



Five Reasons why Agile Won't Scale
Without Automation

September 2011

Introduction

The Agile software development methodology is composed of a number of techniques designed to help deliver incremental releases of high-quality software more quickly than ever before. Although Agile adoption is proceeding steadily, many enterprises aren't yet fully converted to this approach. However, the trend lines are very clear: industry surveys have shown that the almost all new software development projects will be using Agile methodologies.

While Agile introduces many benefits, it also exponentially increases the pressure on the process of building, testing, and deploying software. Embracing the agile principle of "deliver working software frequently" means that the build, test and deploy processes are done frequently and in compressed timeframes, which can lead to many inefficiencies and poor software quality. So it is imperative for development organizations adopting agile to automate these processes. Without implementing this process automation, the great efficiencies and benefits which can come from Agile methodologies will not be realized.

In this paper, we present five distinct challenges that if left unaddressed will drastically reduce an organization's ability to gain the benefits promised by a move to Agile development methods. These challenges are:

- Manual procedures
- Performance inefficiencies
- Error-prone processes
- Disparate techniques
- Limited oversight

For each trait, we explain why it serves as an obstacle to reaping the benefits of a move to Agile methods, as well as how automation helps surmount the barrier to deliver better Agile results. The intended audience for this paper includes software development leaders as well as IT management.

Understanding Agile

Before we describe why Agile will not succeed without automation, let's first examine why this new software development approach has won over so many disciples.

Regardless of industry or enterprise, far too many software projects share the following unfortunate attributes:

- Late delivery
- Over budget
- Poor quality
- Mismatch with customer expectations

Despite decades of attempts and uncountable expenditures, centralized planning and rigid, uniform software development environments haven't been able to overcome these difficulties. Agile was born from the realization that a far better approach is to accept that unpredictability is natural, and that flexibility is indispensable.

In a nutshell, Agile means responding quickly to change. Agile software development teams keep pace with their targets, no matter where these targets may move. The first duty of any Agile project is to create a stable, shippable baseline product. Once that has been done, software developers continually add new functionality to their stable product and deliver incremental releases. This technique slashes delivery times from years or months down to weeks or even days. However, Agile is only successful if the development organization can keep pace.

Ensuring Agile Truly Works

From faster development to faster product applications

While Agile improves the tempo and efficiency of developing software, it introduces many challenges in the downstream software release process, which is composed of building, testing, and deploying software to production. As the fast pace of software development accelerates, the development organization also needs to conduct many more build-test-deploy cycles—often hundreds of times per week.

To solve these issues many organizations have adopted techniques like continuous integration and continuous delivery:

- **Continuous Integration.** A software development practice where members of a team integrate their work frequently: usually each person integrates at least once a day. In turn, this creates multiple integrations for each software team per day. Automated build and test scripts verify each instance of integration. This helps detect and correct errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly.
- **Continuous Delivery.** Although Continuous Integration helps synchronize developers' work more effectively, many organizations are still finding it hard to translate that approach into delivering software to the business more frequently. This challenge has often been referred to as “the last mile problem”: the ability to reduce cycle time between idea and usable software, where releasing to production becomes routine. To solve these issues, enterprises are also exploring “Continuous Delivery” methodologies, which extend the Continuous Integration methodology to full delivery of the software to the production environment. Continuous Delivery (also sometimes referred to as DevOps) is all about breaking down the barriers that stop organizations from effectively delivering working software to the business at the pace the business needs.

But moving to this brave new world requires some fundamental changes. While manual software release processes may have been sufficient in the past, they are no longer able to keep up in an Agile era. Automation is a must and at the heart of these new methodologies.

Let's examine five reasons why Agile demands automation.

Five Reasons Why Agile Won't Scale Without Automation

The behaviors we are about to portray greatly complicate the process of migrating to Agile. If ignored, each one increases risk and reduces the likelihood of a successful experience.

1. Manual Procedures

Building, testing, and deploying software by hand is tedious and inefficient. There is little commonality and repeatability among these tasks, especially when diverse, distributed teams are creating software. Communication workarounds such as email and team wikis are helpful, but don't simplify the daunting challenges of coordinating the scattered, often-dissimilar systems that perform the work.

Automation replaces ad-hoc, manual procedures with a collection of well-designed, reusable, and flexible templates. This is a much faster and more accurate way to build software. It also simplifies the job of synchronizing heterogeneous systems: a single build and test method will work across all platforms, freeing developers and testers to focus on their primary responsibilities.

2. Performance Inefficiencies

For complex modern applications, a single pass through the software build-test-deploy process can consume a substantial amount of time. Agile exacerbates this issue by requiring dozens to hundreds of builds each week, each of which may necessitate thousands of compilations. To make things worse, the number of mandatory tests necessary to validate all these compilations also grows exponentially.

By dividing these processes into smaller, more manageable segments, automation leverages the power of parallel processing to spread the work across multiple systems. These distributed platforms then work together to deliver results more quickly than ever before. Administrators can further optimize this process by tuning dependencies and minimizing network traffic.

3. Error-prone Processes

There are many explanations why manually initiated, slow-running build-test-deploy procedures are much more likely to experience failures. These include:

- Human-induced errors
- The rigors of harmonizing diverse, non-standardized environments
- Attempting to synchronize primitive tools in parallel operations
- The sheer number of builds required by Agile techniques

One of the best ways to diminish errors is to discover and correct them before they even occur. Automation makes it easy to validate the software build-test-deploy process up-front, using reproducible systems. Designers can configure and provision different build paths with a single click, thereby streamlining and speeding the essential task of setting up these important jobs.

4. Disparate Techniques

Even in the most rigid, hierarchical organizations, there's a good chance that each software development team has evolved its own routines, systems, and set of tools. By extension, this means that every piece of delivered software was created using a unique, non-repeatable process. To further complicate matters, today's software development teams are commonly dispersed throughout the world, and rarely share best practices.

Automation furnishes everyone with a centralized and standardized approach for sharing and re-using common procedures and workflows. This simplifies matters for users and administrators alike, and promotes widespread best practice dissemination. Well-designed supporting technology also recognizes that heterogeneity is inherent in modern software environments: it's simply unrealistic to expect everyone to standardize on a single platform, development environment, and so on.

5. Limited Insight

It's a sad fact that in most enterprises, the process of building, testing, and deploying software is a 'black box': very few people are able to explain exactly what occurs when software is delivered. This is a natural outgrowth of the 'one-off' manual processes that are so prevalent. There's limited capture of inputs, and even less analysis of the results of this process. Any automated logic that does exist is locked away in homegrown scripts.

Automation encourages and facilitates migration towards a meta-driven software delivery process. All results are captured and summarized at the appropriate granularity, whether at the executive, team, or job level. Keeping track of this critical information makes it much easier to identify and remedy bottlenecks, errors, and other complications that can sabotage any Agile initiative.

Conclusion

Agile is the future for many software development organizations. While Agile introduces many benefits, it also exponentially increases the pressure on the tasks of building, testing, and deploying software. A team practicing Agile techniques without any supporting technology won't be able to achieve the advantages Agile promises. To help them deliver working software quickly, these groups need to be supported by specialized automation.

Electric Cloud provides solutions that streamline the Agile software release process. Their products support the Continuous Integration and Continuous Delivery methodologies that help development teams deliver high-quality software rapidly. Electric Cloud automates and accelerates the application development (build/test/deploy) processes and their supporting IT resources (physical, virtual, or public/private cloud). The Electric Cloud solution increases the speed and efficiency of the application lifecycle by automating any process, eliminating manual errors, and maximizing utilization of cloud infrastructure. When Agile software development is mission-critical, Electric Cloud creates a competitive advantage through higher software quality, reduced costs, and increased developer productivity.

About Electric Cloud

Electric Cloud is the smart development cloud company. Electric Cloud bridges the gap between physical and virtual IT environments and development processes and tools to create smart development clouds. The company's patented and award-winning products help development organizations to speed time to market, boost developer productivity, and improve software while leveraging the operational efficiencies provided by virtualized/cloud infrastructures. Leading companies across a variety of industries, including financial services, ISVs, mobile devices, semiconductors and transactional websites rely on Electric Cloud's development cloud solutions. To learn more about Electric Cloud, visit www.electric-cloud.com.

Corporate Headquarters

Electric Cloud, Inc.

676 W. Maude Avenue, Sunnyvale, CA 94085 USA

Tel: 408.419.4300 **Fax:** 408.419.4399

info@electric-cloud.com

www.electric-cloud.com

Electric Cloud Asia Pacific

Suite 1905 Lippo Centre Tower 2

89 Queensway Admiralty

Hong Kong

Tel: +852.2918.8745

asia-info@electric-cloud.com

Electric Cloud Europe

1650 Arlington Business Park

Theale, Reading

Berkshire RG7 4SA United Kingdom

Tel: +44(0)1865.487177

europe.info@electric-cloud.com

Electric Cloud Japan KK

22F Shibuya Mark City West

1-12-1 Dogenzaka, Shibuya-ku

Tokyo 150-0043 Japan

Tel: +81.3.4360.5375

japan-info@electric-cloud.com

