paying back its cost in less than 10 months.

Another company, this time in financial services, has built a large-scale agile capability to support its ambitious automation objectives. In a phased approach, the company first introduced agile techniques in its software-development teams. It then applied agile across teams to coordinate efforts and share best practices. Finally, the company persuaded its program leadership to adopt the approach as the standard for all automation efforts. Since the change, the company

has seen project-delivery time fall by around 30 percent and costs by 15 to 20 percent across six different business lines.



For large businesses, today's automation will reach its full potential only when it reaches full scale. A thoughtful application of agile concepts helps cut through the complexity for those willing to commit to change—not only in how they think about software, but in how they work every day. There's no time to wait. ■

**Federico Berruti** is a partner in McKinsey's Toronto office, where **Geet Chandratre** is a digital specialist and **Zaid Rab** is a digital analyst.

Copyright © 2018 McKinsey & Company.  
All rights reserved.