The Future of Application Development

VI Annual Enterprise Integration Summit

Joseph Feiman

April 10-11, 2007
WTC Hotel
São Paulo, Brazil
The Future of Application Development

The goal of the estimating process is to know as much about the project as is possible at any given time. Documenting what's known and unknown will enable changes to the estimate to be tracked and justified.

Agility has become a key for most development organizations today. With the rapidity of business change accelerating, and with the increasing need for software to support those changes, application development (AD) organizations are faced with the need to do something different. AD productivity has been increasing steadily, but at a rate that simply won't support the even-more-rapid need for business change.

Something different requires attention to the usual three key areas: people, process and technology. In the past, typical improvements in development productivity have resulted from technology, through new generations of programming languages or new development tools. Today's radical changes will require attention to the first two areas, which have typically been the most difficult for development organizations to address.

The real keys to unlocking the agility paradox are:

• Architecture
• A focus on software process and engineering
• Reuse — of everything

In the past, we've been builders of custom software or deployers of packages. In the new, agile AD, we'll find that reuse and assembly will be the keys. AD organizations can't code themselves into the future.
Key Issues

1. How will business and technology trends affect AD organizations during the next five years?

2. Which technologies and vendors will provide leading solutions for applications delivery during the next five years?

3. Which management approaches and practices will enable AD organizations to succeed in delivering high business value during the next five years?

4. What organizational changes are necessary to deliver applications to business users through 2009?

During the past couple of years, we have seen little innovation making its way into the business portfolio of application software. New application software approaches, such as services-based applications and business process fusion, were still in their infancy — few utilities or standards were in place to enable the infrastructure to support these approaches; therefore, applications must wait on infrastructure. Application software vendors are among the businesses that depend on reliable infrastructure to build and deploy their applications. Most vendors have been re-architecting their platforms and applications during this period.

Companies are also still grappling with extracting value from their installed base of applications software and using it to meet current business demands (transparency, real-time processing and productivity). We expect a focus on better execution — getting what's in place working better, getting it connected and, finally, getting a payback. Better process execution implies the following:

• Businesses will enable critical processes with real-time management and real-time enterprise (RTE) capabilities.
• They will make progress toward end-to-end processing and visibility.
• Work will be done to continually tune and better use established software.
• Businesses will make progress toward improved corporate governance and transparency.

By 2006 and 2007, activity will begin in earnest to implement the next generation of applications.
AD Organizations: Variations on a Theme

Organizational variations:
- Functional (development/maintenance)
- Skills (programmer, analyst, tester)
- Pyramid (by business group)
- Centralized/decentralized

Organizational imperatives:
- Client relationships
- Consistent development processes
- Strong HR processes
- Metrics to demonstrate success

The "best in class" AD organization

What is "best in class"? For AD organizations, it's not "highest productivity," "lowest cost," "best quality" or a combination thereof. Rather, best in class involves meeting the business needs each and every time, in all the dimensions we previously discussed. In other words, there's no exact profile for best in class. It's much the same for AD organizations. We've seen "boxed" organizations succeed wildly — and fail spectacularly. It's the same for pooled organizations, and for those who attempt to segment based on work type.

What is important, then? The single most critical issue in designing an AD organization is to clearly understand the functions being performed, the roles and skill sets involved and the responsibilities that must be lived up to. This requires several key success factors:
- Development organizations must have strong relationships with their clients codified into service-level agreements.
- Have clearly documented processes — one for each major type of work they do.
- Have a clear HR management process, in which skill sets are planned, cultivated and understood.
- Use a combination of process, skills and roles to staff their projects (typically, about 60% of their work).

Quite often, the culture of the AD organization, or the culture of the business itself, will define the style of the organization. That's fine; attempting to organize in one fashion when corporate culture clearly favors another will cause more problems than its solves. Clarity of function, process, role and skill, then, should be the mantra of those who are designing an AD organizational structure. Once those are in place, any of the models we've talked about will work, and work well.
Application portfolio management (APM) is often addressed as simply an IT-centric initiative. This is wrong. APM needs to be understood across the enterprise. APM requires a fundamental change in the culture or philosophy of an organization. Because this is a continuous process, organizations have to accept this as part of their belief system. APM is a process; results are best seen over time. Initial cost savings are there, but the value is a more-disciplined approach to managing applications over time. Most of the APM approach is designed to better manage costs of the largest part of most application portfolios — the utility applications. It can also be used to manage future investments in IT spending, particularly for identifying how established applications can fit into future architecture decisions.

The broad notion of APM can be distinguished by two different goals. For enhancement and frontier applications, making appropriate investment decisions is the goal. These types of applications have large payoffs for many businesses, but organizations are often faced with myriad choices. For utility applications or infrastructure, the issues are more likely to be cost management. Often, a more-refined level of granularity is required, and automated tools for collecting potentially a large volume of information are necessary. Investment decisions are based on business value or opportunity and are often about managing a variety of options in the future.
The CIO priorities for 2006 reflect a clear understanding of business expectations. They focus on business growth projects, improving business alignment and measuring IT organizations' business contribution. These priorities align with business expectations of growth, process improvement and building competitive advantage.

Delivering against those priorities requires attracting, upgrading and retaining business skills in information systems (IS). Why? Because IS needs these business skills to deliver the differentiated solutions that drive enterprise growth and competitiveness. An increased need for business skills illustrates how the world is changing IS. To address the business skills issue, CIOs plan to: 1) Raise the business skills in IT through training or selected recruitment; 2) Attract and retain key IT personnel who are able to deliver business results and manage technical complexity; 3) Upgrade IS scorecards and replace technical performance metrics with business contribution measures.

The growth imperative was particularly strong in 2006, with:
• 70% of CIOs agreeing that "delivering projects that enable business growth" is important.
• 57% of CIOs were concentrating on delivering this type of project in 2006, and they see themselves as doing this from 2006 through 2009.
The common belief is that, to improve their agility, enterprises must purchase new technology. This is not true. The ability to be agile also involves optimized use of established technologies. In addition, reducing the failure rates of an IS organization in implementing solutions can improve agility as much as the addition of a new technology or a new business process. Willingness is calculated by examining a company's commitment to getting the right people, paying the right money and allotting time correctly, then comparing that to the company's ability to be effective in IT. Stronger effectiveness in IT (that is, leverage) improves the agility rating, because it potentially means that fewer people and less money and time can be allocated to solving a particular problem. Alternately, IS organizations with a high risk of failure in project implementations will reduce overall agility, because failure will tend to increase the need to spend money, people or time to solve problems.

**Action Item**: Do not "throw" people or money at projects to improve agility. Instead, apply money to improving IT effectiveness and reducing day-to-day risk of implementation failure.
Business process management (BPM) requires stakeholders to reduce the importance of functional unit-level performance targets and focus on shared process improvement objectives. The goals for BPM initiatives generally fall into six patterns: 1) reducing the latency of operational business processes because time is money; 2) standardizing or institutionalizing best practices because requisite experts are expensive, and the learning curve is steep; 3) eliminating gaps at handoff points, where work crosses real or artificial boundaries; 4) reducing errors as work progresses and crosses boundaries to reduce loss between applications, functions, departments, buildings and even companies; 5) matching HR to demand by leveraging alternative sourcing strategies; and 6) documenting work processes and decisions to meet compliance mandates. For example, in financial services, many products are rapidly commoditizing (such as mortgages) as information is increasingly transparent to consumers. Many industry participants have made major investments in their loan application processes to streamline the process, improve the integrity of form-driven data collection and sustain applicant interest by closely managing their interactions with applicants. Once goals become clear, stakeholders can collaborate on how best to achieve the goal.

Action Item: Implementing BPM is difficult. The main problems to any significant change are the human barriers — inertia and vested interests.
With a business process management suite (BPMS), the full business process is made explicit in a graphical process flow model. This process model is completely autonomous from the resources performing the work steps, whether they are human or machine resources. By making it explicit and autonomous, changes to the process model can be made independently from changes in the resources. This is the "loose coupling" principle familiar to many from earlier middleware forms and from service-oriented architecture (SOA) design guidelines. Making process flow control explicit and decoupling it from the underlying technology allow processes to be changed more quickly. For example: Processes are easily changed because other system elements are not affected by flow control changes and need not, therefore, be retested. Some process changes are made by business professionals who need only limited knowledge of IT systems. In a growing number of cases, changes such as work item routing, business rule overrides, and parametric changes to approval levels, are made in real time. Near-real-time reporting of process steps deliver never-before-seen analysis of current operating conditions customized as appropriate to the organization. Tightly coupling business dashboards/cockpits/business activity monitoring (BAM) to the underlying runtime allows rapid and precise process change. Orchestration engines are the runtime environments. There will be different kinds of runtime environments — some for Web-service-based components, others for human workflow components, and others for rules. The process model is simply an XML metadata description of how all the activities and events should be coordinated. And because it is XML, it can be made executable at runtime, and the resources that perform the steps can be dynamically late-bound into the execution.

Action Item: The separation between process design and process implementation and execution (a separation not found in conventional applications) will permit the emergence of a market for "process components" — a new kind of intellectual property. These components will include templates for specific horizontal and vertical processes, business content, and rule sets.
To Be Agile You Must Change!

Keys for Change:
- Context for change
- Clearly defined processes
- Marketing and communication plan
- Rollout plan
- Incentives

Drivers for change (agility, cost, consistency, and so on)
Organizational vision (future state)
Current process models (behaviors), if any
Goals and objectives (metrics)

Fundamentally, changing an organization begins with the realization that something's not right. This "level of dissatisfaction" could be as a result of measurements, hallway conversations, formal satisfaction surveys, and so on. What's needed is a change to make the feeling disappear. In the absence of a clear vision of what will be (a clear organizational vision and set of architectural principles), literally any change will do.

Next, the "burning platform" for change must be established. This platform is the basis for why each individual in the organization must change rather than remain in their current behavior pattern. Given this impetus for change, a new set of behaviors can be codified into processes.

Finally, measurements must be established, along with personal goals, objectives and other incentives. Basically, an organization is attempting to break current behavior and establish new ones. Without a strong incentive process, it's unlikely any of these new behaviors will be practiced long term.
What Is the Business Process Platform, and Why Should AD Care?

Definition: The business process platform is an organization’s IT model that enables more-rapid and controlled business process change through the use of an integrated platform of business process composition technologies and business process components or services.

Enabling business process composition requires an organization IT model that combines common technologies that are made up of business processes, business process components and cultural change. The business process platform is an organization IT model that leverages emerging technologies along with cultural change to enable rapid and controlled business process orchestration and reorchestration. Although many technologies are available today to enable business process composition, in most cases, these technologies must be acquired from different vendors and integrated by user IT organizations. Additionally, the content required doesn't exist in the open market. In the future, we will begin to see the emergence of major vendors that will provide integrated composition technology platforms and technologies, enabling users to populate and manage a container of business components/services. But the content and technology composition must still be combined with cultural change, which brings IT and business closer together in defining the process requirements and the process rate of change. Users will never buy a complete business process platform from a single vendor, but, rather, will buy the technologies to enable orchestration and populate the container used for composition content.
**What Is a Business Process Management Suite?**

- A development and runtime environment that enables *process* modeling and design, development and execution, and ongoing management and optimization
- Automates and manages the end-to-end flow of work as it progresses *across boundaries* — people, systems or value-chain boundaries
- People, process and information are *all first-class objects*
- Unifies previously independent software infrastructure categories (such as workflow, EAI, document/content management, portals, Web servers and application servers)
- Supports SOA principles and XML Web services standards

---

**Warning: A BPMS is one of many emerging technologies that can be used for creating SOA composite applications.**

BPM refers to a concept and technology. The particular processes may have a direct or indirect effect on the chosen metric of corporate performance. Direct is better. A goal might be to establish service-level agreements (SLAs) for transactions with best partners. The strategy to do that might be by fulfilling orders from best partners first and then tracking the orders against some SLA metric. The objective is to become your partners' preferred supplier and increase your total amount of business done with the partner. These are senior-executive-level concerns.

Not every process needs to be re-engineered — just particular ones.

Part of the goal is creating the metric by which success is to be measured. More than quantifying goals, everyone must understand how the execution of the operational process affects the metric, and how the metric relates to some other measure of corporate performance (usually a financial metric).
Process Modeling Identifies SOA Re-engineering Priorities and Aspects for Business Control

- Business view of the process (all stakeholders and functions), highlighting:
  - Decision points/conditional flow rules
  - Work task/information dependencies
  - Activity flow
- Tasks and information dependencies are mapped to:
  - Application logic (rules)
  - Data sources
  - User access points
- Flows get mapped to:
  - Integration points
  - Event infrastructure
  - Business rules and policies

The importance of process modeling cannot be understated. Good process models communicate how work is accomplished, reflecting the concerns of all the stakeholders and participating functions. Process models are needed to help business and IT managers understand actual processes and enable them, by visualization and simulation, to propose improvements. Explicit process models are easily changed because nontechnical managers understand them easily, and they are independent of the underlying resources. Models provide a basis for cross-organizational collaboration between managers responsible for the separate tasks within a process, as well as with IT professionals on the implementation of the resulting design. The key elements to be identified in a process model are the business events that trigger actions, the sequence of steps, and the business rules used in and between those steps to support decision making and execution flow. Once these actions are done, IT professionals (architects and systems analysts) can begin to map the work tasks and information dependencies to existing logic, data and user interfaces. This kind of multilevel modeling effort identifies valuable existing IT assets to be leveraged in new process designs, and highlights those areas where business users want more control over process change. As in manufacturing (where a broken finished product can easily be fixed when its design is based on a component-assembly approach), so too can re-engineering to SOA turn existing IT assets into reusable services to achieve the desired flexibility.

Action Item: Modeling must become a business discipline — not a creative pastime.
One of the biggest drivers affecting AD decisions is business process management. Business process efficiency is forcing organizations to evaluate not only their human-to-human workflow effectiveness, but also the relationship of yesterday's business applications and their role in tomorrow's business needs. For many companies, yesterday's applications were built during a time of independent business processes, and consequently, stovepipes were built with no real connection. Over time, piecemeal integration has led to even more complex IT infrastructure. Organizational demands for business process flexibility and adaptability are often difficult in an environment of hardened applications. Trying to extract business functions from larger applications is like trying to extract bricks from a large wall. It's possible, but not simple. However, the business functions, as implemented, are often valuable building blocks for tomorrow's AD. Using packaged software solutions, new custom development and restructured legacy systems can be a less risky way to migrate forward. Although this noninvasive approach to legacy reuse is a tactical step, more-strategic thinking and effort are necessary over time.

*Action Item*: Start with a tactical approach, wrapping and reusing established systems in which high value can be gained. Develop a long-term evolution plan of new development or legacy restructuring.
Strategic Imperative: A complete and balanced reuse program is needed to maximize success in an SOA environment.

**Reuse: The Key to Unlocking Service Potential**

### Reuse Program Elements

<table>
<thead>
<tr>
<th>Inventory</th>
<th>Quality Assurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata Management</td>
<td>Methodology</td>
</tr>
<tr>
<td>Standards and Principles</td>
<td>Measurement</td>
</tr>
<tr>
<td>Catalog/Publish</td>
<td>Performance Incentives</td>
</tr>
</tbody>
</table>

Services, in general, and Web services, in particular, are all about reuse. In fact, without a formal reuse culture and set of processes, the potential benefits of Web services drop from substantial to trivial. A formal reuse program contains:

- Formal processes and methods that explicitly require reuse as the primary option of software deployment. Although Web services are about the acquisition of software services over the Web, their benefits will be greatly enhanced by processes that formalize the reuse of all development artifacts.

- New or formalized roles that facilitate reuse. For example, an administrator who oversees the reuse process (a "cybrarian," perhaps) will be a new function. The role of a programmer will move from builder to assembler. HR staff must be employed during the process of transition to service-oriented development of applications (SODA) to aid in the definition of new roles and in the use of performance incentives to sustain the changes.

- Repositories and libraries must be provided to facilitate the reuse effort. Although they will be officially maintained by the cybrarian, they must be usable by all. These repositories not only should manage the reusable artifacts, but also must be used to create metrics that demonstrate the depth of penetration for an artifact's reuse.
The '3 Rs' of SODA

Service-oriented development of applications is based on the concept of reuse, which reduces programming effort, cycle time and inserted defects.

Bottom Line:

Requirements Model Code Test
Services/reuse
Less code, less time Responsive Fewer defects to discover Reliable
Lower costs = higher ROI. Services are good business!

With all the media hype about Web services, it's useful to step back and examine who is contributing to the hype, and why they may be right (or wrong). Adoption of a given technology, or set of technologies, typically lags substantially behind the maturity of technology. This is predominantly because technology exists not for itself, but for its use. We believe there are three key reasons for this, which we call the "3 Rs": responsiveness, reliability and return on investment (ROI). Responsiveness deals with the ability of a technology organization to react to business needs and demands.

• The AD organization can be more responsive — because it has not only the ability, but also the imperative, to reuse established services, so it can deploy software quicker.

• The software service provider (SSP) will be more reliable in two ways. First, because it will be reusing a certain percentage of established software, less variability will exist in terms of creating new code. Second, if the reused software is governed by an SLA, the SSP likely will deliver higher-quality code.

• The SSP will provide better return to the business, because the cost of acquiring functionality will almost certainly be less than building it.
### Application Security

<table>
<thead>
<tr>
<th>Drivers</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Relentless attacks</td>
<td>• Application security</td>
</tr>
<tr>
<td>• Network security is not sufficient</td>
<td>• Applied along the entire SLC</td>
</tr>
<tr>
<td>• QA is not a security testing</td>
<td>• Built-in applications</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Impediments</th>
<th>Evolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Developers’ refusal</td>
<td>Security evolution toward</td>
</tr>
<tr>
<td>• Immature technology, methodology</td>
<td>Security 3.0: Security built in</td>
</tr>
<tr>
<td>• Misconceptions about inward-facing application, outsourcing</td>
<td>Integration into SLC platforms</td>
</tr>
<tr>
<td></td>
<td>Security testing as a service</td>
</tr>
</tbody>
</table>

**Security is not just a network security. AD security is not assured just through application testing. Testing should not be just testing for compliance with requirements.**

Key findings include: 1) Enterprises often mistakenly believe that network security measures (for example, firewalls, intrusion detection systems) are sufficient to make their applications secure; 2) enterprises often mistakenly believe that their non-Internet-based applications (for example, intranet-based applications) are immune to attacks; 3) early adopters of application security are organizations that heavily conduct e-business, and also those that are strongly pressed by government and industry laws and regulations; 4) application testing technologies perfectly fit the evolving Security 3.0 concept of proactive security efforts; 5) technologies are already "good enough" for adoption; and 6) application testing markets will undergo substantial mergers and acquisitions (M&As) within the next 24 months.

Recommendations include the following: 1) Enterprises should expedite adoption of application security discipline; 2) application developers should accept that they — not network security specialists — are responsible for this discipline adoption; and 3) enterprises should assume ultimate responsibility for application security, even if AD and maintenance are outsourced.

**Action Item:** The need for application security technologies is strategic, and enterprises should be immediately evaluating and acquiring respective technologies. Yet, vendor selection is tactical because of continuous market evolution.
Strategic Imperative: Consider outsourcing technical skills. Business skills should be cultivated in-house.

It is a challenge to determine which skills are best delivered by external service providers (ESPs), and which skills by internal resources. We suggest using two criteria — business and technical expertise — to respond to that challenge.

Jobs that require little business domain expertise and little technical expertise (for example, graphics artists for Web site design) should be candidates for full-scale outsourcing.

Jobs that require a great deal of business domain expertise but little technical expertise (for example, business analysts) should be kept in-house: After all, an enterprise itself is the best expert in its own business.

Jobs that require little business domain expertise but substantial technical expertise (such as programmers) should be outsourced by augmentation: mixed teams of in-house and external resources, so that enough technical expertise stays in-house, even when an ESP finished its job and left.

Finally, jobs that require substantial business and technical knowledge (CTO, senior architects) should be kept in-house: Those few roles could make a difference between success and failure.
Conclusions

- The shift of culture and process from coding-centric to assembly-centric is imperative for success.
- Reusing legacy applications in service-oriented architecture provides building blocks during the organizational evolution.
- BPM represents a significant opportunity to be agile, but AD groups must become more flexible to adapt.
- The future of AD is attitude. Blend the old with the new. Reuse is not free, but it is required. Evolve the culture and the technology.
- Application security is the responsibility of application developers, not network security specialists.