Commentary

Application Architecture Standards/Guidelines: Case Study

In a recent multiclient case study, 12 Gartner clients shared their application architecture standards for their most recent development projects and told us what worked for them and why.

During a sustained period of working with Gartner clients on application architecture, we noted consistent success by those that developed standards and guidelines. This made the process of architecture easier and more consistent, project to project. Enterprises that learned from each architecture effort and applied those learnings to new projects were more successful. These shared best practices from our study examine what comprises effective standards and guidelines.

Definitions

Standards contain the mandatory and optional requirements that ensure a uniform approach to design and development.

Guidelines facilitate the design and development by providing organization-specific "how-tos" and best practices.

It is important to note that effective standards and guidelines are only one factor in an organization's ability to successfully manage its architecture. All study participants had a number of governance and technical architecture efforts under way to improve their overall architectural capability (see "Architectural Maturity: Acting on the Signs").

Standards

All 12 enterprises that participated in the study maintained some level of enterprise architecture standards. Eight of the 12 enterprises believed that their standards were up to date, while the remaining four were not sure or believed that the standards had lagged their current architecture.

More than half of the participants used a common root document that contained references to all application architecture standards and mapped standards to project or application areas.
Table 1 shows a sample framework that maps three typical types of projects to applicable standards. These projects include:

**Multienterprise applications:** Includes order, inventory, and other business-to-business and enterprise-to-enterprise applications. These applications are typically XML-centric and heavy users of enterprise application integration (EAI) services.

**Customer-facing Internet:** Comprised of customer shopping cart and customer portal applications. These applications are typically integration-intensive, often integrating with horizontal portals, other Web applications and EAI services.

**Corporate Intranet:** Comprised of human resources and benefits administration applications. These applications typically feature richer user interfaces, locked-in browsers and minimal enterprise integration requirements.

Against these typical application project types, we show four common standards.

<table>
<thead>
<tr>
<th>Projects/Applications</th>
<th>Multienterprise</th>
<th>Customer-Facing Internet</th>
<th>Corporate Intranet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configuration Management</strong></td>
<td>Medium importance</td>
<td>Very important</td>
<td>Medium importance</td>
</tr>
<tr>
<td>Java/J2EE/WebSphere applications standard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Middleware Integration</strong></td>
<td>Low importance</td>
<td>Very important</td>
<td>Low importance</td>
</tr>
<tr>
<td>Java to MQ Series standard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Corporate Data Warehouse</strong></td>
<td>High importance</td>
<td>Medium importance</td>
<td>Low importance</td>
</tr>
<tr>
<td>Data update and access standards</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Interoperability</strong></td>
<td>High importance</td>
<td>Medium importance</td>
<td>Low importance</td>
</tr>
<tr>
<td>XML/XSLT standards</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Gartner Research

We noted four key best practices for architectural standards:

- To be effective, standards need to list specific technology versions and relevant application programming interfaces (APIs) or utility versions. Examples include specifying database management system product version, Java Database connector versions, and XML version and encoding standards.
- Standards must define prerequisite and post-requisite tasks. Examples include account creation, ordering appropriate licenses, and any internal reporting or notification requirements.
- Standards must provide a verification checklist that details the review or compliance process.
- Standards document how the standard is maintained. Standards have to be reviewed and updated periodically.
Guidelines

Six of the 12 participants maintained some level of architecture guidelines. Four of these six felt strongly that guidelines were instrumental in increasing project quality. Best practices discovered included:

- Guidelines facilitate the design and development by providing organization specific "how–tos" and best practices. Guidelines do not replace technical documentation, but put the guideline subject in context with the organization environment.

- Good guidelines focus on the specific common problems faced by designers and developers. Most participants identified common "pain points" with their latest projects. These included integrating applications with EAI, horizontal portals and version control.

- Guideline topics can cover all the technology and process areas of a project. Guidelines cover technology uses, but can also cover processes. For example, guidelines that cover the built process and peer-review process were found to be very useful.

- The most-effective guidelines were very short and very specific. Guidelines should be limited to four pages.

Guidelines include an example. The example can be a code snippet or a command example. Guideline examples include:

- **Java access to enterprise sales database**
  
  This guideline demonstrates how to connect from a Java environment to a corporate database using the applicable standards.

- **Using enterprise EAI in .NET**
  
  This guideline shows how to use the required APIs and gives examples on how to list services, subscribe to services and incorporate services into an application.

- **Configuration management: Checking your work in and out**
  
  Describes the steps and commands involved in checking work in and out of a version-control system. This guideline provides examples for managing all project artifacts, including source code, application configuration files and database creation scripts.

- **Configuration management: The release process**
  
  Describes the process and procedures for releasing new application versions, depreciating previous versions and rolling back applications when required.

**Bottom Line:** Organizations that are successful at enterprise architecture have standards and guidelines that aid development teams. Consistent use of these standards and guidelines will lead to better project outcomes.