API Strategy: Design for Analytics
Accenture’s 2013 Technology Vision identifies “Design for Analytics” as a key principle in using analytics to uncover critical business insights. Accenture research finds that half the time companies lack the right data needed to answer critical questions. Design for Analytics starts with the business questions, and then designs the gathering of data to feed into analytics to supply the answers. API Analytics applies this principle to ensure the success of an application programming interface (API) program.

The API as a Product

Today’s APIs are business-level artifacts. They provide the channel to scale business, offer elements to improve innovation and agility, and even serve as a representation of brand. Analytics play a critical role in ensuring the success of APIs by providing the necessary insights to assess value, measure success and identify how to improve performance.

API usage is the core driver of the success of any API program. In fact, API Analytics looks at APIs as if they are products, which is a key mindset to adopt. Thinking of an API as a product engenders related questions. Is it the right product for the purpose? Is it functioning the way it should? Are people using it in the way it was designed and intended to be used? Is it easy to use and differentiated from other API products?

To address these questions, API Analytics goes beyond traditional trend and volume reports that only divulge what happened, and answers more mission-critical questions, such as what is happening and why. For instance, early detection of low API volumes, combined with visibility into whether the problem stems from IT issues or from confusion due to poor documentation, allows for more speedy and effective resolution before the problem becomes widespread.

API Analytics delivers the insights to improve an API product in terms of design, functionality, support and offered terms in order to achieve program success.

Start with the Insights

Design for Analytics starts with identifying the insights needed to guide a successful API program.

Insight into what makes a good API product varies based on the perspective along an API value chain, including views from the professionals who design and operate the APIs, as well as from application developers who use the APIs.

API product managers oversee an API product and need to meet certain business goals. While the goals themselves may vary—from increased efficiency, to sparking innovation, to pure monetization—a common thread is that success hinges on making the API product appealing to consumers. For example, it should make things easier for consumers and be something they want to use and reuse.

This, in turn, raises questions about the end consumer. Who are the application developers who build the APIs into applications? Are they satisfied? What types of applications are they building? Companies may look for answers using the traditional, fixed reporting on volumes and trends that serve to identify popular APIs and heavy users; however, these insights are not enough because they only identify what happened.

A better approach is to use machine learning to assist in the real-time detection of access patterns defined by the timing and ordering of the API calls that an application makes. Consider the example of an online store where there is a specific order of API calls: catalog, shopping cart, account, shipping, payment. Access patterns that deviate from this order may signal incorrect use, whereas a pattern that accesses the catalog in quick succession may indicate a case of data extraction. With this insight, the API product manager can determine whether to stop allowing this activity or to promote it as part of an opportunity to monetize. Understanding what is happening and why helps the product manager get a holistic view of the business effectiveness of the API and take the appropriate proactive action.

On the API operations side, IT focuses on keeping APIs running and meeting service level agreements (SLAs). This focus requires real-time insights to troubleshoot performance and security issues, and also long-term insights for capacity planning.

Similar to the API product manager, IT can leverage machine learning to identify events and patterns that are undetectable by fixed reports or human inspection. However, IT also has to move quickly to identify issues, troubleshoot root causes, and apply fixes to minimize performance degradations or even outages.

For example, suppose there is an unexpected surge in API volume. Insight into the context helps determine the right response. If the calls seem suspicious, IT can request re-authentication, mask sensitive fields or block access altogether. In the case of legitimate use, IT can make a choice whether or not to scale the backend systems to meet the demand, depending on if the traffic is from test systems or from production systems.
Application developers are the direct consumers of an API product. They determine whether or not to use an API within their applications by evaluating API functionality, reliability and ease of use. If one or more of these factors fail the test, application developers may switch to another option or forgo the API altogether.

For better insight, application developers and API providers need to help each other. Consider differentiating an API product by offering application developers direct visibility into performance and availability, along with explicit notification of issues or degradation. This insight helps application developers rule out provider issues in troubleshooting their applications.

In return, application developers can collect error logs, communicate use cases and embed the capability into the application to trace an API call from the deployed application. Working together allows the API provider to offer better support and quicker resolution of issues.

Design for Analytics

Given the desired insights, Design for Analytics explicitly designs for the answers identifying the needed data and how to get it.

Exploring general perspectives along the API value chain identifies a number of desired insights. These include availability and performance monitoring, volumes and trends, access patterns, and visibility into the API end-to-end conversation. For example, instrumentation across the vantage points within the API call gathers the data needed to formulate these insights. Metrics can be derived from the deployed application (whether on an end-user mobile device, server or cloud-based system), through the API management gateway that processes the API request and response, into the backend resources that serve the data and services.

The takeaway is to not confuse access to metrics as sufficient to measure and evolve an API program. Instead success starts with identifying the questions and then designing the analytics to obtain the right data to answer them.

API Analytics Maturity

Companies getting started with API enablement may not see the immediate value of API Analytics. When the number of APIs is small and the use cases are well-known (as when supporting one internal application and mapping API success is strongly correlated), analytics may not seem necessary. But as API use opens up and more APIs are created, organizations will quickly require a more industrialized model for analytics to properly measure and ensure success.

We introduce a tiered maturity model (lowest equals 1 to most mature equals 5) of API Industrialization for Analytics (see Figure 1):

1. Ad-hoc: Initial API forays often start with ad-hoc API development and siloed use within parts of the organization. There is no standard use of analytics or measures.

2. Organize: Organized API development is driven by an API strategy and business case. When there is identifiable business outcome, it requires a standard definition of the metrics and measures needed to quantify success.

3. Tactical: At the tactical level, a common organization directs API projects by working out standard cost-benefit analysis along with pervasive end-to-end instrumentation needed to generate visualization and reporting.

4. Critical: Implementation of mission-critical services occurs via a mature API design, development process and platform. In critical systems, companies integrate API analytics with IT systems for insight-to-action automation needed to support and evolve the API program.

5. Industrial: Industrialized programs use APIs as the fabric of business operations and the connected enterprise. These programs leverage predictive models and visibility beyond the API program to understand what is happening and why.

As the level of maturity increases, API Analytics leverages a greater abundance of data guided by increasingly automated analytics to pick out relevant insights. This answers questions such as: How to effectively allocate resources and introduce updates (e.g., add/remove APIs, monetization, versioning)? How to factor in changes in usage? Is the program meeting the defined goals?

API Analytics ultimately becomes more proactive in helping to adjust the API program—and guide its success. It is an ongoing process and not a one-time assessment. And it all starts with Design for Analytics.

Figure 1: Tiered maturity model of API Industrialization for Analytics

<table>
<thead>
<tr>
<th>Phase 0</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad-hoc</td>
<td>Organize and plan</td>
<td>Tactical implementations</td>
<td>Mission critical</td>
<td>Industrialized API</td>
</tr>
<tr>
<td>First API projects</td>
<td>Defining business case: objectives, use-cases and value proposition</td>
<td>Planning for API platform</td>
<td>Emphasis on strategic and business services</td>
<td>API as the fabric of business operations</td>
</tr>
<tr>
<td>Expose services as needed-basis</td>
<td>API readiness assessment Planning for API transformation</td>
<td>Centralized infrastructure and security</td>
<td>Mature API platform</td>
<td>Use of operational analytics to continuously optimize and evolve API</td>
</tr>
<tr>
<td>No coordination across groups</td>
<td>Cross group governance</td>
<td>Emphasis on developer life cycle</td>
<td>Integrated design and development using API tools</td>
<td>End-to-end automation</td>
</tr>
</tbody>
</table>
For more information about API Analytics, please contact:

**Teresa Tung**
teresa.tung@accenture.com

**Edy Liongosari**
edy.s.liongosari@accenture.com

**Anthony Harris**
anthony.e.harris@accenture.com

### About Accenture Technology Labs

Accenture Technology Labs, the dedicated technology research and development (R&D) organization within Accenture, has been turning technology innovation into business results for more than 20 years. Our R&D team explores new and emerging technologies to create a vision of how technology will shape the future and invent the next wave of cutting-edge business solutions. Working closely with Accenture’s global network of specialists, Accenture Technology Labs help clients innovate to achieve high performance. The Labs are located in Silicon Valley, California; Sophia Antipolis, France; Arlington, Virginia; Beijing, China and Bangalore, India.

For more information, please visit: www.accenture.com/accenturetechlabs

### About Accenture

Accenture is a global management consulting, technology services and outsourcing company, with approximately 266,000 people serving clients in more than 120 countries. Combining unparalleled experience, comprehensive capabilities across all industries and business functions, and extensive research on the world’s most successful companies, Accenture collaborates with clients to help them become high-performance businesses and governments. The company generated net revenues of US$27.9 billion for the fiscal year ended Aug. 31, 2012. Its home page is www.accenture.com.