

For: Application
Development
& Delivery
Professionals

A Developer's Guide To Forrester's Strategies For API Success

by Randy Heffner, June 9, 2015



A Developer's Guide To Forrester's Strategies For API Success

A Guided Tour Of Forrester's Research On The What, Why, And How Of APIs

by [Randy Heffner](#)
with [Christopher Mines](#) and Derek Nahabedian

WHY READ THIS REPORT

In these times of customer empowerment, digital business, and mobile mind shifts, APIs and service-oriented architecture (SOA) form a critical foundation for modern business and applications. The biggest mistake we see application development and delivery (AD&D) pros make is to think of APIs only as a technical strategy for application integration and mobile apps. Instead, think of them as integral to your strategies for rapid digital business change, broad ecosystem connectivity, and world-class customer engagement. Forrester's research in this area helps clients establish and evolve a well-grounded software strategy. This report ties together Forrester's body of research on APIs and SOA, making a cohesive whole out of 34 reports, clarifying how each adds value to the whole, and filling in gaps between reports.

Table Of Contents

- 2 **APIs And SOA Are Critical For Digital Business Success**
- 4 **How To Set Strategy For APIs And SOA**
- 5 **Architecture Sets The Proper Context For Design Of APIs And SOA**
- 8 **Effective API Strategy Requires Agile Processes And Governance**
- 9 **Mature API Platforms Cover Five Major Areas**

RECOMMENDATIONS

- 13 **Make Agile-Plus-Architecture The Foundation Of API Strategy**

Notes & Resources

Forrester synthesized its most important research relating to APIs and SOA, telling the broader story to tie our body of research together and fill in gaps between reports.

Related Research Documents

- [Drive Business Agility And Value By Increasing Your API And SOA Maturity](#)
- [Establish Your API Design Strategy](#)
- [How APIs Reframe Business Strategy](#)



APIS AND SOA ARE CRITICAL FOR DIGITAL BUSINESS SUCCESS

Forrester often advises AD&D clients about APIs in the context of mobile apps or some other targeted initiative. Sometimes these conversations extend to open APIs, such as those available from big Internet players like Facebook, Google, Pinterest, and others. It's not as often that we hear API conversations about ensuring the future of an enterprise in a world of digital disruption and the age of the customer. But this broader context is the most important discussion to have about APIs, as described in this report:

[How APIs Reframe Business Strategy](#)

Randy Heffner

Why do APIs rise to the level of business strategy? The need for digital business transformation means that executives — both business and technology executives — must put their organizations in play in multiple ecosystems of value that connect digital resources inside and outside the company, as described in this report:

[The Digital Business Imperative](#)

Martin Gill, Nigel Fenwick

In this context, it is important for AD&D pros and businesspeople to understand that most anything can be a digital resource because either it is already digital (e.g., data, applications, connected devices, etc.) or it can have a digital proxy (e.g., the location code posted by a public transit stop, a Twitter handle that serves as an entry point to the customer service team, etc.). Furthermore, an organization's most important digital resources are its unique core competencies and business assets, which are the foundation of API-enabled business agility. API enablement is a key means of embedding digital agility into everyday operations, which is part of a digital business road map:

[Develop A Digital Business Road Map That Drives Innovation](#)

Martin Gill

APIs feed multiple angles into rapid business change, as described here:

[Selecting Tools That Enable Agility](#)

Martin Gill

APIs And SOA Embody Business Design In Modern Applications

To understand and pursue a mature and effective enterprise strategy for APIs and SOA, AD&D pros should begin by getting two key things clear:

1. **SOA is alive and well, so the strategy should include both APIs and SOA.** Although it may surprise some, SOA is still very much alive, well, and necessary for modern applications. Why is this an important clarification? Because as adoption of APIs and representational state transfer (REST)-based messaging grows, Forrester hears many voices discounting SOA as old, dead, and irrelevant. However, they're using poor definitions of SOA.¹ The reality is that much new work is happening under the banner of SOA. This means that the proper conversation for enterprises to have about service-based strategies is not merely about APIs, but about APIs and SOA together. Furthermore, it means that most SOA best practices have corollary or equivalent API best practices.

However, neither SOA nor APIs have a single definition with which one can go to the market and expect others to be using the same definition. Forrester recommends that clients treat SOA as the part of the strategy aimed at building core business agility and treat APIs as extending the reach of their business agility to many new contexts.² The continuing importance of both SOA and APIs is clear from the enterprise adoption data presented in this report:

Customer Engagement Needs APIs, But REST Is Far From A Complete Integration Strategy

Randy Heffner

2. **APIs and SOA require first and foremost a business design perspective.** While SOA services and APIs can (and should) both be used for technical scenarios such as integration and application delivery, it is critical to understand that your most valuable and strategic APIs and SOA services are those that embody business transactions and queries. These services provide business building blocks (rather than mere technical building blocks) for consistent business results no matter from what customer touchpoint, B2B partner integration, or internal business process a transaction originates.

This means that an effective approach to services will differentiate between different types of services and be quite deliberate about design, coordination, and governance for business services and APIs. This earlier report described the value and impact of a business-focused approach to SOA, and the discussion applies equally to business APIs:

Build SOA Success With A Business-Focused Approach To SOA Design And Governance

Randy Heffner

HOW TO SET STRATEGY FOR APIS AND SOA

Armed with a business-focused and broad-based view that combines APIs and SOA, AD&D pros have the right perspectives from which to begin evolving the rest of their enterprise API strategy. Central to success is an incremental approach embodying what Forrester calls street-level strategy, which has three major steps:

1. **Craft a high-level vision — and stop there.** This is the strategy part of a street-level strategy. Don't write a 300-page architectural treatise about how to do APIs or SOA. Instead, outline key concepts and aspects of strategy at the "mile wide, inch deep" level. Make it only as detailed as needed to a) make stakeholders aware of the breadth of considerations and possible investments; b) identify (but not design) the major processes, delivery patterns, and governance structures; and c) guide just-in-time drill-downs and elaborations as part of step 2.
2. **Do projects, leveraging them for incremental strategy implementation.** This is the street-level part. Use each project to advance and mature the implementation of one or more aspects of the service-based strategy. Use Agile-plus-Architecture practices to collaboratively decide which aspects are most important to each project's success or which practices the project provides an excellent opportunity to develop.³ Don't worry too much if street-level investments don't take you straight toward the vision — some zigging and zagging is inevitable along the way.
3. **Use project experience to adjust the vision.** As you gain more real-world experience with each project, you become much smarter about how to structure the vision and design the architecture to achieve it. So keep adjusting the vision and strategy by returning to step 1.

Understand The Separate Starting Points For APIs And SOA

To help accomplish a first iteration of step 1, this report provides a comprehensive overview of eight major areas of maturity common to APIs and SOA, together with unique aspects of each:

[Drive Business Agility And Value By Increasing Your API And SOA Maturity](#)

Randy Heffner

Although there are major areas of maturity common to both, there are different viewpoints and starting points for APIs and for SOA. Specifically:

- **API strategy starts by understanding four API categories, especially product APIs.** To structure the API side of the strategy, you must start by understanding the four major categories of APIs: open Web, B2B, internal, and product APIs. The first three of these are commonly discussed in the industry, sometimes using the monikers public, partner, and private APIs. The fourth category, product APIs, is not often discussed, but is critical as an alternate perspective into brainstorming possible APIs and business ecosystems. These reports define and describe the four categories, as well as calling out major aspects of API design and key similarities and differences between APIs and SOA:

Establish Your API Design Strategy

Randy Heffner

Brief: Product APIs Create Distinct Customer Value And Opportunity

Randy Heffner

- **The starting point for SOA is a focus on business agility.** To illustrate the continuing value of SOA to core business flexibility, this report tells the story of how SOA was a critical element of success in the merger that created EE, a UK-based telco, including key aspects of how EE approached Forrester's eight major areas of maturity for APIs and SOA:

SOA Plays An Important Role In A Telco Merger

Randy Heffner

ARCHITECTURE SETS THE PROPER CONTEXT FOR DESIGN OF APIS AND SOA

Business-focused design is the most critical design aspect for AD&D pros to center on for APIs and SOA, however, because not all services are business services/APIs, a complete discussion of API design best practices must understand and account for the architecture context around various services. This sets the right foundation for detailed service design.

Put Strong Architecture Context Around APIs And SOA

Forrester's business-centered vision for the future of solution architecture shows how to organize the constantly expanding universe of technology infrastructure and options around business design concepts. At the center of Forrester's vision, business APIs and services embody an enterprise's core digital business capabilities. The architecture context begins by putting a finer point on this central position for APIs:

- **Understand that interface design is the fulcrum of the architecture with APIs and SOA.** The very center of the architecture is the interface design for an API or an SOA service. Interface design, separate from the details of how a service is implemented, is the fulcrum of the architecture. Interfaces are the leverage point and, as with a mechanical lever, their placement (i.e., their designs) relative to other aspects of services are the single most important factor determining whether your API/SOA strategy machine will accomplish what you intended it to and how much work it will require to accomplish the strategy's goals.
- **Set a business-centered solution architecture context around business services/APIs.** To understand how solutions can simultaneously benefit from APIs and a large number of other technology trends, Forrester's solution architecture vision places business APIs in the context of 1) multitouchpoint role-based workspaces; 2) process definitions that control the flow of customer engagement and work across an organization's internal and external ecosystems; 3)

virtualized views of data that unify across siloed legacy and cloud solutions; and 4) a variety of technologies for analytics, rules, collaboration, and more that provide business insight, control, and optimization. This report describes the vision; note that as a vision report intended to reframe enterprise thinking, this report centers on the strategic endpoint, not the street-level path to get there:⁴

The Future Of Solution Architecture: Six Business Design Focal Points

Randy Heffner

- **Extend the business-centered context into the integration architecture.** When APIs and SOA services are applied to integration scenarios, a business-design focus is often left behind. To counter this tendency, we crafted an alternate articulation of our solution architecture vision from the perspective of integration strategy. To this articulation, we apply the term “digital business design” to emphasize the need to recenter away from technical integration software and application silos and toward business design. Again, business APIs and services are central to the vision. As with the previous report, this provides the strategic vision without discussing the street-level path to get there:⁵

Digital Business Design Is The New Integration

Randy Heffner

Establish Strong Design Guidance For A Variety Of APIs And SOA Service Types

Many AD&D pros put too much of their effort about API design into theoretical discussions about REST, including hypermedia as the engine of application state (HATEOAS), nouns versus verbs, HTTP error codes, and the like. These are useful discussions, but they miss, more important concerns about designing comprehensible APIs, ensuring developers understand how to design different types of APIs, and evolving coherent portfolios of APIs. Forrester's guidance begins with the layering needed when designing APIs for mobile, then continues with a comprehensive API design series:

- **APIs for mobile apps require three major layers, each with different design concerns.** It is clear that APIs are needed for mobile, but beyond that, design guidance is often hard to find. The central concept is to plan for three major layers: 1) business APIs, as described earlier in this report; 2) multitouchpoint APIs, which are built to provide common and familiar customer engagement across all of an organization's touchpoints; and 3) touchpoint-specific APIs, which handle specific requirements for presentation and engagement through individual touchpoints. Multitouchpoint and touchpoint-specific APIs are best thought of as being part of the user experience (UX) layer of one's solution architecture. This report provides guidance and clarification, using a point-counterpoint structure to balance API design considerations:

How To Design APIs For Mobile

Randy Heffner

- **Understand that REST APIs are not the only option.** To guide clients in rounding out their comprehensive API design guidelines, Forrester has created a four-part series on API design. Although the industry is currently favoring REST-based APIs, Forrester clients shouldn't fear using simple object access protocol (SOAP) if the situation makes it appropriate. Our API design series begins by positioning various styles of REST-based messaging against SOAP and message-oriented middleware (MOM), including a comparison of pragmatic REST and high-end REST theory:

API Design, Part 1: REST Is The Leading But Not Only Option For Your APIs

Randy Heffner

- **Design APIs for the audience that will use them.** The second part of the series covers a broad set of basic considerations for API design, focusing heavily on REST while touching on scenarios where SOAP or MOM may be a better choice. A key part of Forrester's guidance includes consideration of the target audience for an API, including the possibility that multiple audiences may require different messaging styles. The report also notes multiple alternative styles of APIs, including JavaScript APIs and language bindings via software development kits (SDKs):

API Design, Part 2: Design Messaging Styles By Balancing Reach With Your Other Design Goals

Randy Heffner

- **Design APIs for high quality of service.** With all the discussion about REST being simple and easy, and with so many REST APIs being created for simple, quick, and low quality-of-service (QoS) data access, there is a dearth of guidance in the industry on how to achieve high QoS with REST. To address this gap, transaction management and error handling are key parts of the third report in Forrester's API design series:

API Design, Part 3: Make Transactions And Error Handling Clear In Your API Designs

Randy Heffner

- **Round out design guidelines with security and future-proofing.** The last part in the API design series outlines five major scenarios for trust enablement with APIs (i.e., authentication and authorization), including scenarios for third-party authorization using OAuth2. The other major topic that the report addresses is designing APIs for future change, which requires balancing open-ended design with data integrity:

API Design, Part 4: Future-Proof And Secure Your APIs To Fit Your Usage Scenarios

Randy Heffner

As a final note on API design, the industry conversation about microservices is growing, but there is great confusion over what microservices are and how to design them. The first thing to note is that there are two ways to think about and use the term “microservices”:

- **Microservices as a component-based, container-based structure.** In this definition of microservices, container technologies such as Docker are used to structure an application as a collection of relatively small, separately deployable units.⁶ This use of microservices borrows heavily from early 2000s concepts of component-based development, including the potential for marketplaces for components.⁷
- **Microservices as an API design concept.** In this definition of microservices, a common phrase is “an API should do one thing and do one thing well.” Forrester often hears this notion contrasted with ways that SOA services were created, but in such cases, the reference is almost always to SOA design worst practices. It is still important to have coarse-grained business services (e.g., submitOrder) to achieve consistent results through all customer touchpoints, while also creating very fine-grained services (e.g., a type-ahead API).

These two notions of microservices may be used together or separately. This report puts the two definitions in the context of a major shift in application delivery:

[From Application Design To Application Composition](#)

Kurt Bittner, Michael Facemire

EFFECTIVE API STRATEGY REQUIRES AGILE PROCESSES AND GOVERNANCE

A key reason to have strong design guidance for APIs is that understanding the various types of APIs acts as a foundation for AD&D pros to set the right agile governance strategies. Some APIs require less governance (e.g., touchpoint-specific APIs for mobile) while others require more (e.g., core business APIs). But either way, governance is important because:

- **No governance means an incoherent collection of APIs and negative customer impact.** As we learned with SOA, so it still is with APIs: If application delivery teams simply toss together whatever APIs seem good for their immediate purpose, with no collaboration across teams, they may achieve good results for their isolated applications, but the enterprise as a whole will achieve little, if any, synergy with APIs. Negative enterprise impact may include disconnected and confusing customer engagement across touchpoints, unreliable transaction handling, inefficient and duplicative back-end processes, and unnecessary costs for developing and maintaining duplicate APIs.

- **Agile-plus-Architecture provides a foundation for collaborative success.** These sorts of bad results can be prevented with agile governance structures, developer-architect connections, and multilayered architecture collaboration, which are all best practices for infusing architecture governance into Agile and continuous delivery methods. With Agile-plus-Architecture, delivery teams gain the requisite context to design APIs that fit within a broader portfolio, and architects' activities are better focused on the near-term context and needs of delivery teams. This report describes more than 30 Agile-plus-Architecture best practices across four major categories (business architecture and project context, project delivery guidance and governance, architecture management and technology selection, and organization and culture):

[Best Practices For Agile-Plus-Architecture](#)

Randy Heffner

This report provides background and support for Agile-plus-Architecture through Forrester survey results that highlight ways that architects and developers can work better together:

[A Guide To More Effective Developer-Architect Relations](#)

Randy Heffner

- **Service portfolio management is a key governance discipline.** Among the SOA best practices that can help to structure Agile-plus-Architecture collaboration for APIs is service portfolio management, which guides service design and evolution using lightweight definitions of target service portfolios. Teams use service portfolios to identify when to build new APIs and to more reliably identify which APIs should be used on any given project. In addition to being an overall API/SOA best practice, service portfolio management is often a key responsibility of a center of excellence (CoE), as described in these earlier reports:

[Survey Results Show SOA Governance Improves SOA Benefit Realization](#)

Randy Heffner

[The Five Most Valuable SOA Governance Practices](#)

Randy Heffner

[SOA Centers of Excellence: The Five Most Valuable Practices That Keep SOA On Track](#)

Randy Heffner

MATURE API PLATFORMS COVER FIVE MAJOR AREAS

Although API management is the hot new product category for APIs, a mature API platform requires much more than API management. Forrester identifies five major elements of a mature API platform:

- **Service delivery infrastructure.** This area covers a broad array of alternatives for where and how an API or SOA service implementation runs. The business logic of the API may reside within a vintage application on a mainframe, on a Node.js server, on a traditional application server (e.g., Java EE, .NET), in an off-the-shelf application (whether on-premises or cloud), or on other platforms.⁸ The business logic may be accessed directly (e.g., direct call to a Salesforce or SAP API), through integration software (e.g., integration platform-as-a-service [iPaaS], enterprise service bus [ESB]), or through an application gateway.⁹ Each of these different runtime environments will have its own built-in or other available development tools.
- **Service testing and virtualization.** Every API must be thoroughly tested on its own, separately from any application that calls it.¹⁰ This includes both functional testing and performance testing. In addition, it is important to be able to retest APIs on demand as part of regression testing and production problem resolution. Finally, it facilitates Agile and continuous delivery for teams to be able to test using a virtual service (i.e., a simulated implementation of a service) so that teams that are using APIs and creating APIs don't have to proceed in lock-step.
- **API management.** This element centers on the relationships between API users and API providers. API users may be internal or external to the API provider's organization. External API users may further be categorized as individual developers or B2B trading partners. API management solutions include a developer portal, API product manager features, security and policy enforcement capabilities, and functions for adapting an existing API/service the way the provider desires it to be used.
- **Service runtime management.** This element ensures top-quality API operations by monitoring and managing APIs not only at the edge where API requests and responses traverse but also across the various layers of infrastructure and APIs behind the API interface.
- **Service life-cycle management.** Aside from managing usage and operations of APIs/services, a mature API program will have defined and managed life cycles for creating various types of APIs and services.

The reasons for and interactions among these last three platform elements (service runtime management, service life-cycle management, and API management) are described in this report:

[How To Manage APIs For Customer Engagement](#)

Randy Heffner

Diverse Enterprise Requirements Feed A Diverse API Management Market

Among these five elements of API platforms, the newest kid on the block is API management. Because APIs themselves are used in many diverse scenarios, there is room in the market for a variety of styles of API management solutions from a variety of vendors. Forrester fields many client inquiries on the space because:

- **API management is a rapidly growing market with four major segments.** In our sizing of the API management market, we identify four segments: API management pure plays; vendors that combine API management with integration and platform-as-a-service (PaaS) offerings; major services vendors with defined API management offerings; and vendors with platforms for API management, governance, and SOA:

[Sizing The Market For API Management Solutions](#)

Randy Heffner, Michael Yamnitsky

- **API management solutions vary widely.** Because of the wide diversity across API management solutions, Forrester published two corollary reports with our Forrester Wave™ evaluation for API management. These reports provide clients with additional insight and resources for selecting among the vendors. Clients can download the detailed Forrester Wave evaluation spreadsheet by mousing over the Forrester Wave graphic to bring up the download link:

[The Forrester Wave™: API Management Solutions, Q3 2014](#)

Randy Heffner

[The API Management Buyer's Guide, Q3 2014](#)

Randy Heffner

[Applying The Forrester Wave™: API Management Solutions, Q3 2014](#)

Randy Heffner

Note that since the Forrester Wave published, at least five additional product vendors have entered the market with API management offerings: Microsoft, Oracle, SAP, Software AG, and WaveMaker. Forrester will be providing an initial look at these solutions in an upcoming API management market update report, which will also note key systems integrators with well-developed API management implementation offerings, such as Torry Harris Business Solutions.

Other API Platform Elements Include Both Active And Mature Market Spaces

For the other API platform elements, Forrester's coverage for AD&D pros is guided by the level of maturity versus active product development within the market space. Here are some key themes for other API platform elements:

- **Application gateways continue as an active space aside from API management.** Every API management solution includes embedded gateway functionality, but application gateways also continue on as a standalone market space. We will be updating our Forrester Wave in this space in late 2015 or early 2016. Since our most recent Forrester Wave, we've seen these changes to the vendor landscape for gateways include 1) CA Technologies bought Layer 7; 2) Axway bought Vordel; 3) Tibco Software has rewritten its gateway into a more competitive offering; 4) Progress Software's gateway is part of the Actional product, which Aurea bought, and 5) Software AG no longer resells Layer 7's gateway:

[The Forrester Wave™: SOA Application Gateways, Q4 2011](#)

Randy Heffner

- **API/service testing is critical, though often undervalued by enterprises.** As with other software, APIs require both functional and performance testing. In addition, service virtualization allows API users to proceed with development without waiting for APIs to be developed and enhanced. This parallel development facilitates delivery speed. However, Forrester often observes that clients do not place a high enough priority on tools for service testing. These two Forrester Waves provide a foundation for API testing tool selection:

[The Forrester Wave™: Modern Application Functional Test Automation Tools, Q2 2015](#)

Diego Lo Giudice

[The Forrester Wave™: Service Virtualization And Testing Solutions, Q1 2014](#)

Diego Lo Giudice

- **The term “API management” wrongly implies that the category covers runtime management.** API management solutions are limited when it comes to providing top-quality production operation of APIs and SOA services. On their own, API management solutions can provide only surface-level visibility into service operations. Service runtime management provides deeper visibility into the layered implementations common for APIs and services. This is why, even before the rise of the industry API discussion, service runtime management showed signs of being the most beneficial specialty product category for services:

[SOA Product Adoption: SOA Management Solutions Provide The Strongest Benefit](#)

Randy Heffner

- **Service runtime management is a relatively mature product category.** Because of this maturity, we will not be updating our most recent Forrester Wave of the space. The rate of change in the market is low. Since we published this Forrester Wave report, the most significant change is that Progress Software sold its Actional product (and other products) to Aurea. Software AG continues to OEM Actional as webMethods Insight. SOA Software has renamed itself to Akana.

The Forrester Wave™: Standalone SOA Management Solutions, Q4 2011

Randy Heffner

- **Service life-cycle management is mature, excepting its inclusion in API management.** As with service runtime management, we will not be updating our most recent Forrester Wave (but the functionality of life-cycle management appears as a part of our API management solutions Forrester Wave). The two changes to the vendor landscape are that 1) SOA Software is now Akana and 2) Progress Software's solution was discontinued:

The Forrester Wave™: SOA Service Life-Cycle Management, Q1 2012

Randy Heffner

We will include vendor and product guidance for API/service delivery infrastructure in an upcoming guided tour for integration architecture.

RECOMMENDATIONS

MAKE AGILE-PLUS-ARCHITECTURE THE FOUNDATION OF API STRATEGY

Across all of Forrester's advice for APIs, SOA, and service-based strategy, the best starting place for AD&D pros building an API strategy is to ensure an Agile-plus-Architecture foundation. It is only by combining the two that an enterprise can meet and sustain the speed-of-delivery promises of APIs and services. Furthermore, only by combining the two can an enterprise evolve coherent portfolios of APIs and services within each major business domain. Each report referenced above has other important recommendations for AD&D pros relevant to its specific topic area, and every one of these separate recommendations will deliver better results within a street-level strategy grounded in Agile-plus-Architecture best practices.

ENDNOTES

- ¹ More evidence that SOA is alive and well is the number of conference sessions on or referencing SOA at major vendor events. After one IBM event, Forrester described what we saw. Source: Randy Heffner, "Sorry, Kids: APIs Have Not And Will Not Kill SOA," Randy Heffner's Blog, May 2, 2014 (http://blogs.forrester.com/randy_heffner/14-05-02-sorry_kids_apis_have_not_and_will_not_kill_soa).
- ² Forrester describes the different ways to relate the terms "SOA" and "APIs" as a foundation for understanding API and SOA maturity. See the "[Drive Business Agility And Value By Increasing Your API And SOA Maturity](#)" Forrester report.

- ³ Organizations must be able to sustain and increase their rate of change over time. Agile development practices and continuous delivery are essential tools, but so is an architecture that enables resilience. Combining Agile and architecture is challenging because their respective goals of delivering now and preparing for the future often appear to be at odds. Forrester outlines the challenges that developers and architects face in trying to collaborate, and identifies important resources that both sides can draw upon. See the [“Agile-Plus-Architecture: Embrace The Oxymoron”](#) Forrester report.
- ⁴ Two separate reports provide perspectives and decision models for street-level strategy for evolving toward Forrester's vision for the future of solution architecture. See the [“The Future Of Solution Architecture, Part 1: Business Processes Within A Capability”](#) Forrester report and see the [“The Future Of Solution Architecture, Part 2: User Roles Within A Business Capability”](#) Forrester report.
- ⁵ A separate report provides a perspective on street-level strategy for evolving toward Forrester's vision for digital business design. See the [“How To Implement Digital Business Design”](#) Forrester report.
- ⁶ There is a quiet revolution underway in software development that leverages openly available services fronted by APIs, service-rich platforms, and deployment technologies like microservices and containers. See the [“From Application Design To Application Composition”](#) Forrester report.
- ⁷ It is clear that OS-level containers are beneficial for application architectures and life cycles, but several important questions remain about Docker. Forrester clarifies for app developers the most significant things that are known about Docker and the critical questions that remain. See the [“Nine Questions To Ask About Docker”](#) Forrester report.
- ⁸ JavaScript and the Node.js runtime environment in particular are becoming an increasingly important part of an enterprise environment. See the [“The Dawn Of Enterprise JavaScript”](#) Forrester report.
- ⁹ Forrester's most recent Forrester Wave for enterprise service buses will be superseded by one or more Forrester Waves including cloud integration and other options for application integration. Since we published our last ESB Forrester Wave, we've seen many changes to the vendor landscape: 1) Red Hat bought FuseSource, 2) Progress Software sold its Sonic ESB to Aurea, and 3) IBM has combined its ESB offerings into one product, IBM Integration Bus. For more information, see the [“The Forrester Wave”: Enterprise Service Buses, Q1 2009”](#) Forrester report.
- ¹⁰ With increasingly complex applications and layering, it is important to automate tests at all layers of an architecture by going beyond GUI automation testing to testing at the API, service, and process levels. See the [“Five Must-Do's For Testing Quality At Speed”](#) Forrester report.

About Forrester

A global research and advisory firm, Forrester inspires leaders, informs better decisions, and helps the world's top companies turn the complexity of change into business advantage. Our research-based insight and objective advice enable IT professionals to lead more successfully within IT and extend their impact beyond the traditional IT organization. Tailored to your individual role, our resources allow you to focus on important business issues — margin, speed, growth — first, technology second.

FOR MORE INFORMATION

To find out how Forrester Research can help you be successful every day, please contact the office nearest you, or visit us at www.forrester.com. For a complete list of worldwide locations, visit www.forrester.com/about.

CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866.367.7378, +1 617.613.5730, or clientsupport@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.

Forrester Focuses On Application Development & Delivery Professionals

Responsible for leading the development and delivery of applications that support your company's business strategies, you also choose technology and architecture while managing people, skills, practices, and organization to maximize value. Forrester's subject-matter expertise and deep understanding of your role will help you create forward-thinking strategies; weigh opportunity against risk; justify decisions; and optimize your individual, team, and corporate performance.