Systems of Differentiation: How to Build Capabilities That Provide Competitive Advantage

Published: 8 June 2017    ID: G00328674

Analyst(s): Bill Swanton, Mark O’Neill

Application leaders charged with delivering fast-changing, differentiating business capabilities must develop appropriate governance and technology platforms for systems of differentiation as part of an overarching Pace-Layered Application Strategy. This document describes how.

Key Challenges

- Governance processes hold back many organizations by not being agile enough to meet business needs for differentiating capability.
- IT organizations will need to engage with the business more intensely to generate maximum value from systems of differentiation.
- To realize a key benefit of Pace-Layered Application Strategy, unique, customized software must be cleanly decoupled from more commoditized, packaged applications.

Recommendations

Application leaders developing governance and technology platforms for systems of differentiation in pursuit of a Pace-Layered Application Strategy should:

- Decide with business leaders which business capabilities are differentiating, and then segment the application portfolio into systems of differentiation and systems of record.
- Create streamlined governance processes for systems of differentiation.
- Assess their options for building systems of differentiation. These include developing new applications and using packaged applications, SaaS, mediated APIs, and mesh app and service architecture (MASA).
- Take a product-centric approach to enable faster responses to new business requirements.
Introduction

Systems of differentiation are the heart of a Pace-Layered Application Strategy. As shown in Figure 1, they are the applications, or parts of applications, that deliver unique and differentiating capabilities that distinguish an enterprise from its competitors. They are not readily available on the packaged software market. (For more details, see "What Is Gartner’s Pace-Layered Application Strategy and Why Should You Use It?").
Figure 1. A Pace-Layered View of Systems

Analysis

Decide With Business Leaders Which Business Capabilities Are Differentiating

Systems of differentiation have capabilities that deliver the most important business value that is desired and appreciated by the business. In some cases, they are parts of applications. Since an application often supports multiple business capabilities, it can contain capabilities that fit into both system-of-record and system-of-differentiation layers. The differentiating capabilities are often critical enablers of the business strategies that the organization is counting on to win in the market in which it competes.

Systems of differentiation are critical tools for delivering unique products and services, for becoming faster or more effective than competitors, and for closing deals faster or delighting customers more. Used correctly, they enable IT to contribute to increasing an enterprise's revenue, market share and profit margin.

Examples of system-of-differentiation capabilities include:

- A unique product configuration capability that gives customers more choice without adding to sales costs.
A streamlined capability that responds to an RFP faster than competitors, while still involving sales, engineering and manufacturing departments to ensure the bid will be profitable.

An automated insurance underwriting capability that reduces time and labor, while managing risk better and responding to customers faster.

As an application leader, you should engage with your business peers to reach clear agreement about which business capabilities are (or should be) differentiating in light of the business's strategy. You can then segment applications into separate portfolios for systems of differentiation and systems of record.

Create Streamlined Governance Processes for Systems of Differentiation

Slow, deliberate governance processes for changing systems of record may delay the availability of differentiating capabilities until the limited window of opportunity for differentiation has closed. Most organizations will therefore need to alter their governance processes to become more responsive and agile.

Where systems of differentiation are concerned, there’s a "need for speed" that doesn't exist for systems of record. This doesn't mean that systems of record aren’t critical or shouldn’t receive funding. However, governance of systems of differentiation requires more agility and perhaps even different prioritization models (see “System of Record: You Can't Innovate on an Unstable Foundation”).

A Pace-Layered Application Strategy helps an IT organization fully engage with the business and respond quickly to opportunities and threats in the market. It does so by making governance and change management processes more responsive for the system-of-differentiation layer. By restricting access to systems of record to well-defined services, systems of differentiation can be changed faster, without the risk of destabilizing systems of record or requiring time-consuming regression testing.

As an application leader, you should implement support for systems of differentiation. Specific actions to take include:

- Creating a streamlined governance process for approving the development of, and changes to, systems of differentiation. Many companies are adopting bimodal IT practices and taking an agile, product-centric approach to Mode 2. Keep in mind that there still needs to be control and that systems of differentiation can be just as critical to the business as systems of record. The governance process should recognize that these systems are ideally loosely coupled to others, so the risk is primarily within the differentiating functions (see "How to Differentiate Governance and Change Management in Your Pace-Layered Application Strategy").

- Organizing support for the streamlined governance process and engaging more intensely with the business. IT people working on systems of differentiation must fully understand the value stream for the business, and work through it in great detail with their business counterparts to show how the differentiating process will generate value. This will be easier if the IT people are aligned with particular business capabilities or products. Because knowledge of the process will grow over time, it's important to have the same people involved for an extended period. Many
companies accomplish this by moving from projects to products, with aligned teams developing and managing applications like vendor software products, even if the "customers" are internal. (see "Moving From Project to Products Requires a Product Manager").

- Using this heightened business engagement to help understand exactly how each proposed system of differentiation will change business processes or capabilities in order to deliver competitive advantage. Agree on the definition of success, and especially the key performance indicators that will be used to measure it, and what improvements from the baseline measure are necessary for success. (For more information, see "Business Outcomes Are the Milestones on an Application Strategy Roadmap.")

- Applying agile development and DevOps methods. Given that the pace of change is faster, traditional waterfall development methods with six-to-24-month delivery cycles are untenable. Agile development and DevOps methods presuppose more interactive engagement between the IT organization and the business.

Assess Your Options for Building Systems of Differentiation

Although systems of differentiation can be sourced in the same ways as any other application, some approaches are more suited than others to the required pace of change. As Figure 2 shows, there are no hard-and-fast approaches to each layer.
Your decision on sourcing methods for systems of differentiation should be based partly on your overall enterprise architecture work. The key requirement for systems of differentiation is a demonstrated ability to make changes rapidly as requirements evolve. Your options, with their pros and cons, include:

- **"Vanilla" packaged applications or SaaS.** As packages (on-premises or cloud-based) are built for the mass market, the bar for them to be classed as differentiating is set fairly high — anyone can buy the same thing. To be classed as systems of differentiation, these packages need to be configured or applied uniquely. This might be done by using functionality built for one industry in another industry, or by configuring unique analytical or decision-making capabilities by, for example, using an optimization model or rule engine.

- **Customized packaged applications or SaaS.** Traditionally, companies with unique requirements have taken packages and modified them to suit their needs. As described in
"Customization: The Cost That Keeps On Costing," this approach usually creates a maintenance nightmare, making it costly or impossible to upgrade the package. Segregating unique capabilities using one of the other techniques in this list can help you avoid the high cost of customization.

- **Developing stand-alone applications.** Traditional application development tools and methodologies can be used to build customized, stand-alone applications. Where there are no deep dependencies on other systems or data, or when such applications can be developed using well-defined, nonintrusive interfaces, this may be the best approach.

- **Developing composite applications using mediated APIs and MASA.** APIs are a key technology to enable systems of differentiation. Differentiating capabilities can be developed by first identifying capabilities that may be deficient or missing in systems of record, and then deploying these capabilities separately as software services accessible through APIs. This approach is preferable to adding the differentiating features directly within systems of record, because the latter would increase technical debt as more features are added and complexity increases (see "A Primer on Technical Debt"). At this point, a "mediated API" pattern may be used to establish a consistent model for managing APIs across systems of differentiation and systems of record, as well to increase agility and reusability (see "Design an API Mediation Layer to Underpin Your Digital Business Technology Platform"). APIs exposed in this way can be used through MASA, which enables digital business applications to support multiple client channels (including desktop, mobile and social) and multiple services in the back end (see further "Top 10 Strategic Technology Trends for 2017: Mesh App and Service Architecture").

**Take a Product-Centric Approach to Respond Faster**

More advanced systems of differentiation, especially those that also engage with third-party APIs, may use more flexible and agile development technologies. "Citizen developers" may use APIs to create composite applications, using integration SaaS (iSaaS) tools. To engage developers, API management platforms may be used to create API developer portals. Integration platform as a service (iPaaS) tools may also be used as part of a hybrid integration platform that embraces on-premises systems and cloud systems (see "Innovation Insight for Hybrid Integration Platforms"). In these ways, more open and agile development may be performed, using the capabilities of systems of differentiation, exposed as APIs.

Take a product-centric approach to enable faster response times to new business requirements. "Update Your Strategy for Connecting Experiences and Applications for a Digital and Bimodal World" describes how application governance and change management can be adjusted to reflect pace layering, in order to meet the challenge of having to evolve at the rapid pace of digital business.

Moving from a project-centric approach to a product-centric approach is key. A product-centric approach means that business capabilities are not delivered via a series of independent, one-off projects, but rather through a customer-centric focus on current and future business capabilities. Delivering differentiating business capabilities as a product means managing a product roadmap, to
achieve continuity and to satisfy changing requirements. This requires the establishment of a product manager role (see "Moving From Project to Products Requires a Product Manager").

Since APIs are used to deliver differentiating business capabilities, as part of a mediated API architecture, APIs must also be managed as products. Managing APIs as products means treating API consumers as customers, even when these consumers are inside the organization. It also means managing ongoing versions of APIs, according to a roadmap. A product manager maintains this API roadmap and prioritizes work based on business goals (see "Create the Role of API Product Manager as Part of Treating APIs as Products").

Figure 3 shows how business capabilities exposed as APIs can be assembled to create a platform for applications to access data and services in other layers.

Figure 3. A Pace-Layered Application Strategy Approach to a Differentiation Platform

This platform provides security and monitoring services, as well as access to data from systems of record. Many legacy and packaged applications have complex data models and operations that don’t match the specific needs of digital business applications, and therefore require data-mapping capability in the platform. Business services, supplied by packaged applications or composed using the mediated API platform, provide controlled access to systems-of-record functionality, allowing the upper layers to be changed frequently without requiring changes to the systems of record (see "Use Mediated APIs to Connect Your Legacy and Packaged Systems With Modern Applications").

A variety of development and integration tools are employed to build differentiating applications that use the platform. In many cases, these differentiating applications may be mobile apps, in which
case a rapid mobile app development tool may be used (see "Market Guide for Rapid Mobile App Development Tools"). When APIs exposed by the platform for systems of differentiation are combined with other APIs (for example, cloud APIs like Twilio for SMS messaging, and Xignite for financial information) to form composite applications and services, an iPaaS or iSaaS solution may be used. In each case, the goal is to exploit the differentiating business capabilities rapidly, without having to deal directly with underlying systems.

Your Pace-Layered Application Strategy platform won’t be developed overnight, but earlier service-oriented architecture, integration and business process management efforts at your company may provide a starting point. Companies will "bootstrap" the platform’s creation as part of early system-of-differentiation projects, adding pieces and services as they are needed for particular projects (see "Application Deployment Options Through the Pace Layer Lens").

One of the most important goals of a Pace-Layered Application Strategy is to make the IT organization more responsive to business requests for competitive capabilities. Building this platform should reduce the time it takes your IT organization to develop or modify systems of differentiation. In the process, grumbling by business leaders about the IT organization’s lack of responsiveness may be transformed into enthusiastic engagement in a business-IT partnership that helps the company win business.

Gartner Recommended Reading

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

"What Is Gartner's Pace-Layered Application Strategy and Why Should You Use It?"

"How to Develop a Pace-Layered Application Strategy"

"Market Guide for Rapid Mobile App Development Tools"

"Use Mediated APIs to Connect Your Legacy and Packaged Systems With Modern Applications"

"Magic Quadrant for Enterprise Integration Platform as a Service"

"Where to Start (or Restart) With Service-Oriented Architecture"

"Design an API Mediation Layer to Underpin Your Digital Business Technology Platform"
Gartner Headquarters

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
USA
+1 203 964 0096

Regional Headquarters
AUSTRALIA
BRAZIL
JAPAN
UNITED KINGDOM

For a complete list of worldwide locations, visit http://www.gartner.com/technology/about.jsp