Deep Dive

Digital transformation: The power of permanent agility

Transformation is hard. But when you build a platform for continuous change, putting new ideas into production gets a lot easier

BY ERIC KNORR AND DAN TYNAN

At last IT pros and developers can associate themselves directly with revenue generation rather than just answering management demands to reduce costs and increase efficiency.

As best we can tell, the phrase "digital transformation" sprung from the mind of a marketing copywriter in the late last century. For years it was just a nebulous catchphrase. But digital transformation has shown remarkable staying power, actually gaining currency over time and earning a real and important meaning in enterprise tech.

Transformation usually implies moving from one fixed state to another. Yet digital transformation has come to mean a journey from inflexible platforms, products, and workflows to a "permanently agile" condition. Getting there may involve the adoption of new programming, infrastructure, or IoT (Internet of things) advances and processes. But the goal is to create a platform for continuous experimentation and to establish mechanisms to measure results.

The most visible manifestation of digital transformation has been in customer-facing mobile and web applications that change constantly. Underlying them is an array of shiny new technologies and methodologies: cloud infrastructure, devops, microservices, big data analytics, and much more. This nucleus of adaptable enterprise tech connects to an expanding circle of mobile and IoT endpoints that continually gather actionable data.

Chasing digital value

Why the emphasis on customer-facing applications first? In part because a new generation of consumer-facing internet companies has opened fresh revenue streams with a vengeance not seen since the dot-com boom. The millennials who built those ventures have developed an abundance of exciting new frameworks, tools, platforms, and methodologies to create applications with unprecedented speed and agility.

Inside established enterprises, IT pros and developers are catching on to this trend: At last they can associate themselves directly with revenue generation rather than just answering management demands to reduce costs and increase efficiency. The success of such outbound efforts has caught the imagination of the C-suite. In a recent survey by Unisys and IDG Research, more than 70 percent of CXOs said they were committed to implementing a "digital business model."

That model extends beyond customer-facing applications. Most promising is making agility a permanent part of the custom software that defines a company's core business—from manufacturing software to collaborative design platforms to logistical systems. Here, digital transformation embraces physical "things," from cars to thermostats to industrial engines.

Take the software GE uses to run its rail freight business. The company collects velocity and location data from freight trains around the country. Using analytics to manage rail traffic and increase average train speed could potentially shave hundreds of millions of dollars off the cost of delivering goods.

That's clearly an example of delivering real business value, but as with any hot trend, claims of digital transformation have a habit of going over the top. Some established corporate giants seem to redefine themselves as "digital" mainly to impress Wall Street. In truth, particularly in organizations of enormous scale, digital transformation never occurs all at once.

Deep Dive

"You need to recognize that digital transformation is not accomplished in one massive effort, like it was in the ERP and CRM days," says Gerald C. Kane, associate professor of information systems at the Carroll School of Management at Boston College. "The companies that are doing it right are doing lots of little experiments, capitalizing on the ones that work, and killing off quickly the ones that don't."

The allure of transformation

The specific motivations behind transformation and continuous experimentation vary widely depending on the industry, business, and company culture. Nonetheless, some incentives are common to nearly all of them:

Faster time to market. The principal idea behind permanently agile platforms is the capability to build, test, and deploy applications or physical products faster—in some cases, magnitudes faster than before.

Higher-quality results. Increased communication with customers and measurement of customer behavior—along with close internal collaboration among stakeholders and practitioners—yield superior end products.

Increases in revenue. Software or physical products tuned to customers' needs and desires demonstrably increases revenue per customer, intensifies loyalty, and aids in the acquisition of new customers.

Greater reliability. The distributed architecture behind digital transformation often includes redundancies that guard against errors, downtime, or performance troughs to deliver a consistent user experience and avoid costly downtime.

Cost reduction. Transformation frequently involves tapping into existing services via cloud APIs or incorporating open source code, so that practitioners no longer need to build everything from scratch.

Best of all from an organizational standpoint may be the ability of a permanently agile enterprise to attract and retain the best talent, particularly if the transformative outcome includes effective internal collaboration. Modern platforms appeal to young, creative technologists—and all practitioners enjoy greater job satisfaction when tools and processes support the free exchange of insights, requirements, and challenges as part of the culture of digital transformation

Transformation essentials

Digital transformation does not stipulate a specific set of technologies or even necessarily methodologies—it simply requires the adoption tools, platforms, and processes to enable continuous experimentation and change. That said, enterprises that have undergone recent, successful transformation—along with the startup ventures that have inspired them—gravitate toward a common set of essentials:

Devops. A mashup of "development" and "operations," devops dictates that software developers should be empowered to provision their own environments, while operations should have the ability to automate continuous, reliable deployment at scale. Devops is the engine of delivering more and better software faster.

Service orientation. When big, monolithic applications or processes are broken down into services, each service can be scaled, maintained, and updated independently—and ultimately reassembled into new applications or processes. Many modern web and mobile apps are developed using microservices, the latest manifestation of service orientation.

Monitoring and analytics. A key aspect of digital transformation is collecting and acting on continuous feedback from customers, both explicit and implicit—the latter collected through monitoring interactions with web and mobile applications. Analytics and machine learning enable the recognition of patterns in clickstreams to inform continuous improvement.

Scalable infrastructure. The abstraction of hardware as software-defined compute, storage, and network resources—call it "the cloud" if you like—enables capacity to be applied where it's needed almost instantaneously. It establishes an all-purpose platform for a multitude of new applications that can scale on a dime.

Coordinated decentralization. A centralized, top-down IT structure can't possibly move fast enough to develop all solutions for internal

Deep Dive

stakeholders. Lines of business and departments that understand their customers best need to be empowered to make decisions independently, yet avoid missteps that result in redundancies, security vulnerabilities, or siloed systems.

Identity management. When groups are empowered to operate independently, and individuals in those groups may subscribe to various cloud services or tap multiple databases, identity management must be in place to control user access to resources based on established rights and restrictions.

Note that the permanent agility brought about by digital transformation goes beyond applications. An exciting aspect of the IoT, for example, is the ability to continually improve the functionality of products—from TVs to cars to manufacturing equipment—through ongoing software updates post sale.

At the same time, the IoT offers extraordinary new opportunities to collect and analyze data about how customers use products. For example, Ford has put remote sensors in 4 million cars to collect data about gas consumption and driving style, with an eye towards increasing fuel efficiency and avoiding equipment failure.

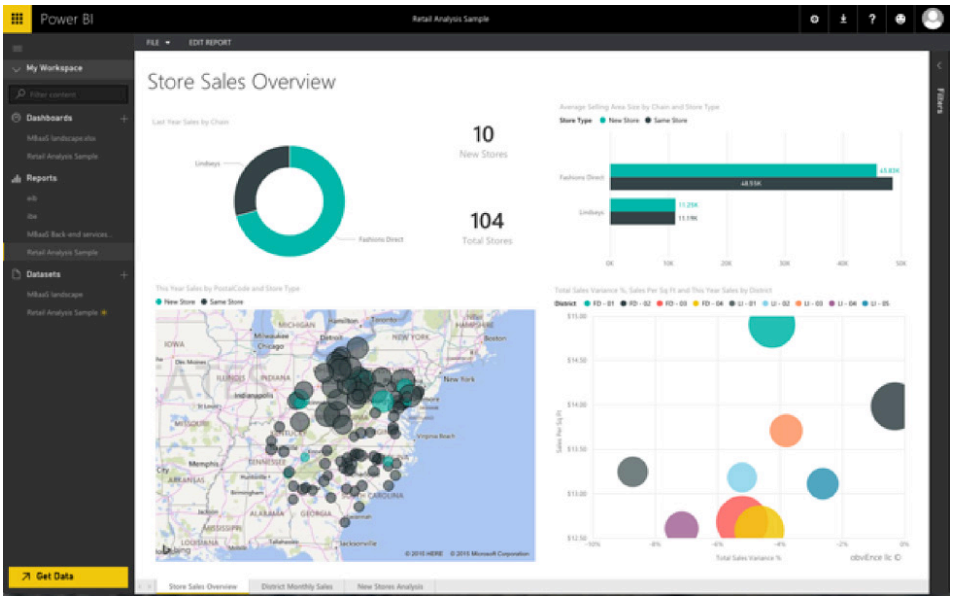
The permanent agility brought about by digital transformation goes beyond applications. An exciting aspect of the IoT, for example, is the ability to continually improve the functionality of products.

Insight through analytics

The signature aspect of permanent agility is the ability to modify, add, or replace functionality with unprecedented speed—but change should not be blind. Data continually collected from interactions with systems on the edge provides the raw material for analyses, which in turn guides the development of new and better solutions.

Again, this extends beyond improving customer-facing applications to optimizations in the real world. To take a simple example, supermarket chains such as Carrefour, Niemann Foods, and Woolworths are tracking Bluetooth signals from smartphones to measure the flow of customers through their aisles, so they can improve the effectiveness of store layouts, reduce lines at cash registers, and offer personalized coupons.

Much of the data collected from "systems of engagement" may be semi-structured: clickstreams, time-series data, event log files, mobile location data, and so on. A bundle of open source big data technologies centering on the distributed processing frameworks Spark and Hadoop operates on the semi-structured stuff, typically trans-



This Microsoft Power BI screen illustrates how data can be collected from multiple, diverse sources and combined into a single dashboard. Understanding such analytics is key to the continuous evolution of applications and products.

Deep Dive

forming the data so that traditional SQL analytics software can handle it. Once converted, it's often mashed up with existing SQL data such as transaction records, product data, pricing information, and so on to gain new insights.

Increasingly, business analysts rather than BI specialists are being given the analytics tools to plumb all that data and tease out the patterns. These include such BI tools as Tableau, Qlik Sense, or Microsoft Power BI, which make visualizing data easy. In addition, more and more enterprises are turning to the cloud versions of the whole analytics stack, where all that software has been pre-provisioned. A huge cloud advantage is the availability of new machine learning APIs offered by Amazon Web Services, Microsoft Azure, and Google Cloud, which can provide vital assistance in data pattern recognition.

The Holy Grail of all this is accurate predictive analytics. Take mobile and IoT data, combine it with information from the supply chain and sales funnel, and enterprises gain the ability to make business decisions based on things that haven't happened yet: Utilities use smart meter data to predict spikes in power usage. Hotels use historic trends to dynamically price rooms based on demand. Construction firms anticipate change orders and avoid work stoppages. Plus, most of the major e-commerce sites use predictive analytics to anticipate spikes and troughs in demand.

Capital One Bank (an InfoWorld 2015 Enterprise Architecture Award winner) was one of the first financial institutions to use analytics to predict what credit products a customer is most likely to buy, decrease customer acquisition costs, and increase retention rates. Marriott is using predictive analytics to determine optimal pricing for hotel rooms based on vacancy rates. General Electric has installed sensors inside its locomotive engines and uses analytics to predict when engines will require maintenance. The list goes on and on.

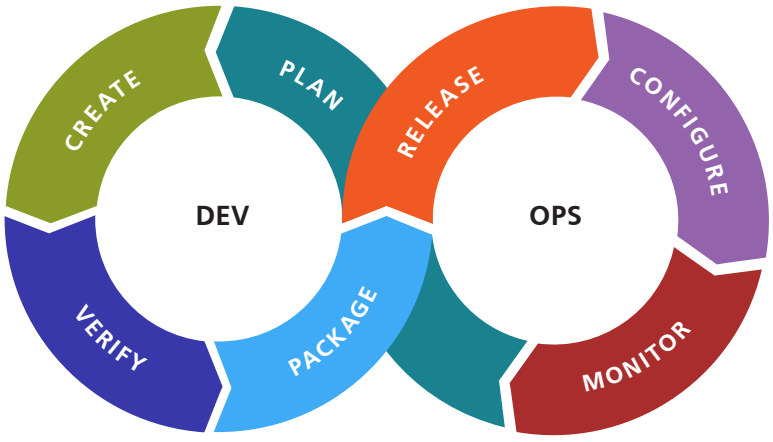
The Holy Grail of all this is accurate predictive analytics. Take mobile and IoT data, combine it with information from the supply chain and sales funnel, and enterprises gain the ability to make business decisions based on things that haven't happened yet.

Devops at the core

Behind those examples typically lies some combination of custom and open source software, fashioned into line-of-business applications or customer-facing apps—the latter now venturing into new territory, from fitness bands to home appliances. How that software is developed and delivered determines whether it can sustain products and solutions successfully over time.

No trend embodies digital transformation to a greater degree than devops, which encompasses the entire dev, test, and deployment pipeline. In the devops model, there's no such thing as "one and done" with software. Developers add improvements and new capabilities continually in small chunks of code, which are delivered through an automated pipeline to production.

Although the goal of devops is to shorten the time to market of software that meets or exceeds



Devops vastly increases the efficiency of software development, giving developers greater control and enabling operations to fully and safely automate deployment.

Deep Dive

expectations, devops has no direct effect on how developers write code. Instead, devops adds automation and streamlines workflows through the entire cycle, enabling developers to build, test, and deploy incrementally. It also allows stakeholders to review applications in progress, provide feedback, and change direction if necessary.

Devops vastly increases the efficiency of software development, giving developers greater control and enabling operations to fully and safely automate deployment.

These ideas are not new. In fact, they originate with agile development, a methodology first articulated in the Agile Manifesto 15 years ago. Agile development was conceived as an antidote to waterfall methodologies, which progress in linear fashion through a series of stages, such as feasibility, requirements, external design, program specifications, coding, testing, and finally production. After the requirements stage, stakeholders were cut out of the process, leaving them no choice but to accept whatever developers delivered—which often missed the mark.

With shorter cycles and ongoing stakeholder involvement, agile development promises vastly improved results. Only recently, however, has a constellation of technologies gathered to support devops effectively. Devops implementations vary considerably, but most include these components:

Cloud-based code repositories. Today's development teams are often widely distributed, which makes moving code repos to the cloud a no-brainer. GitHub and Bitbucket are the most popular cloud-based revision control and repository systems, both based on Linus Torvald's Git.

Continuous integration. In agile methodology, developers work on discrete parts of a project and check them into a shared repository several times a day, at which point an automated testing system checks everything in order to catch problems early. The Jenkins continuous integration platform for automating developer workflows is the clear leader in this space.

Continuous delivery. Popular PaaS products such as Cloud Foundry or OpenShift have expanded to embrace a greater portion of the devops pipeline, but their most important role is to provide a single runtime environment for both development and production. Both support Docker, which enables developers to package applications for deployment in Linux containers. Many shops develop their own PaaS-like solutions using configuration management or container management solutions such as Kubernetes, Mesos, or Swarm instead.

Configuration management. This is where devops began and where its emphasis remains. Tools such as Puppet, Chef, Ansible, and Salt automate the provisioning of infrastructure and applications with scripting environments and script libraries that either developers or operations can use to configure hundreds or even thousands of virtual machines to accommodate dev, test, and continuous delivery of applications. Configuration management tools can be used to manage the delivery and deployment process end to end.

Application monitoring. Just as customer feedback suggests how software should be improved, application monitoring gives developers and operations teams real-time insight into how well or poorly applications fare in production. Monitoring is a key part of ensuring that continuous delivery works, identifying problems that demand immediate attention. Popular solutions in this space include offerings by AppDynamics, New Relic, and Sumo Logic.

With shorter cycles and ongoing stakeholder involvement, **agile development promises vastly improved results** over waterfall methodologies.

It's important to remember that this is a simplified view and that devops workflows and toolchains vary widely from case to case. Many, for example, include dedicated tools for lean project management, unit testing, preproduction packaging, and release management.

In an interview with InfoWorld, devops guru Gene Kim shared the insight that whether or not operations used version control was the top predictor of devops success: "When you think about where more things can go wrong—is it in the code or in the environment—there are probably a hundred or a thousand times more configurable settings in the environment. … So you put ops and dev in the same version control repo where everything is reproducible."

The various mechanisms of devops fuel the dramatic increases in developer productivity necessary to keep an enterprise agile, as software spreads to play a role in nearly every aspect of

Deep Dive

business. The result, when managed properly, is at the heart of digital transformation.

Transforming the organization

Powered by devops, analytics, and intense collaboration among technologists and stakeholders, organizations that transform successfully make a regular practice of throwing things against the wall to see what sticks. They've adopted the Silicon Valley start-up mentality of "fail fast, fail often," followed by equally rapid iterations or pivots.

But you need to conduct the right kind of experiments and measure them in the right way, says Mike Mason, technology activist for ThoughtWorks, a global technology consultancy.

"Its all about the business value you manage to create," he says. "If you're just putting the wrong stuff into production faster, it doesn't actually help you with business value. If you take somebody like LinkedIn or Netflix or whoever, they are running dozens of experiments a day, putting small things into production and getting telemetry from the production usage of that feature or experiment and then figuring out which things they want to double-down on."

Many of these experiments will fail. That means upper management must be both risk tolerant and failure friendly, which requires both top-down vision and the willingness to stick to it, notes the Carroll School of Management's Gerald Kane.

"One of the biggest factors is senior leadership vision," says Kane, who is co-author of MIT Sloan's 2015 report on digital transformation. "When Wall Street demands short-term financial performance, it can be really hard to say 'No, we need to equip our organization for the long term.' You'll never achieve it if you can only plan 18 months at a time—you need a 10-year vision."

Digital transformation is more about creating a culture of change than about technology per se, Kane says, and many organizations simply aren't prepared. In the aforementioned Unisys/IDG survey, only 15 percent of executives believed their companies were nimble enough to adopt a digital business model.

"The far greater challenges are organizational ones," he says. "Talking about implementing a new platform is not enough. It's about how to start thinking differently, operating differently—how your culture aligns to be more amenable to change."

Organizations also need to have a clear idea what it is they want to transform, says Phil Simon, author of seven books on management and a lecturer at Arizona State University's W.P. Carey School of Business.

"People talk about transformation like it's binary," he says. "But it might be different parts of a company that need to change—like a website built 20 years ago, or sending out mass emails to communicate with employees. What are you transforming and what's the gauge? Are you trying to grow revenue, market share, number of users? Are you trying to innovate quicker? It's so broad, everyone is going to have a different interpretation."

Enterprises need to understand that digital transformation is an ongoing process. Perhaps the most vexing conundrum for management is that organizations will never be able to claim that their transformation is complete.

"There's also this notion that transformation ends—that you have transformed," he says. "I would argue that more than ever we are constantly tranforming, that there is never an end point. That's why things like agile development methods have blown up, and why you see things like devops explode in IT shops."

Another important imperative is to relax central control of technology development. Yes, you need to build in security, compliance, and data governance to prevent practitioners from making poor choices. But the new, decentralized IT accepts that those closest to the internal and external customers understand requirements best, and a centralized, top-down IT structure can't possibly move fast enough to develop all solutions.

Now more than ever, we need leadership that can envision the fluid nature of business and create frameworks that empower rather than constrain creative developers and consumers of systems inside and outside the enterprise. That's how digital transformation fulfills its promise. ■

Eric Knorr *is editor in chief and Dan Tynan is a contributing editor for InfoWorld.*