Gartner

Research

Publication Date: 3 August 2011 **ID Number: G00214710**

Apps Intelligence: A Dynamic Capability for Application Development

Brian Prentice, Eric Knipp

Apps intelligence is a dynamic capability involving a continuous cycle of application improvement that incorporates instrumentation, forensic examination, agile development and operational practices, and user-centric design. It is a dynamic capability that will lead to ever-improving applications.

Key Findings

- By using application performance management (APM) and customized instrumentation within a Web application, developers and business managers gain visibility into user behaviors and preferences. This visibility is not based purely on intuition, but on measurable data about how users actually interact with a system.
- Apps intelligence is particularly important for application development (AD) initiatives involving systems of innovation and systems of differentiation that are intended to deliver competitive advantage to the enterprise.
- The most compelling examples of apps intelligence can be found among cloud services providers, like salesforce.com, and top consumer Web properties, like Facebook, as they take advantage of Web-oriented architectures.
- Apps intelligence will lead to industry consolidation. Cloud services providers that execute this capability at scale first will rapidly create barriers to imitation that will be difficult for new entrants to overcome.
- Enterprise AD organizations can realize the benefits of apps intelligence in similar ways to large-scale cloud services providers; larger user bases will deliver more statistically significant data.

Recommendations

- Enterprise IT organizations should apply apps intelligence to help in determining ways to improve user experience and in quantifying the payback of user satisfaction. This will ease the process of justifying the investments needed to take advantage of application intelligence within their own portfolios of custom-built applications.
- Apps intelligence should be coupled with the measurement of developers on quality metrics tied to user satisfaction.

- Expect apps intelligence to become a unique selling proposition of cloud services providers as they increasingly take advantage of their unique visibility into their customers' behavior while aggregating the data at a scale impossible elsewhere.

TABLE OF CONTENTS

LIST OF FIGURES

Solution Design Is Hard

Traditional, custom-made Web solutions are constructed over time through a continuous, iterative approach that includes soliciting requests for functionality, prioritizing those requests and then developing, testing and deploying as many of those requests as time will permit. Requests for functionality are generally driven by direct solicitation and, once collected, are prioritized. The methods used to prioritize are generally little more than intuition or a careful assessment of the political power of each feature's requestor relative to others. Although truer for custom enterprise solutions than for packaged solution providers, quantitative data is rarely the "forcing function" behind new or changed applications and features.

Packaged Software Vendors and Lead Customers

Packaged software vendors creating a new release have few sources of data they can use. Customers are often surveyed, but surveys tend to be of limited use thanks to a response rate that is often less than statistically significant. Trouble tickets can be consulted, but, again, these are unreliable, revealing only problem areas, rather than also areas of opportunity. Furthermore, it is challenging, if not impossible, to judge how many customers had problems, but did not report them. Most large software vendors use lead customers to assist in the release development process. Besides these efforts being time-consuming and expensive to manage, there is usually a selection bias associated with this process. These lead customers tend to be selected from the largest companies that have made the most significant investment in the vendor's products. But their requirements and their ability to optimize an ever-expanding solution are generally not indicative of the majority of the vendor's customers.

Enterprise Web AD groups and Basic Web analytics

Enterprise Web AD groups rarely use data beyond basic Web analytics (if that) to prioritize and select features for inclusion in a new revision of existing software. There are further differences tied to the dominant AD project methodology in place. In waterfall-style projects, requirements are typically gathered in a "big bang," whereby subject matter experts — usually business analysts or other internal "customers" — are interrogated and their specifications put to paper. In a gated process, project participants sign "in blood" that the requirements are known, final and that implementation may proceed. Rarely, if ever, are data used in planning. When data points are used, they consist primarily of Web analytics data, in particular for consumer-facing e-commerce platforms. While Web analytics are valuable in their own right, they are only a jumping-off point — to best understand user behavior and where incremental application investment should occur, deeper analysis is required.

Usability Study Data and Ad Hoc Reporting

Occasionally, usability study data is incorporated into the design process, and in some organizations other ad hoc reporting may be used to identify problems but is rarely used to tie new features to specific data points that justify their existence. Usability studies are generally conducted once the feature sets have been determined and have reached the testing stage. Usability studies seldom precede the design phase and as a result do not have any impact on the overall conceptual design of the application. Incorporating usability practices that do not depend on a finished product (such as paper prototyping) is a good enhancement to AD projects, but by themselves these ideas do not "close the loop" with real-world practices when the software is deployed.

Agile and Iterative Projects

In agile and iterative project styles, the continuous feedback loop between stakeholder and developer is a dramatic improvement over the waterfall style, but it is still solicitation-driven, rather than data-driven. Indeed, the very argument often used to sell agile development to the business is the reality that users don't know what they want until they see it. Through their craftsman like approach, agile and iterative methodologies overcome many challenges of big-upfront requirements processes, but fail to approach the objective of good application design scientifically.

Creeping Irrelevance Plagues All Software Apps

Whether it's a vendor or an enterprise AD team designing an application, and whether they're building it using waterfall, iterative or agile development methodologies, creeping irrelevance is a problem that plagues all software applications. The problem with the process of solicitation and prioritization is that there is no associated effort to establish meaningful distinctions between people and the objectives that drive their feature requests. Applications, over time, expand to meet the needs of as many people as possible, and application designers generally see that as a good thing. But as an application expands to meet the requirements of as many people as possible, the application as a whole becomes less and less relevant to all its intended audience. Furthermore, functional density makes an application increasingly complex, which directly impacts ease of use. In the case of software, the whole can often become less than the sum of the parts.

A study published in "Harvard Business Review" (see R. T. Rust, D. V. Thompson and R. W. Hamilton, "Defeating Feature Fatigue," 1 February 2006) discovered that, "before use, capability mattered more to participants than usability, but after use, usability drove satisfaction rates. As a result, satisfaction was higher with the simpler version of the product, and in a complete reversal from the earlier studies, the high-feature model was now rejected by most participants."

Agile and Iterative Practices, and a User-Centric Design Process

These complexity challenges can be mitigated somewhat through agile and iterative practices that depend on a user-centric design process (see "Top 10 Mistakes in Web and User Experience Design Projects"), but not without disciplined and continuous refactoring and repayment of technical debt. Even in agile projects, only a subset of users is included in the design and development of a software product. Features that some users want — or don't want — may still be missed. Despite its strengths, agile still relies heavily on expertise and intuition. Even when usability data is used, it is often subjective and difficult to incorporate scientifically. What is needed is a way to sustain relevance. Tools and techniques are needed by AD teams to be able to better correlate users and features. This can help direct new functionality in a way that will have the maximum value to a majority of a solution's intended audience. And critically, it can help in determined how applications can be streamlined to increase relevance where it's been faltering.

Apps Intelligence: Blending Art and Science

Apps intelligence is a dynamic capability that naturally flows from APM and instrumentation, forensic application examination, data- and human-driven analysis and prediction, conceptual and user-centric design, and agile development and operational practices. A dynamic capability "is located in those activities and business processes that govern the design and execution of key innovations that improve customer value or lower the firm's cost. Without such a capability, enacted over and over, a firm cannot neutralize and overcome the onslaught of innovations by startups" (see G. Walker, "Modern Competitive Strategy," McGraw-Hill, 2009). While particularly

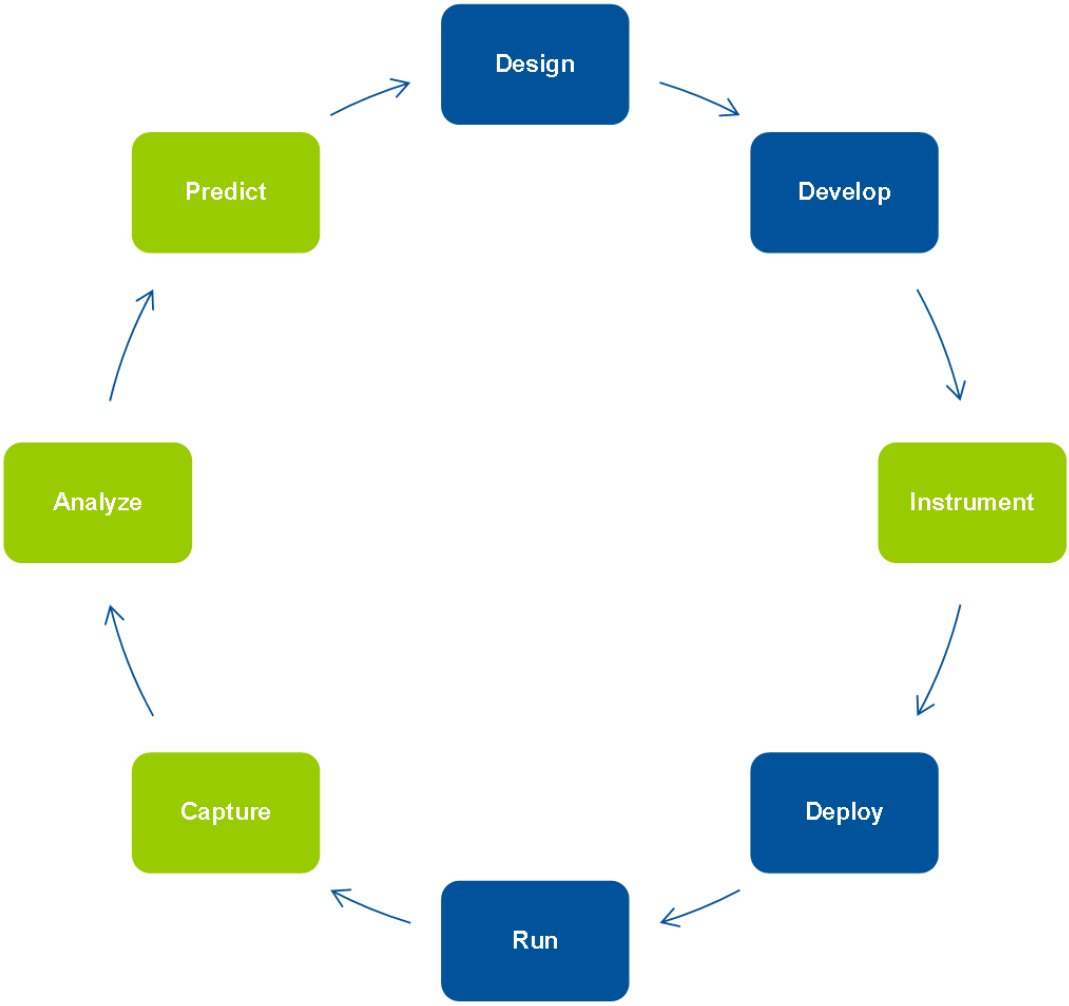
applicable to solution and service providers, apps intelligence can be applied to any organization that uses custom-developed applications to generate competitive advantage.

By using APM and customized instrumentation within a Web application, developers and business managers gain visibility into user behaviors and preferences. This visibility is not based purely on intuition or "horse sense," but on measurable data about how users actually interact with a system. In a way, a well-instrumented solution is a continuous usability experiment. Highly relevant features are heavily used, while features that are superfluous to the requirements of a majority of an application's user base are barely used. Solution providers have direct, line-of-sight visibility into where users get the most and least satisfaction, and can act accordingly in a rapid, iterative manner, delivering incremental improvement over time, again and again. Additionally, this visibility can help in better understanding the different types of user constituencies that exist in an organization. Course groupings of users can evolve to more nuanced, finely tuned communities, which further assists in the application design process.

Web Apps and Visibility Into Customer Use of a Running Solution

We focus in this research on Web applications, because the apps intelligence capability depends upon perfecting the visibility into customer use of a running solution. However, any solution delivered over the Internet or other continuously connected network can benefit from apps intelligence (see Figure 1).

Figure 1. The Apps Intelligence Dynamic Growth Cycle



Source: Gartner (August 2011)

- **Design:** This stage involves the assessment of the application's underlying purpose, the problem it solves and the people who will benefit from it. This is not a "technical" design process. It is a conceptual design process, where the observation of how people interact with their environment can be improved with a software application. The data that flows from the apps intelligence dynamic growth cycle is central in assuring that the design assumptions are valid.
- **Develop:** This stage encompasses design, development, testing and quality practices. Initial construction is largely intuition-driven, as there may not be existing instrumented data to explain the reason for a new solution's existence. For solutions in place, growing volumes of insight derived from actual user data should direct design and development choices. Agile practices are used to improve productivity and visibility into the development stage. Automation, short iterations and, most importantly, close collaboration between the AD team and the business team are required for success (see "Enterprise-Class Agile Development Defined").

- **Instrument:** This stage includes the act of linking measurement capabilities to the software features and functions users interact with. Example instrumentation activities include configuring a packaged APM suite to monitor particular events, or incorporating customized auditing and activity monitoring features into a solution. In practice, some percentage of instrumentation activities must be completed in concert with development. More advanced apps intelligence practitioners will always link instrumentation and development tightly together, as the insights gained during analysis and prediction will often necessitate additional instrumentation to capture finer-grained user preferences and behaviors.

- **Deploy:** This stage includes the act of moving the new or improved solution capabilities into a running environment so that customers and users can interact with them. For AD organizations delivering a single application to a single set of users, deployment may seem a relatively trivial task, compared with the rest of the apps intelligence cycle. However, for large cloud services providers, deployment is a major challenge, as customers may demand different SLAs governing the frequency and timing of upgrades, and as a large, distributed infrastructure environment may necessitate a closely managed and staged upgrade process. A growing percentage of cloud services providers and, indeed, traditional IT organizations are employing heavier automation to manage increasingly complex deployment challenges. A set of practices known as DevOps is emerging to address this new trend (see "Deconstructing DevOps").

- **Run:** This stage includes delivering the new and improved capabilities to solution consumers. For enterprise AD organizations, this is the point when users are cut over to the new system from the old, and when monitoring of the new and improved capabilities begins. For cloud services providers, this is the point when a customer accepts the upgrade and begins using it. It is also the point when new customers can be expected to evaluate and possibly adopt the offering on the basis of improved usefulness.

- **Capture:** Data capture begins as soon as the solution starts running. All of the instrumentation and monitoring functionality kicks in, and measurable data around user behaviors and preferences accrue. Solution delivery organizations may choose to construct dashboards or other types of real-time data analytics at this point, or they may periodically extract and load the data into an analytics platform for further mining.

- **Analyze:** Analysis of data can be automated (machine intelligence) or ad hoc (human intelligence) or some combination of both. The best approach involves automating the repeatable, while using experienced human analysts to generate new insights and identify new automation opportunities. In an end-user AD organization, business analysts should expect to work closely with solution architects and developers in order to better understand the data exposed by the system. Cloud services providers will deploy a combination of technical product managers and business-oriented product marketers. Large scale applications will require "big data" analytics expertise in addition to business expertise in the given solution domain (see "'Big Data' Is only the Beginning of Extreme Information Management").

- **Predict:** This stage requires that project or product owners make predictive decisions about the functionality or design they expect will improve the value proposition the solution delivers to the user. Closely linked to the analysis stage, the predictive stage assembles the insights gained through the combination of data and intuition into "marching orders" for the solution delivery team.

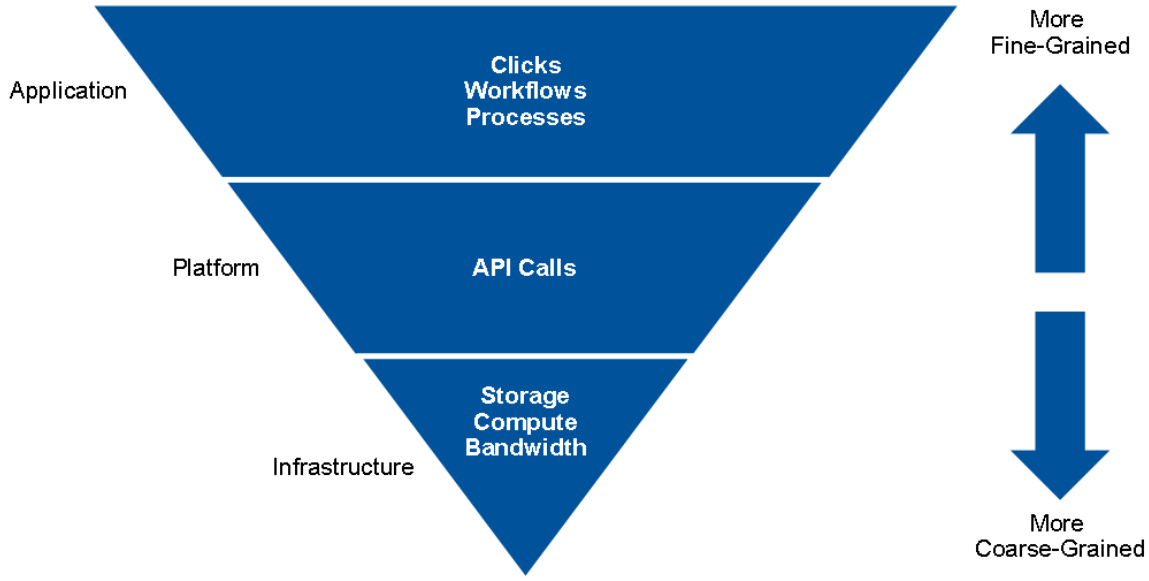
Most organizations today do the design, development, deploy and run steps. But it's by combining the instrument, capture, analyze and predict steps that the apps intelligence dynamic capability is created.

With time, iteration and sufficient instrumentation and analytical capabilities, a solution provider is in a position to predict and satisfy the needs of its users and customers. This predictive capability is enabled by analysis that takes into consideration data and human intuition and experience. A recent study from the MIT Center for Collective Intelligence reveals that while "statistical models almost always yield predictions that are more accurate than those of human experts," "humans are better at data acquisition and at recognizing atypical circumstances." (see Y. Nagar and T. Malone, "Combining Human and Machine Intelligence for Making Predictions," MIT Center for Collective Intelligence Working Paper, March 2011). The analytical and predictive stages of the apps intelligence cycle are, therefore, predicated on practitioners combining machine and human intelligence to achieve better outcomes. For this reason, we believe that apps intelligence is a learned capability, and that solution providers that practice it more frequently and with greater discipline than their peers will achieve better results from it. More users with more data, therefore, facilitate more practice and, thus, better execution of the apps intelligence capability.

Apps Intelligence in the Field: Clicks, Calls and Compute

While we view apps intelligence primarily as a weapon in the battle to continuously improve software, in practice, the concept is not limited to applications. The benefits of apps intelligence should be obvious to any service provider delivering IT solutions at scale, as shown in Figure 2.

Figure 2. Application Intelligence and Provider Visibility



Source: Gartner (August 2011)

The popular n-tier Web application architecture supports capture of user behavior at several levels. At the surface level, users interact with Web applications via clicks and keystrokes. Instrumentation can be configured to capture the details of these clicks and translate them directly into the workflow tendencies of users. This data can help application architects and business managers answer important questions that drive where R&D dollars go:

- Which features are most important (either to the user or to the provider)?

- Which features are too cumbersome (or cost more to deliver than the value they generate)?
- Which features are missing altogether (and would provide new value opportunities)?

While user clicks will be the most easily understood measurement for business managers, developers will be interested in looking just below the surface, at API calls included in the solution environment. This maps the activities of programmers, and can answer questions like:

- Which API calls should be optimized (or are called most frequently or perform most poorly)?
- Which API calls should be expanded (to deliver additional business capabilities or simplify and streamline the solution architecture)?
- Which API calls are problematic (and frequently cause system errors or user disruption)?

Software and physical infrastructure form the bedrock of any IT solution, providing the storage, network and compute platform upon which further value is derived. The better solution architects and system administrators understand the resource requirements of a solution, the more efficiently those needs can be addressed. In particular, instrumentation and monitoring at this level can help provide answers to important questions like:

- When are resources used (and how can we better predict peaks and valleys)?
- Where are resources used (and how can we provide them more inexpensively)?
- When do solutions fail (and how can those failures be avoided or mitigated)?

Useful Tool Categories Help Capture Data

There are several useful categories of tools that can help solution architects and business managers instrument and capture data. Web analytics platforms like Omniture are an obvious example, providing a wealth of information about user interactions on a website, such as click paths and abandon rates. In fact, without Web analytics as a starting point, most apps intelligence initiatives will not be successful. In addition to Web analytics, Web services analytics offerings like Mashery's API Value Tracking can help establish links between API utilization and customer value. Workflow analytics available in modern business process management (BPM) suites like IBM Lombardi can provide information about frequently and less frequently used business processes. General and custom log files can be instrumented with services like Splunk and Loggly. Beyond packaged offerings, architects and business managers can devise customized data capture mechanisms suited to their particular solutions.

Adopting Apps Intelligence: Market Dynamics and the Impact of Cloud Computing

While apps intelligence defines a powerful dynamic capability for both enterprise IT organizations and software providers, it is highly likely that it will be initially driven by the cloud services provider community. Assuring the relevancy of an application ultimately leads to user satisfaction. Although user satisfaction is an objective both enterprise IT organizations and software providers aspire to deliver, the fact is that it is very difficult to quantify the direct payback from any investment made to achieve that objective. The software vendor community largely overlooks this challenge, because its products are being delivered in the general marketplace, where user

satisfaction is one of several key considerations driving demand for their products. Additionally, some vendors seek to make user satisfaction the essence of their unique selling proposition.

Enterprise IT Organizations

For the enterprise IT organization, the challenge of not being able to quantify the payback of user satisfaction creates tangible restrictions. The question comes down to whether the investments required to increase user satisfaction will deliver greater business value, compared with other investments opportunities. Custom-built software, after all, supports the objectives of the broader business — it is generally not a business in its own right.

Enterprise IT organizations have also had the luxury of dealing with a captive audience. But a "take it, or leave it" perspective is getting harder and harder to get away with. Users are not only demanding significant improvements in the experience and relevancy of their applications as a result of the advances being made by software vendors, but they're also increasingly turning to cloud services to bypass the IT department in order to obtain the solutions they want to use. A recent survey conducted by professional services firm Avanade reveals that one in five employees has procured cloud services without the knowledge or consent of IT (see "Cloud Computing"). Enterprise IT organizations will eventually be forced by the vendor community to either focus on apps intelligence, or risk squandering development resources through lack of user engagement with the end solution.

However, the adoption profile of apps intelligence by software vendors will be marked by a clear distinction between on-premises packaged software application providers and cloud services providers. This boils down to one simple fact — packaged application vendors have no visibility into software deployed in an on-premises data center or onto a customer desktop. On the other hand, cloud services providers have visibility into customer activity as, by definition, users interact directly with shared services they provide.

Web-Oriented Architecture

Many cloud services providers fully embrace Web-oriented architecture (WOA), an advantage as WOA naturally provides excellent integration points for APM and custom instrumentation in the space between service calls. And there are well-known examples of apps intelligence in action with major cloud services providers today. Social media platforms Twitter, LinkedIn and Facebook are some of the most data-driven software operations in the world, relying on massive numbers of user interactions to yield useful data points that lead to innovation-generating insights and analyses. Facebook, for example, uses apps intelligence to show code to specific demographics and geographies for testing purposes. It can determine the percentage of people seeing a new feature, and can scale up and down the new features on the fly. This dynamic also holds true as you move down into the platform and infrastructure layers. Microsoft's Azure platform is designed to capture data at several levels across its thousands of users — first, coarse-grained data such as storage, compute and bandwidth consumed are used for capacity planning. Finer-grained instrumentation is also built into the platform so that specific .NET API calls can be visualized, enabling planning at the platform middleware layer to be data-driven.

Software-as-a-Service and Cloud Services Providers

In some instances, cloud services providers have stumbled upon apps intelligence. Software-as-a-service (SaaS) providers, in particular, are more susceptible to high rates of customer churn (the ratio of new customer acquired versus those that leave for other providers). In order to combat this outcome, SaaS providers seek to drive loyalty and retention by proactively increasing the breadth of features of their solution that each customer utilized.

The clear thought leader in this area has been salesforce.com (see "How Salesforce.com Manages Functional Complexity"). It began by providing its customers with a dashboard that detailed each functional area of the application, showing whether end users were utilizing it and, if so, to what extent. Over time, it began aggregating this data with usage data from other clients. This now allows salesforce.com to show how its customers are using each functional area of the application relative to an industry peer group. Each underutilized capability in the report was hyperlinked to information that described a set of best practices for how it could deliver business value.

Salesforce.com demonstrates an opportunity with apps intelligence that is unique to cloud services providers. Usage data can be aggregated across multiple different customer organizations, and customers can get comparative data on the extent to which they're taking advantage of the solution relative to other organizations in a selected peer group. This data and the feedback loop it generates prove critical to cloud services providers, allowing them to fine-tune and rightsize their solutions. The result is a higher overall value proposition, as customers are less likely to end up with a high-cost solution where the bulk of deployed functionality is unused. This stands in stark contrast to on-premises packaged application providers where their solutions are being marked by a growing gap between the functionality being delivered through successive revisions and what their customers are even aware of, much less taking advantage of. There is no excuse for cloud "shelfware."

Larger-scale cloud services providers also have a natural advantage: More users equate to a bigger opportunity for data capture and instrumentation. More captured data translates into more input for the apps intelligence cycle. While scale does not guarantee success, it is difficult to imagine a scenario where a small-scale provider achieves big-scale apps intelligence outcomes. Companies that learn how to do apps intelligence faster will develop critical mass quicker, and will create near-insurmountable barriers to imitation within the scope of their core business.

Bottom Line

Apps intelligence is a product of economies of scale and economies of scope. The ability to serve a single customer cheaper than competitors enables a firm to offer lower pricing, higher margin or a combination of the two. The ability to serve a single customer more completely than others allows a provider to capture a larger percentage of IT expenditure for a particular set of capabilities. Apps intelligence is also applicable to enterprise IT organizations, in particular, those that directly serve large numbers of users with Web applications. For both types of organizations, apps intelligence should become a required component of application design. Limiting factors should not be overlooked:

- **Limiting Factor 1 — Instrumentation:** The information must be captured and made visible before it can be understood. Far too many applications have limited or no instrumentation at all.
- **Limiting Factor 2 — Analysis and Prediction:** Function points must be granularized at the right level in order for information to be digestible. The information must be put to actual use and that requires a rethink in how application maintenance and requirements definition and prioritization activities are managed.
- **Limiting Factor 3 — Operational Tempo:** New and changed function points must be implemented in a rapid and iterative manner so that value predicted can be value generated and enjoyed by users.
- **Limiting Factor 4 — Organizational Culture:** Apps intelligence is a new capability that borrows existing skills and roles and implies new ones. Concepts of project ownership, the software development life cycle, and how value is measured will change as the

organization explores and implements an apps intelligence initiative and grapples with the interdependencies it requires.

Apps intelligence is ideal when the organization desires continuous value improvement in a particular solution set, but it is not without blind spots. Brand new solutions or features totally outside the scope of an existing system will be missed. Data in unknown areas does not exist for analysis. While analytics can help find the gaps, human experience and intuition are required to fill in those gaps. Systems with insufficient users will not generate statistically significant data useful for analysis and prediction. Small-scale applications can benefit from usability studies, but the full gamut of apps intelligence practices may not generate positive ROI.

RECOMMENDED READING

Some documents may not be available as part of your current Gartner subscription.

"Top 10 Mistakes in Web and User Experience Design Projects"

"Enterprise-Class Agile Defined"

"Deconstructing DevOps"

"'Big Data' is only the Beginning of Extreme Information Management"

R. T. Rust, D. V. Thompson and R. W. Hamilton, "Defeating Feature Fatigue," 1 February 2006

G. Walker, "Modern Competitive Strategy," McGraw-Hill, 2009

Y. Nagar and T. Malone, "Combining Human and Machine Intelligence for Making Predictions," MIT Center for Collective Intelligence Working Paper, March 2011

Gartner

REGIONAL HEADQUARTERS

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters
Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters
Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters
Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters
Gartner do Brazil
Av. das Nações Unidas, 12551
9° andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509

Gartner