Modernize Application Development to Succeed as a Digital Business

Published: 30 March 2016

Analyst(s): Kirk Knoernschild, Eric Knipp, Richard Watson, Sean Kenefick, Danny Brian, Gary Olliffe, William R Holz, Brad Dayley, Lee Cheetham

To enable their enterprises to seize digital business opportunities, development teams must modernize architecture, infrastructure and processes to deliver higher-quality applications with greater agility. This special report brings together a range of research to help meet these challenges.

Analysis

In the digital business era, organizations need unprecedented velocity and agility to seize new opportunities — and software development teams are under pressure to deliver high-quality software more quickly than ever before. Drawn-out development projects are no longer acceptable, given business demands. However, at the same time that developers are confronting this challenge to deliver solutions more quickly, they are also facing the most diverse technology ecosystem in the history of computing.

Addressing this challenge in a complex technology environment requires a strong relationship between architecture, infrastructure and process (see Figure 1). Even high-performing teams have reached the limits of agility and efficiency that can be achieved by focusing improvements on only one of these areas. Teams must focus on all three because, now more than ever before, software architecture, infrastructure and process are intertwined. Improvements in one area demand improvements in another, and failing to improve holistically will not deliver optimal results. Moreover, to help teams deal with the diversity that awaits them, an investment in training is also essential.
For development teams to take application agility and efficiency to the next level, they must address all these areas holistically by taking the following steps:

- **Focus on modern software architecture paradigms and frameworks.** Because development teams must deliver richer experiences with greater velocity and agility than ever before, modern software architecture patterns are gaining momentum.

- **Replace heavyweight platforms with lightweight application infrastructure.** Lighter-weight application infrastructure is necessary to achieve the goals of modern architectural patterns and to deliver the scalability organizations require.

- **Invest in automation and strive for increased agility across the life cycle.** Automation is a necessity because modern architecture and lightweight infrastructure have many moving parts that are impossible to manage manually.

Teams that have not begun to embrace these technical trends must do so in 2016 — and Gartner research is available to help. In fact, the urgent need to execute on these priorities is driving many of the most common topics of inquiries from Gartner for Technical Professionals (GTP) clients.

This special report brings together a range of GTP research reports that will help IT professionals best prepare for these and other application-planning challenges they will face. This research
provides the information technical professionals need to seize the opportunity to deliver better quality and to gain the agility needed to respond to digital business imperatives.

Research Highlights

Focus on Modern Software Architecture Paradigms and Frameworks

Modern architectural patterns are a foundational requirement to achieve three key goals:

- Increasing velocity and agility
- Overcoming challenges with endpoint diversity
- Delivering compelling experiences

To gain these benefits, it is important to understand and embrace key trends occurring on the client side of modern applications, on the back end and in the cloud.

Focusing on the client: Modern Web applications feature rich interactions and behave increasingly like native applications. The trend toward building rich browser-based apps is pushing more processing to the client and enabling more scalable and distributed code. As development teams are also tasked with delivering great apps to mobile devices, portability of code across different mobile platforms is a top priority to help ease development cost and maintenance. Apps built with a "hybrid" architecture balance portability with access to native features so that native code can be written to access the native features while Web code is written for the portions of the app where portability is critical.

GTP reports that provide guidance on how to effectively embrace modern Web and hybrid mobile architecture paradigms include:

- "Decision Point for Selecting Your Mobile App Architecture"
- "Modern Web App Architecture"
- "Blueprint for Architecting Modern Web Applications With HTML5, CSS and JavaScript"
- "Blueprint for Architecting a Hybrid Mobile App"
- "Decision Point for Choosing a Web Application Architecture"

Modernizing application architecture for the back end and cloud: The demand for increased agility and the rise of the cloud are fueling a shift on the server side. Pervasive adoption of service-oriented architecture (SOA) — including Web APIs and microservices — as well as software-defined infrastructure and cloud platforms are becoming the new foundational paradigm for software architecture on the back end. Today's applications must support multiple client interfaces and be incrementally deployable. They should use resources efficiently to suit the cloud era because resource conservation is a key consideration in cloud-based, pay-as-you-go, virtualized infrastructure environments. Microservice architecture has emerged as an important means of
applying advanced SOA principles and patterns at a fine-grained level to deliver business capabilities with agility and precise scalability.

The following GTP research will help application teams prepare for these aspects of their architecture modernization efforts:

- "Assessing Integration Architecture for Internet of Things Solutions"
- "How to Architect and Design Cloud-Native Applications"
- "Assessing Microservice Architecture for Scalable Application Delivery"
- "Solution Path for Platform Architecture to Deliver Application Services and APIs"
- "Decision Point for Choosing an Application Services Integration Architecture"

**Tools and frameworks to help implement modern software architecture:** To implement modern software architectures effectively, application teams will need to become familiar with some of the advanced tools and frameworks available and to develop effective "toolboxes" for developing Web applications and APIs.

GTP reports that offer relevant guidance include:

- "Choosing a JavaScript Framework"
- "Building Modern Web Applications With AngularJS"
- "Guidance Framework for Creating Your API Developer Toolbox"
- "Guidance Framework for Creating a Modern Web Application Development Toolbox"
- "Assembling an Open-Source Mobile App Development Stack"
- "Node.js: An Indispensable Tool for Modern Web Development"

**Replace Heavyweight Platforms With Lightweight Infrastructure**

We are moving beyond the era of heavyweight application servers and integration suites. Agility and elasticity are the defining characteristics of modern application infrastructure. Typical applications use only a small portion of the capabilities available on heavyweight platforms, and the complexity of these "superplatforms" often impedes a team's ability to quickly provision new infrastructure.

Newer, popular lightweight Web servers are appearing, including many open-source solutions. Apache Tomcat and Eclipse Jetty are examples of servers frequently associated with microservice and cloud-native Java architectures. Many newer application servers are designed so that they can be packaged with the application, essentially shifting the deployment model away from installing many applications on a single server instance and toward deploying discrete and autonomous microservices.

**Moving application platforms into the cloud:** Organizations are turning to cloud platforms, especially public and private application platform as a service (aPaaS) offerings, to improve
development team productivity, shorten time to market and reduce operational overhead. aPaaS solutions offer these organizations a ready-made platform on which they can load some code and run it. These platforms allow application developers and system administrators to build, ship and run distributed application components with the guarantee of platform parity across different environments.

GTP reports that offer guidance on critical issues related to cloud application platforms include:

- "Decision Point for Cloud Application Migration"
- "Evaluation Criteria for Enterprise Application Platform as a Service"
- "In-Depth Assessment of Microsoft Azure Application PaaS"
- "Blueprint for Deploying a Microservice Architecture With PaaS"

**Containerizing application workloads:** The use of OS containers, led by the popular Docker framework, is also on the rise. Containers allow application developers and system administrators to package, ship and run distributed application components with guaranteed platform parity across different environments. Container offerings like Docker are "democratizing" virtualization by providing it to developers in a usable, application-focused form. Whereas access to virtual machine virtualization tends to be provided through, and governed by, gatekeepers in infrastructure and operations, Docker is being adopted from the ground up by developers using a DevOps approach.

GTP reports that will help teams explore containers and plan for their deployment include:

- "Assessing Docker and Containers for Five Software Delivery Use Cases"
- "Docker Democratizes Virtualization for DevOps-Minded Developers and Administrators"
- "Become More Agile and Get Ready for DevOps by Using Docker in Your Continuous Integration Environments"

**Invest in Automation and Strive for Increased Agility**

If development teams expect to optimize delivery, they can no longer rely on time-consuming and error-prone manual testing and deployment processes. The test and deployment pipeline must be automated to support emerging new architectural patterns and infrastructures, which involve more frequent deployments of lighter-weight processes to virtualized infrastructure. There are simply too many moving parts, and the benefit of frequent change is too important, to rely on managing everything manually.

Gartner recommends that application organizations emphasize automation as the first step toward continuous delivery, which is the DevOps practice of deploying small, frequent application changes to a production environment. The premise of continuous delivery is to ensure that the development team maintains a production-quality version of the application at all points throughout the development life cycle. Doing so requires good, agile practices and a fair helping of automation.

Three focus areas are particularly important for achieving these goals:
Devising effective testing strategies

Adopting effective frameworks and practices to boost agile and DevOps prowess

Using metrics to track progress

**Focusing on testing strategies and practices:** For agile, continuous development practices to succeed in rapidly delivering applications into today’s diverse technology environments, effective testing is a must. Development teams aiming to get better-quality products to market faster must look to automate testing and move it earlier in the development process. Given the key role of APIs and mobile apps in modern software architectures, appropriate testing approaches to suit these new paradigms are key.

GTP reports that provide guidance in these areas include:

- "A Guidance Framework for Testing Mobile Applications"
- "A Guidance Framework for Testing Web APIs"

**Increasing development efficiency and agility:** Teams seeking to boost the effectiveness and efficiency of their application development and integration efforts are increasingly turning to agile and DevOps methodologies, which focus on using continuous improvement to evaluate and update practices, processes, tools, environments and source code. A number of frameworks and disciplines are key to these efforts, such as Extreme Programming (XP), behavior-driven development, continuous integration, and the Scaled Agile Framework (SAFe).

GTP has published a number of reports that offer guidance on these methodologies, practices and frameworks, including:

- "A Guidance Framework for Continuous Integration: The Continuous Delivery 'Heartbeat'"
- "Implementing Enterprise Agile Using the Scaled Agile Framework (SAFe)"
- "Extend IT’s Reach With Citizen Developers"
- "Increase Collaboration and Drive Agility With Behavior-Driven Development"

**Collecting data to measure progress:** Finally, you can’t judge your progress in adopting modern automation and agility practices unless metrics are in place to effectively measure these efforts. Gartner recommends that application teams use a structured approach to data collection in order to track the benefits of their efforts to increase agility and adopt DevOps techniques such as continuous delivery. Teams must be able to discern when changes they make to their processes have been positive and helpful, and continually evaluate themselves and their abilities — not only to ensure that they are ready for change, but also to set a baseline against which they can measure their progress in the future.

The following report offers guidance on how teams can accomplish this goal over the course of their journey to modernize their application practices:
"Use Metrics to Drive Your Agile, DevOps and Continuous Delivery Initiatives"