



IDC TECHNOLOGY SPOTLIGHT

Agile Transformation in the Age of the 3rd Platform

March 2016

Adapted from *The Emergence of Microservices as a New Architectural Approach to Building New Software Systems* by AI Hilwa, IDC #256906

Sponsored by Pivotal

It has been said and repeated that software is eating the world. The heightened sense of industry disruption taking place today is largely enabled by software. A palpable shift in the understanding of the importance of software in recent years is rapidly morphing into a scramble to build software development competencies as enterprises cope with escalating disruption. But software is not easy. Software development is an intense discipline requiring deep and layered investment in technical skills. Software development is hard to learn and an order of magnitude or two harder to master. Software quality, productivity, and project success have persistently remained elusive. It is not enough to be able to build and deliver software. In order to build high-quality software in a timely fashion, organizations have to build a culture for effective software development processes. This Technology Spotlight explores evolving practices in successful software development, highlighting the importance of agile culture and its specific incarnation as offered by Pivotal Labs.

Introduction

Modern enterprises now realize that they need to continually innovate and transform. Enterprises are also discovering that they need new software development competencies to carry through this transformation unlike what they have been practicing. In the past five to 10 years, we have seen a variety of new techniques make inroads in the enterprise. For example, enabled by advancement in testing automation, continuous integration (CI) is reaching mainstream adoption in enterprise development shops. This has happened with or without specific agile methodologies. We are also seeing increasing adoption of test-driven development techniques, which prescribe that the tests be developed prior to the code to ensure complete coverage and provide a level of assurance that the requirements are captured with some fidelity.

Additionally, new approaches to application architecture, such as microservices, have emerged to leverage faster networks by building distributed systems with smaller code modules connected by well-defined APIs. Microservices architecture was evolved by tech industry pioneers such as Netflix, Twitter, and Amazon, which are increasingly leading the trends that eventually become mainstream enterprise IT practices. When combined with the capabilities of elastic clouds and modern DevOps practices that call for 12-factor application designs, treating infrastructure as code, and adopting immutable infrastructure supported by lightweight containers, the scale of transformation facing enterprise software development becomes alarmingly clear.

A Brief History of Enterprise Software

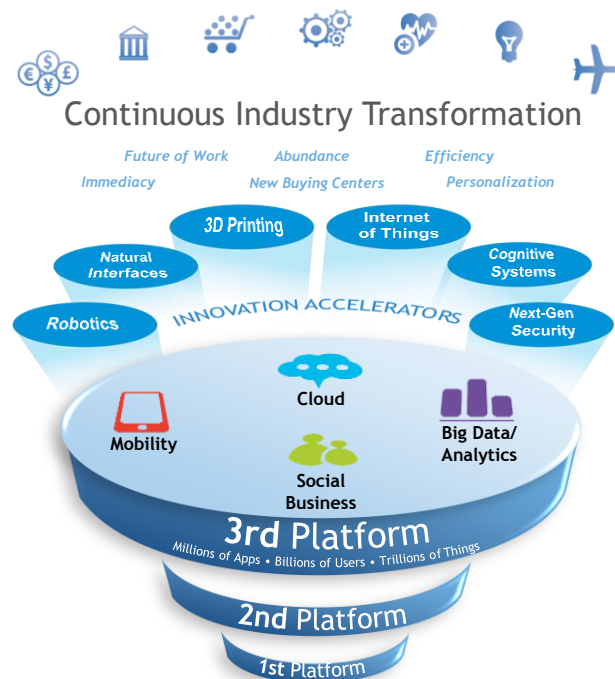
Organizations have been building software for over a half century, and developers have always been seen as important agents of organizational change. In the early days of computer automation, companies built core systems for manufacturing and distribution that became the basis of modern industry. Once such essential operational software was built, businesses turned their attention to horizontal functions inside organizations such as financial management, human resources, supply chain management, and sales force automation. Initially these systems were custom implemented inside of each major enterprise.

Over time, as technology markets matured and the software supply ecosystem thickened, organizations built out such systems through purchased commercial application packages, thus heralding the golden age of enterprise resource planning (ERP). Small to midsize companies were enabled by such applications packages to compete with larger companies on a more level playing field without even retaining an internal development shop. Most midsize companies were in the process of shifting to this buy-versus-build approach. For barely a moment, it seemed that computing was automating such back-office systems with such rapidity that we were reaching saturation and the job of the computer revolution was done. Then, in the mid-to-late 1990s, the Internet revolution kicked in and unleashed a new wave of enterprise transformation. Enterprise application development diversified from customizing and extending ERP, to building significant systems of engagement that targeted customers who were increasingly going online.

In 2007, the iPhone famously changed everything as customers began to interact with businesses at greater levels of frequency and granularity and in richer and more intimate ways enabled by cameras, geolocation, and ubiquitous wireless connectivity. A frenzy of building entertainment and customer engagement apps stimulated a parallel frenzy in the use of public cloud capabilities. IDC's iconic 3rd Platform concept captures this historical evolution (see Figure 1).

FIGURE 1

IDC's 3rd Platform



Source: IDC, 2016

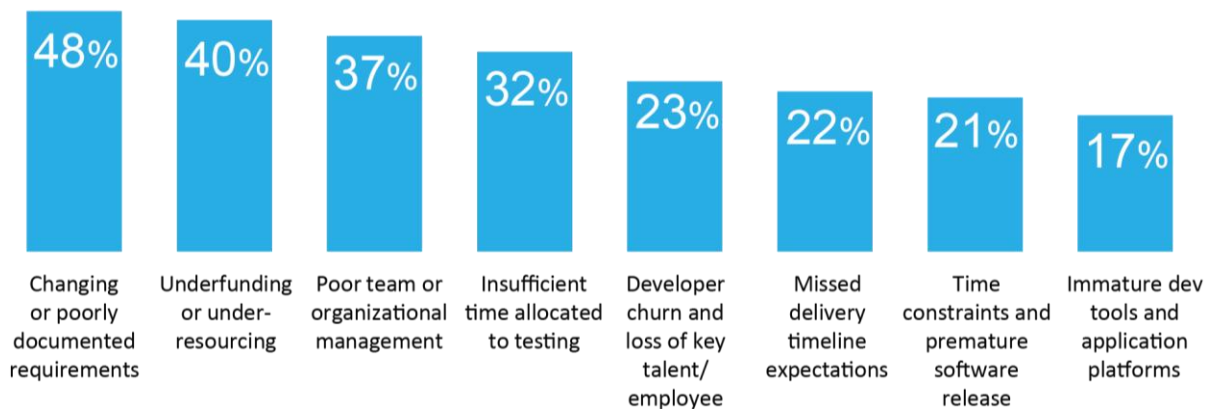
In recent years, mobility has broadened to encompass the reengineering agenda of employee apps making the building of mobile applications and their attendant new back ends a permanent required feature of the changing IT landscape. In short, digital transformation is becoming the new corporate agenda, and mobile and cloud developers have been moved to the center of the agenda. Having taken control of optimizing internal operational efficiencies, developers have become the key enablers for business innovation and growth. More than ever, developers are being marshalled to implement the transformation to what IDC today calls the 3rd Platform of technologies, which rests on the four pillars of mobile, cloud, social, and big data.

Agile to the Rescue

Competency in building software has emerged as an essential ingredient in the digital transformation agenda of every enterprise. However, the development of this competency is as hard as, or harder than, it ever was. IT systems rarely work on deadline (a recent example is the rollout of healthcare.gov in the United States), security of software systems has continued to be a major source of corporate embarrassment, and software projects continue to fail at alarming rates. See Figure 2 for reasons cited for software project failure in a recent IDC survey. In short, it has remained hard to deliver quality software at all levels, even as our critical dependency on software has multiplied.

FIGURE 2

Why Software Projects Fail



Source: IDC, 2016

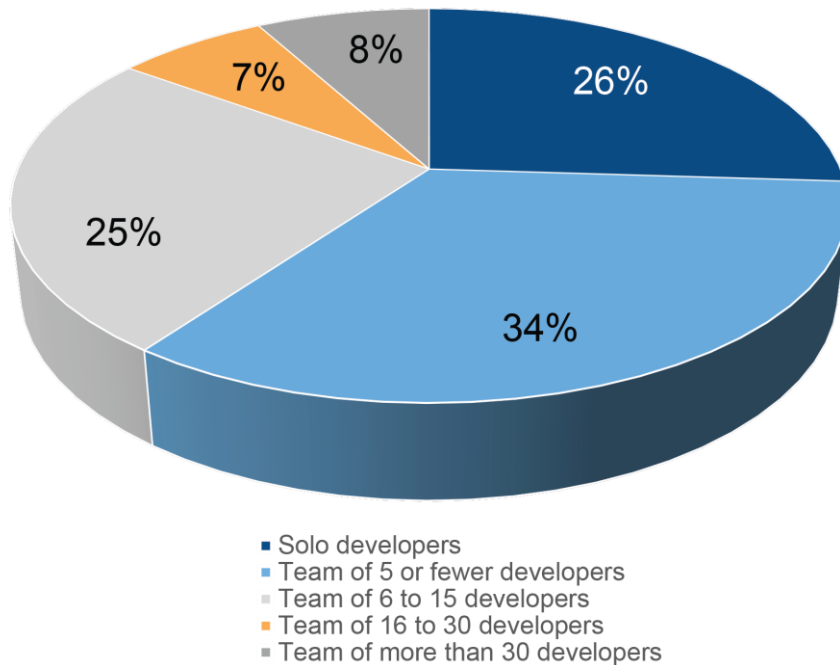
Since the mid-1990s, software engineering has been characterized by a shift in the approaches taken to manage large projects to what is now known as agile development methodologies. These methodologies, in specific popular incarnations such as Extreme Programming, generally focus on rapid iteration to integrate inevitably changing requirements. Continuous or frequent delivery of functioning code and inclusion of a broad circle of stakeholders in a "balanced product team" are principles that characterize agile methodologies. Specific variations such as pair programming, where two developers focus on developing higher-quality code on a single workstation, are interesting variants that have also been used successfully in an increasing number of organizations.

In the past decade and a half, agile development approaches have progressively taken hold in enterprises. Today it is rare to run into a development team that is not following one or more agile principles or techniques even if the team cannot identify a specific agile methodology as the basis of its practice. Research on modern application development practices has highlighted the continued

adoption of agile methodologies by organizations. IDC surveys show that organizations are increasingly focused on frequent delivery of working software through smaller, cross-functional software development teams. A recent IDC survey captured the significant trend toward smaller development teams (see Figure 3).

FIGURE 3

Size of Development Teams



n = 855

Source: IDC, 2016

Overall, organizations have received the message on how to organize agile software development teams. However, the delivery of successful software has remained relatively intractable, highlighting that much of the adoption of agile has happened at a superficial "going through the motions" level.

The Challenge of Agile Transformation

What has become better understood in recent years is that agile is a culture made up of values and habits as much as rules and processes. Adopting a true agile model is a matter of more than training or enforcing certain workflows in the development process. Enterprises not only must invest in building teams with the right skills or invest in project tools but also must take a holistic approach to team and project health to sustain the development of quality software. How to get to higher levels of maturity in agile development is not obvious. In other words, transforming software development teams into effectively functioning agile teams is an exercise of some complexity and challenge requiring its own set of practices.

Learning a new culture requires strong cross-functional organizational commitment reaching up to the highest level, persistent training to reinforce habits and practices, the passage of time and, in most cases, external coaching. The cultural transformation needs to include a deep look at people, process, and technology and should accompany the delivery of important bet-the-company landmark projects, thus accentuating the stakes at hand. To be effective, the coaching has to work through an integrated customer-driven product development approach that informs technical solution building and a specific view of technical architecture to achieve the dual goal of a cultural transformation and the delivery of a successful project that illustrates, motivates, and reinforces the commitment to ongoing change. This is the essential problem of agile transformation.

Pivotal Labs: From Solution to Culture

Company Background

Founded in 2013, Pivotal (Palo Alto, California) was launched to offer cloud platform and data analysis software and services, based upon products and services from co-owners EMC and VMware and more than \$100 million in backing from General Electric (GE), a 10% stakeholder. Pivotal brought together platform technologies that VMware and EMC had built or acquired together with Pivotal Labs, an agile software development company founded in 1989 that EMC acquired in 2012.

The company's main businesses include Pivotal Cloud Foundry, a PaaS for private and public cloud; Pivotal Big Data Suite, a set of data products for data storage and manipulation; and Pivotal Labs, which focuses on an "opinionated" agile approach to modern software development.

Pivotal Labs offers engagements with product designers and developers to help organizations define and prioritize software development in a focused, informed, and disciplined manner. Pivotal trains companies to use their rigorously designed tools and methodologies, such as data modeling, discovery and framing, paired programming, and test-driven development, to deliver products at a predictable, productive, and sustainable pace. The agility and flexibility of their frameworks speed up time to market and enable companies to react more swiftly to the dynamics of a changing market through continuous iteration.

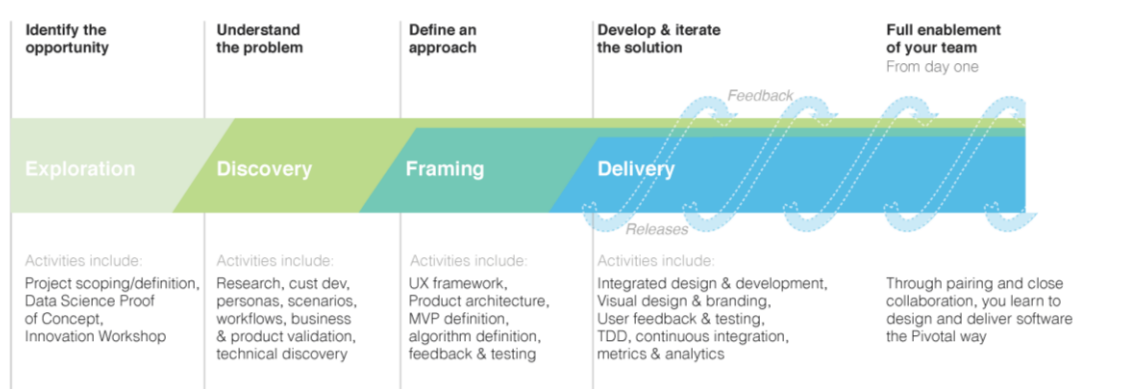
Pivotal also offers engagements with seasoned data scientists, who train companies to define, design, and execute on a data strategy and analytics road map. Pivotal's data scientists identify the best tools and environments for data aggregation and organizations and provide frameworks for how to access and use data and data modeling to provide more expansive perspectives in solving business problems. The result is letting the data uncover insights that drive action. Pivotal's development services and tools are integrated with Pivotal Cloud Foundry and Big Data Suite offerings.

The Pivotal Way

Pivotal Labs employs a derivation of Extreme Programming methodology (see Figure 4). Pivotal engineers, known as "Pivots," design and program jointly with clients in focused coaching. These cross-functional teams are onsite, in-person, colocated teams. The key goal early in the engagement is to develop a minimum viable product (MVP). The objective is to get to the MVP as early as possible to enable the testing and viability process.

FIGURE 4

Pivotal Labs' Approach



Source: Pivotal Labs, 2016

Process and Roles

Iterative Design, Development, Test, and Delivery of Business Value

Agile methodologies use several techniques to deliver working code to stakeholders to ensure that code has value to the business and the customers. A product owner (or similar role) is responsible for defining the customer hypothesis and prioritizing requirements based on customer needs, giving higher priority to those with the most value. Each iteration results in continuous delivery of work that is validated by the customers throughout the product life cycle and can be viewed and touched by stakeholders, providing the basis for refinement of the requirements. A good agile initiative doesn't just deliver a working, viable product or application that meets requirements; the product gets better as work progresses. The iterative incremental nature of agile not only makes it adaptive to changing needs and requirements but also, just as important, mitigates risk. It is difficult for an agile project to go off the rails because work is highly visible and continually reviewed for appropriateness and quality.

Pivotal engagements are active learning and training experiences that introduce clients to a modern software development process. Clients work side by side with Pivots to define, develop, and ship an actual application. For Pivotal, the transformation that occurs during an engagement is not only about the digital transformation of the client but also about transforming their product development approach and application development practices. Soft skills and culture are transferred in these workshops.

Almost all engagements at Pivotal start by bringing the team in for an exploration session involving an innovation workshop and problem-solving exercises designed to get a deep understanding of the customers and their needs as well as the business issues. The process of discovery and framing (few days to a few weeks) happens before any development or coding takes place and is used to socialize understanding of the business problem and define the project approach. The Pivotal delivery process is 100% test-driven development (TDD) to achieve 100% test coverage of the code. Development of test cases is done in conjunction with code development.

The process employed by Pivotal Labs relies on teamwork — specifically an approach known as paired programming in which two team members work as a pair on a single piece of code. Pivotal employs self-organizing teams that emphasize joint ownership and shared responsibility. According to Pivotal personnel, this is "where magic happens." Balanced teams consist of multiple roles and stakeholders, including IT, functions outside of IT, data scientists, product managers, product designers, and software developers.

The Importance of Culture

At the risk of being recursive, high-performing agile organizations have agile cultures. The statement attributed to Peter Drucker —"Culture eats strategy for lunch" — describes the situation perfectly; organizations that introduce agile strategies without first creating an agile culture base will struggle or fail. Some key constituent parts of agile culture are making decisions based on customer values, trust and open communications, accountability, horizontal organizational and power structures, collaboration, the ability to deal with ambiguity and, perhaps most important, curiosity. Agile people and organizations like to ask, "What if we did X?" CIOs and their managers need to treat agile as a transformation process — most IT organizations are not born agile, and developing an agile culture requires time and discipline.

Such a culture is engrained at Pivotal Labs. The workday is designed to deliver the sustained pace of a highly productive eight-hour workday. Because this level of productivity can be exhausting, Pivotal limits the time frames of exposure. Everyone is done by 6:00 p.m. Things that seem arbitrary or luxurious as practiced in development teams in modern technology industry, such as providing breakfast and meals, are actually about productivity. For example, providing breakfast synchronizes biological clocks and sugar/energy levels and ensures that all team members get to work on time and are available for the morning agile stand-up meeting, making it possible for pair programming to work. Teams collaborate better because they take breaks and have meals at the same time.

Challenges and Opportunities

Pivotal's approach to agile is powerful. Enterprises engaging with Pivotal should note the following:

- Agile culture transformation is difficult, takes time, and requires cross-functional organizational commitment at the highest levels. Engagements at the solution level do not always reach high enough to be able to effect comprehensive cultural transformation, especially in large companies. Additionally, the engagements may be too short-lived to do other than to create exposure, albeit in an intense fashion. Nevertheless, transformations to the level of a product team can serve to demonstrate what is possible and set in motion broader organizational change over time.
- Key aspects of the Pivotal way, such as pair programming, which account for much of the productivity and software quality gains achieved in Pivotal engagements, can seem too expensive for smaller organizations or teams. Having to send top developers to Pivotal HQ or remote offices for several weeks may be a difficult commitment that some enterprises are unable to make.
- Pivotal prides itself on following an "opinionated" approach toward application development that focuses on a set of technologies and practices that the company sees as proven to achieve results. Engaging with Pivotal is essentially a decision to follow its philosophy and set of recommended products. While the approach and technologies used have been effective, potential customers may have preferences for other technologies and approaches.

Conclusion

Properly implemented, agile can offer numerous benefits in complex and changing environments. Because work is done in short increments with continuous stakeholder involvement and communication, time to market can be shortened, and applications are more likely to truly meet user needs. Moreover, cross-functional teams bring broad business and functional insight needed to achieve lasting project success.

Pivotal is offering a suite of products and technologies that organizations can leverage to allow their developers, data scientists, and other stakeholders to come together and create valuable business solutions. Enterprises in the throes of digital transformation would do well to engage the coaching and solution-building capabilities of Pivotal Labs. The experience will inject the culture, habits, and practices developed and honed by the tech industry into enterprise development teams. The result is often a deep transformation in the way software is developed.

ABOUT THIS PUBLICATION

This publication was produced by IDC Custom Solutions. The opinion, analysis, and research results presented herein are drawn from more detailed research and analysis independently conducted and published by IDC, unless specific vendor sponsorship is noted. IDC Custom Solutions makes IDC content available in a wide range of formats for distribution by various companies. A license to distribute IDC content does not imply endorsement of or opinion about the licensee.

COPYRIGHT AND RESTRICTIONS

Any IDC information or reference to IDC that is to be used in advertising, press releases, or promotional materials requires prior written approval from IDC. For permission requests, contact the IDC Custom Solutions information line at 508-988-7610 or gms@idc.com. Translation and/or localization of this document require an additional license from IDC.

For more information on IDC, visit www.idc.com. For more information on IDC Custom Solutions, visit http://www.idc.com/prodserv/custom_solutions/index.jsp.

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com