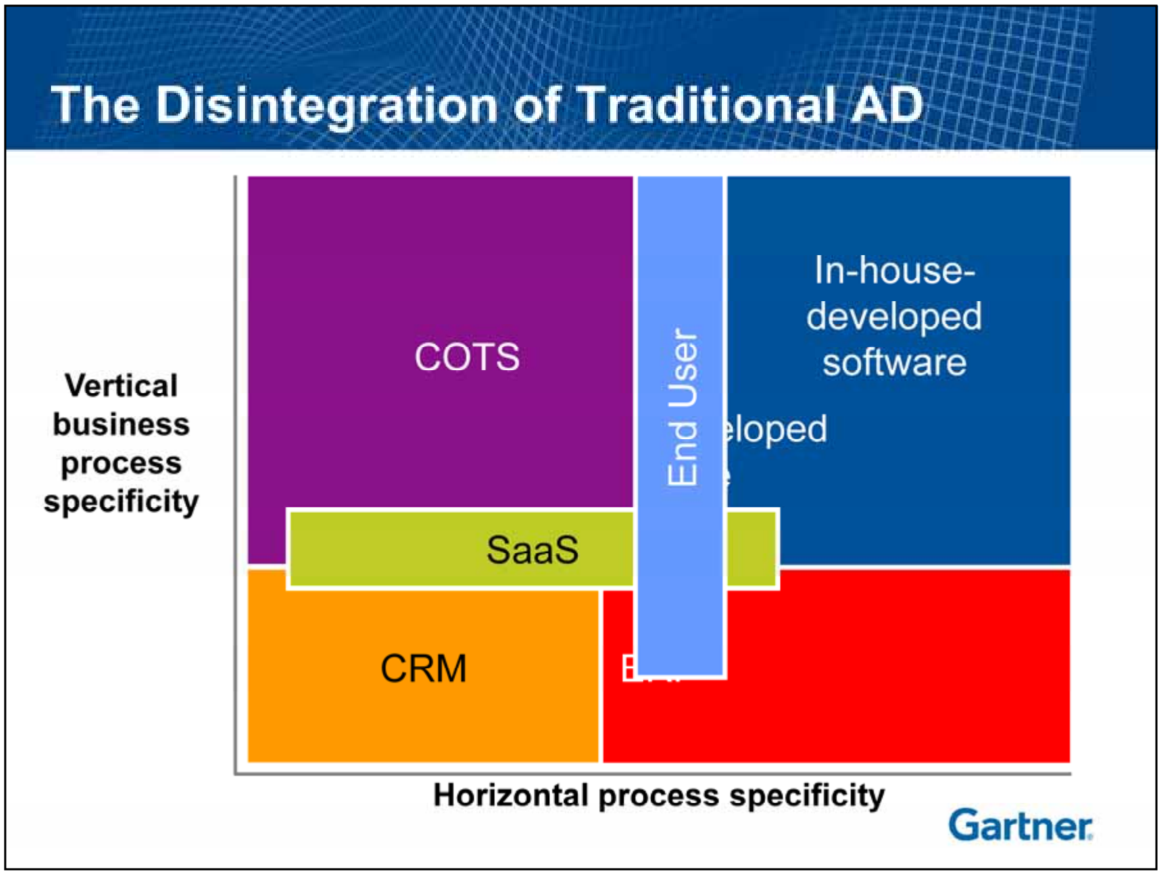
The Disintegration of AD: Putting it Back Together Again

Enterprise Integration Summit

April 13-14, 2010
WTC Hotel
Sao Paulo, Brazil

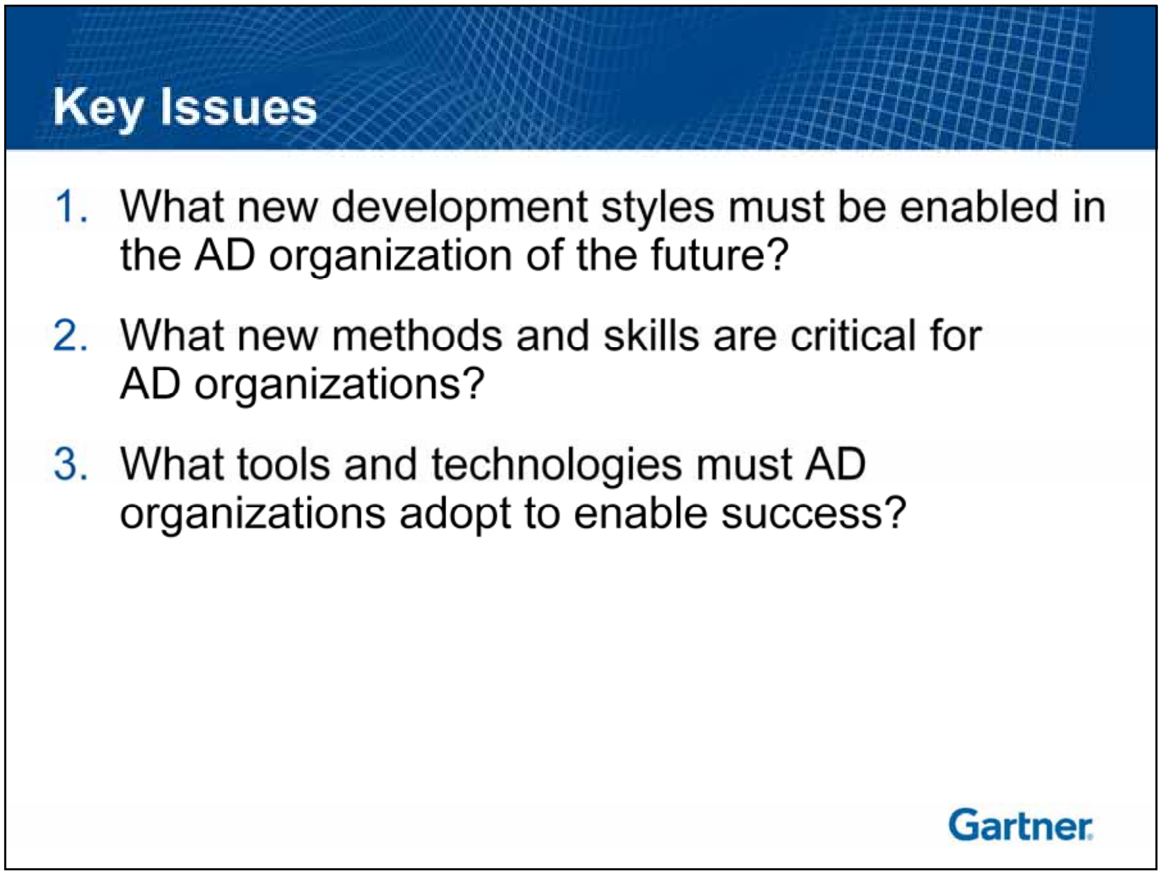
Matt Hotle


The Disintegration of Traditional AD
Vertical business process specificity
COTS
End User
In-house-developed software
SaaS
CRM
ERP
Horizontal process specificity
Gartner

In the "old days" of application development (AD), software was almost totally developed by a dedicated pool of internal developers. As time passed, though, software vendors sprung up to begin to handle both horizontal processes (i.e., processes that are common to multiple businesses regardless of their vertical) and vertical processes. Early on, these were stand-alone software that was relatively self-contained, or required a few interfaces.

That's not the case any longer. The options for software have expanded, and the amount of internal development has shrunk, to the point at which many have predicted the demise of the internal AD function. That hasn't been the case, however. Certain software has always been more appropriate for internal development, and buying has never been the 'best' choice in any condition.

Today, AD organizations are faced with picking up the fragmented pieces of their past. They must have a set of management and governance practices that allow them to form a whole out of the pieces and make that whole work together.

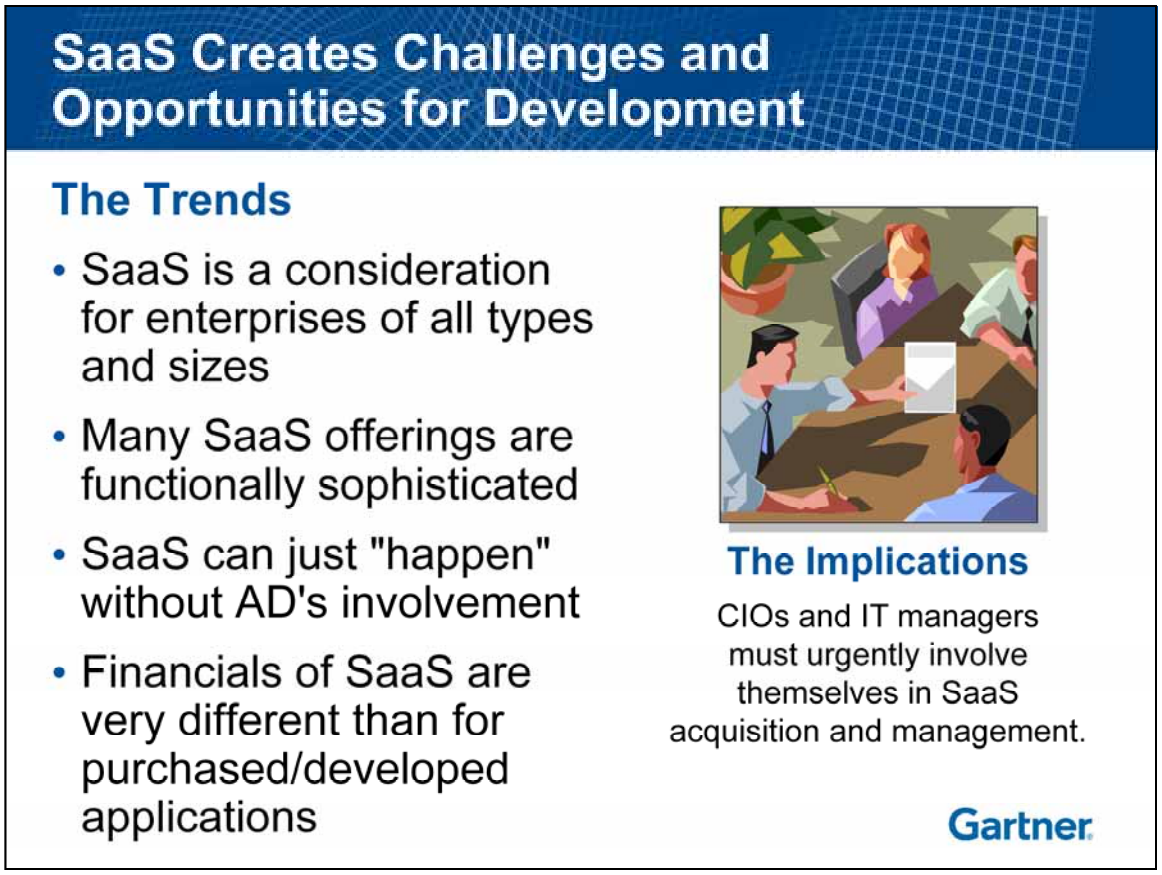


Key Issues

1. What new development styles must be enabled in the AD organization of the future?
2. What new methods and skills are critical for AD organizations?
3. What tools and technologies must AD organizations adopt to enable success?

Gartner

Tactical Guideline: CIOs and IT managers must urgently involve themselves in SaaS acquisition and management.



SaaS Creates Challenges and Opportunities for Development

The Trends

- SaaS is a consideration for enterprises of all types and sizes
- Many SaaS offerings are functionally sophisticated
- SaaS can just "happen" without AD's involvement
- Financials of SaaS are very different than for purchased/developed applications

The Implications

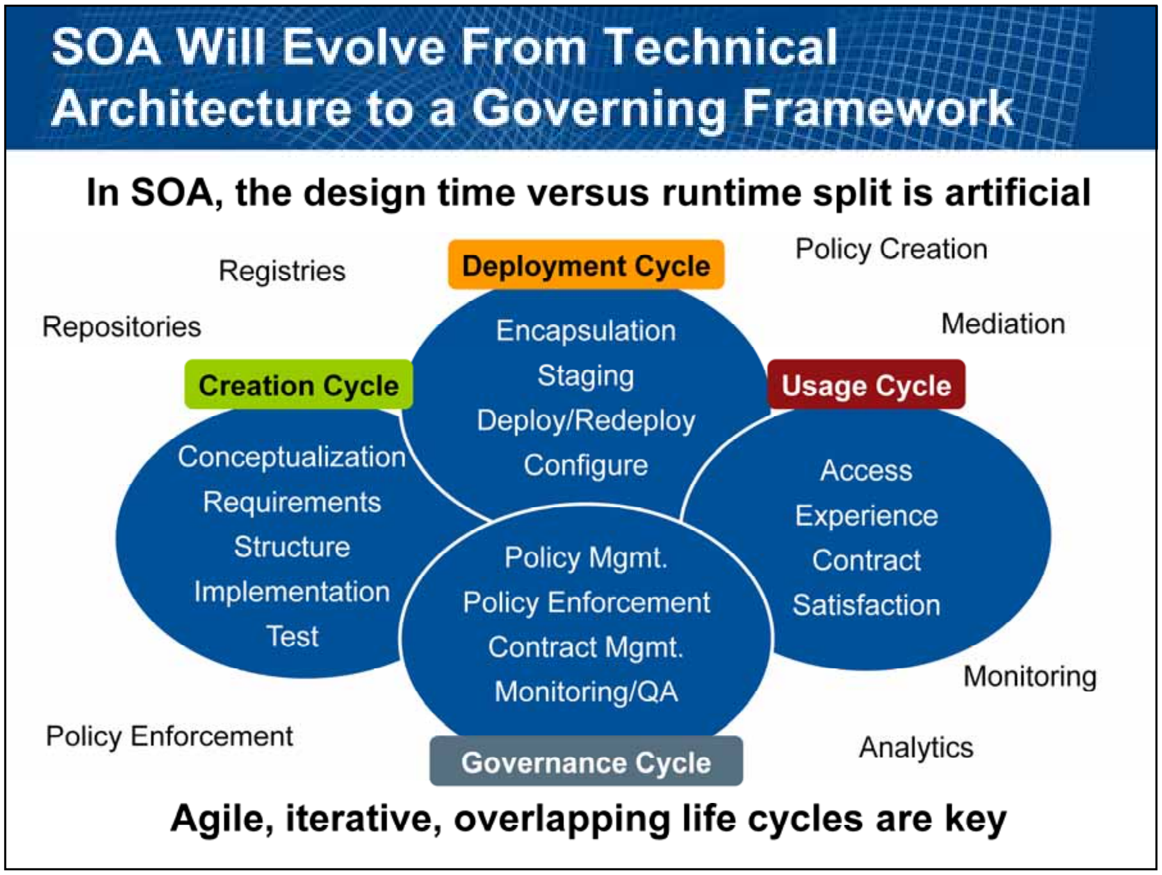
CIOs and IT managers must urgently involve themselves in SaaS acquisition and management.

Gartner

Key Issue: What will be the business and technology drivers for adopting a SaaS model?

The dysfunction of the client/server era is the oxygen for "alternative" approaches to IT development, delivery and management, of which SaaS is the most apparent version. There is now a wide consensus among IT industry leaders that SaaS is an important and meaningful issue that can no longer be regarded as the "lunatic fringe." Internal IT managers are slowly coming around to a similar view, recognizing that being seen as resistant to change is rarely a career-enhancing strategy. IT managers are looking at ways of communicating their "value add" in this new world, and are focusing on areas such as integration, security and vendor/contract management. This is a sensible and appropriate approach in Gartner's view. SaaS adoption is broadening from areas such as CRM and HR where it has been established for some time in new areas, such as procurement and compliance management. The scale of change involved in moving to a SaaS approach is proving hard to manage for many vendors with a heritage in the client/server era. Potential users of SaaS applications should make efforts to understand the dynamics within vendors' dealings with this issue; these dynamics will affect service delivery quality issues.

Action Items: CIOs and IT managers must proactively engage the business on their SaaS usage. SaaS may provide a quick fix for a user's business problem, but, in the long term, will have far-reaching IT implications if left unmanaged.



SOA Will Evolve From Technical Architecture to a Governing Framework

In SOA, the design time versus runtime split is artificial

Agile, iterative, overlapping life cycles are key

Key Issue: How will SOA Governance affect your SOA Portfolio over the next 5 years?

A maturity model for the SOA portfolio is useful to determine where products, vendors, and companies are in the evolution of their initiatives. We have defined five basic levels of maturity in the model.

Level 0: This is the level of infrastructure implementation and deployment. Most companies have entered this level, and most vendors offer something of this type.

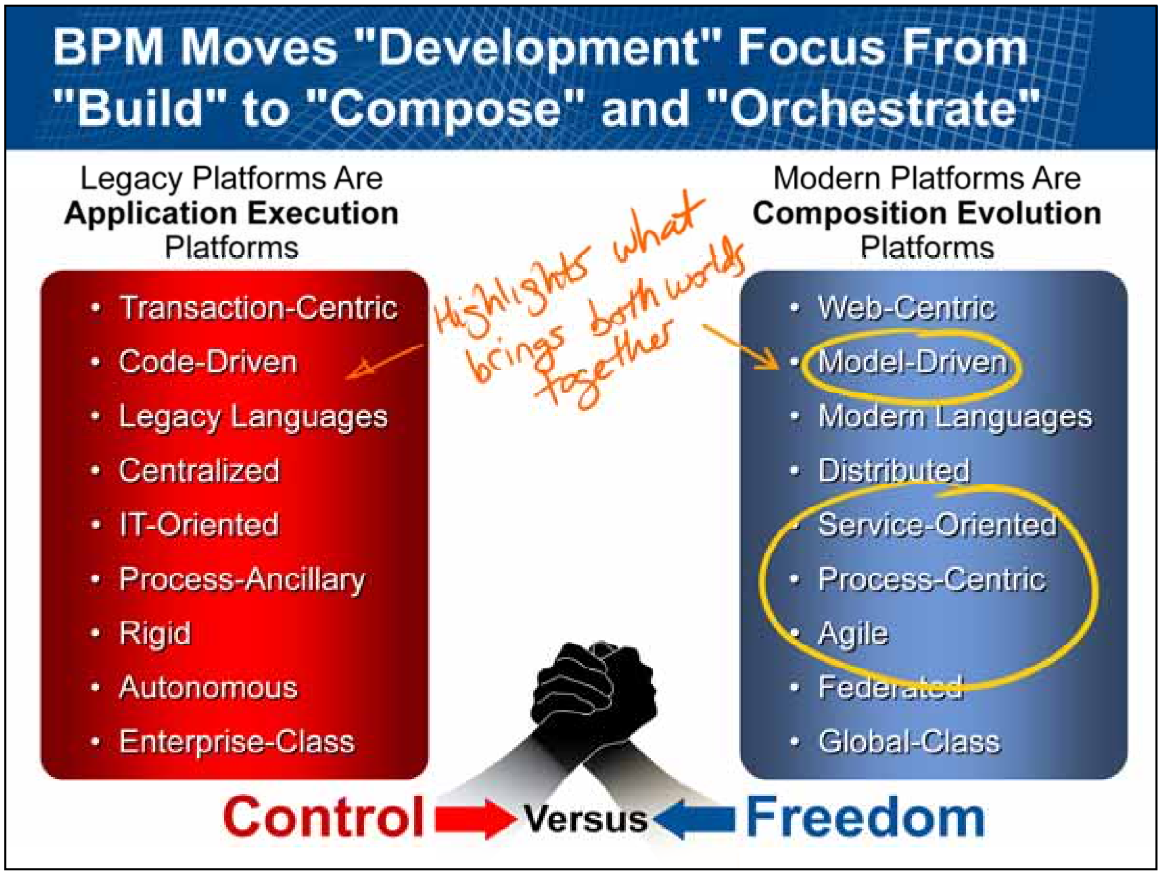
Level 1: This is the level at which governance begins to be planned and executed. Several vendors have begun to establish products for support.

Level 2: Procedural enablement is he process of collecting methodologies, designing blueprints, and producing documents that describe best practices and portfolio requirements.

Level 3: Procedural consistency is about tracking whether or not the procedures are adhered to consistently well in the organization. Very little investment from vendors or organizations has been applied.

Level 4: The level has had very little attention given to it by IT vendors. It is left up to each company to establish its own people initiatives. Those initiatives are characterized by establishment of proper roles like process-centric developers, "cybrarians," and integration competency centers.

Conclusion: SOA and BPM can move the focus of applications from execution back to construction.



BPM Moves "Development" Focus From "Build" to "Compose" and "Orchestrate"

Legacy Platforms Are **Application Execution** Platforms	Modern Platforms Are **Composition Evolution** Platforms
Transaction-Centric	Web-Centric
Code-Driven	Model-Driven
Legacy Languages	Modern Languages
Centralized	Distributed
IT-Oriented	Service-Oriented
Process-Ancillary	Process-Centric
Rigid	Agile
Autonomous	Federated
Enterprise-Class	Global-Class

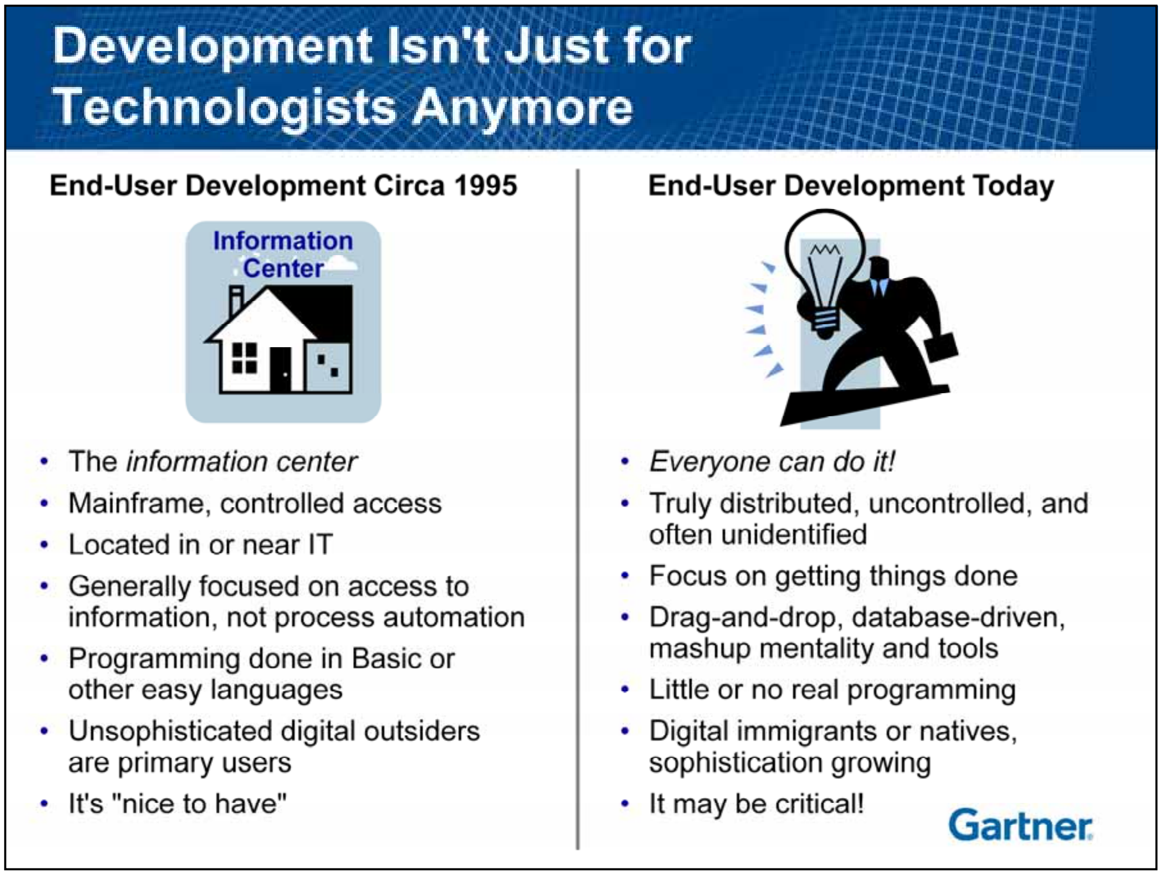
Control → Versus ← Freedom

Highlights what brings both worlds together

Key Issue: What is the "act of composition" and how will it evolve during the next five years?

The compositions of tomorrow will rely more on being model-driven than code-driven. This shift is only one of many changes related to the breakup of the application and the program into several features or characteristics that can be controlled outside the container running them. Transaction and data-centric models are yielding to Web-centric and document-centric models. One benefit is that the systems become more agile and flexible, rather than being rigid and unable to change. This is good due to the increasing pace of change demanded for the business. However, this shift must be coordinated as a collaborative effort among the various participants in the development process. Governance, then, assumes new significance as the tracking, monitoring and enforcing of policies will be the linchpin in successful projects.

Tactical Guideline: The new wave of end-user development requires new governance techniques now!



Development Isn't Just for Technologists Anymore

End-User Development Circa 1995	End-User Development Today
• The *information center*	• *Everyone can do it!*
• Mainframe, controlled access	• Truly distributed, uncontrolled, and often unidentified
• Located in or near IT	• Focus on getting things done
• Generally focused on access to information, not process automation	• Drag-and-drop, database-driven, mashup mentality and tools
• Programming done in Basic or other easy languages	• Little or no real programming
• Unsophisticated digital outsiders are primary users	• Digital immigrants or natives, sophistication growing
• It's "nice to have"	• It may be critical!

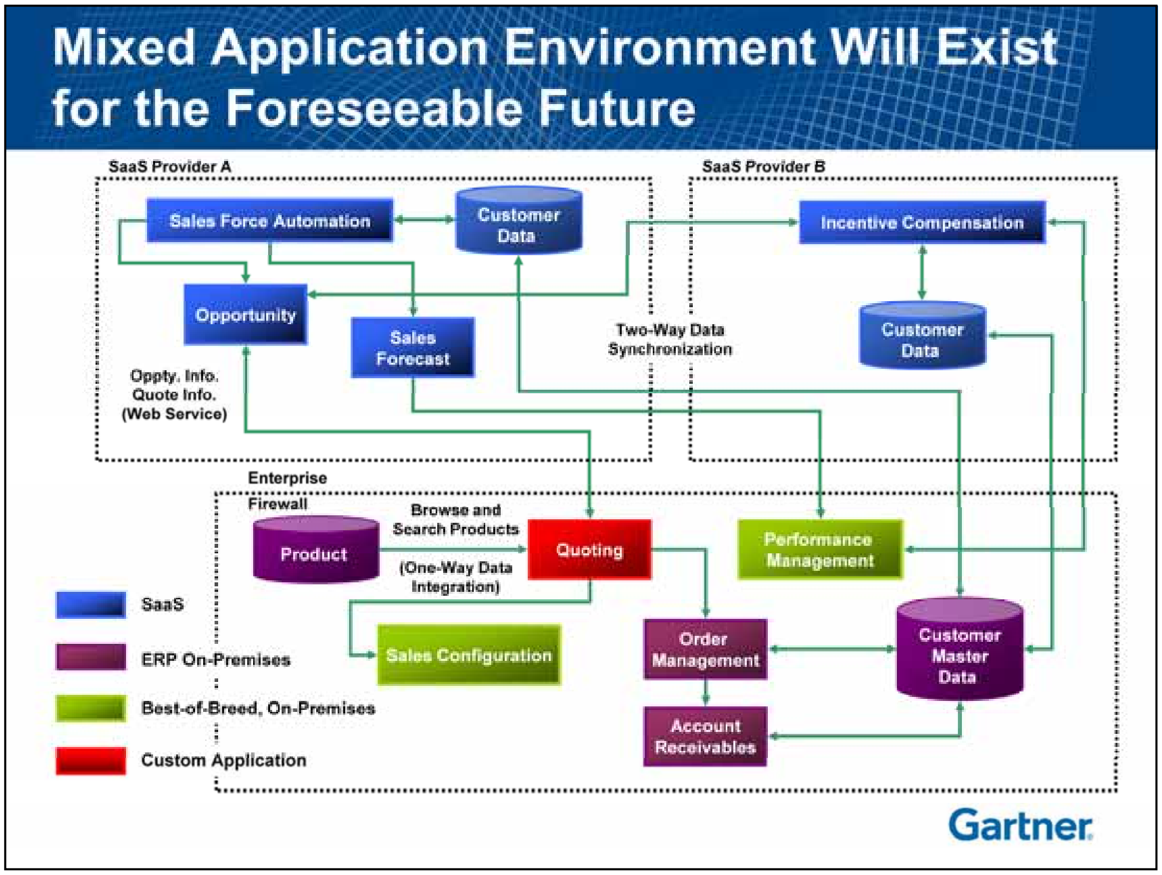
Gartner

After a hiatus in the late 1990s and early years of this century, "real" development by end users again has become an issue for organizations. However, the type of development being done by the business today is quite different from that done in the "old days."

In the 1980s and 1990s, many organizations formed information centers whose charge was to make end-user computing available to the business. The development of mainframe spreadsheets (Dynaplan, for example) and simple languages, such as APL or Basic, combined with the new desktop computer, made access to technology seem easy, even though users remained unsophisticated and were technology outsiders. Because IT hosted the information center and aided users in coding, the IT department knew what was being done and by whom.

In today's truly distributed world, not only can virtually anyone develop software, nearly everyone can. Business users are relatively sophisticated and are at least true immigrants to the digital world, if not natives. Many business users have their own space on a social-networking site or have their own Web sites. They use technology not just for data access, as in the past, but to do "real work."

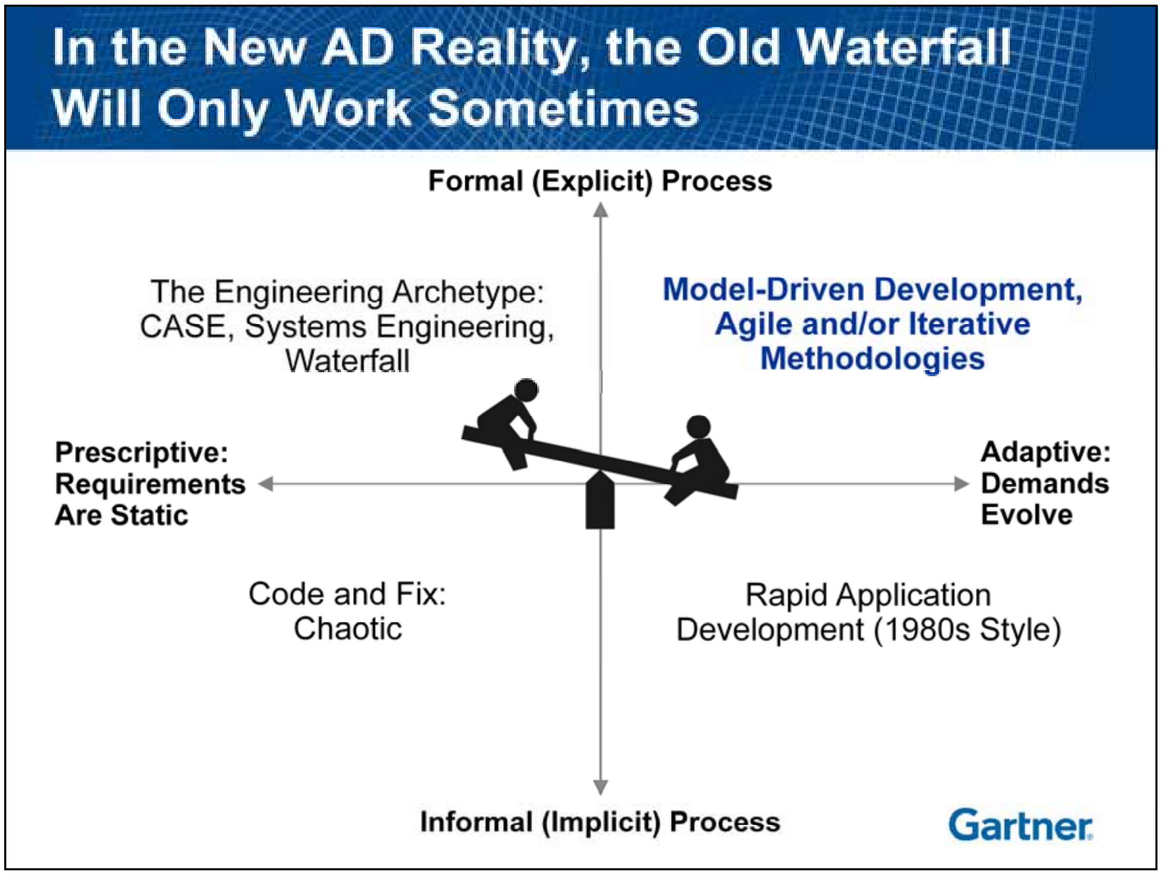
Strategic Planning Assumption: By 2010, 75% of large enterprise SaaS deployments will have at least five integration or interoperable points to on-premises applications.



Key Issue: How will companies weave SaaS into their software delivery portfolio?

Any large enterprise deploying a SaaS solution will do so in the context of already-deployed on-premises applications. No large organization will completely remove its on-premises applications and replace them with a complete SaaS delivery model during the next five years. Therefore, the importance of having a strategy of weaving SaaS into a much broader application deployment strategy is critical. Although many SaaS solutions begin as isolated islands of functionality, requirements ultimately expand because of emerging business processes and/or business strategies (for example, acquiring a new company) that require the SaaS solution to be more integrated into the fabric of established on-premises applications.

Action Items: Do not get trapped in providing a "quick fix" solution to your most urgent requirements. Take a step back to assess the complete business process users need to accomplish their jobs to determine potential on-premises integration or interoperability requirements.



In the New AD Reality, the Old Waterfall Will Only Work Sometimes

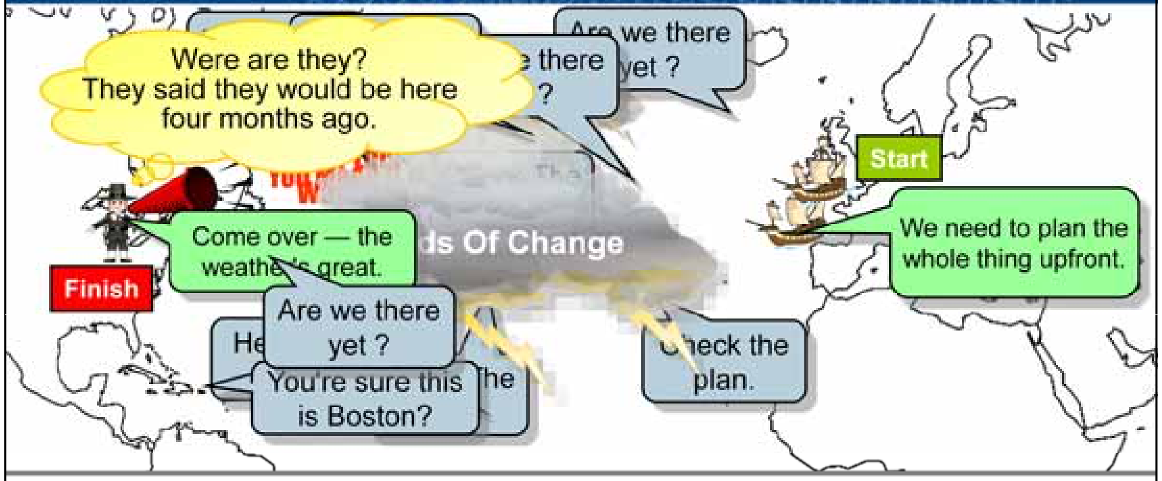
Formal (Explicit) Process
The Engineering Archetype: CASE, Systems Engineering, Waterfall
Model-Driven Development, Agile and/or Iterative Methodologies
Prescriptive: Requirements Are Static
Adaptive: Demands Evolve
Code and Fix: Chaotic
Rapid Application Development (1980s Style)
Informal (Implicit) Process
Gartner

Software methodologies have and continue to seesaw from formal prescriptive methods typified by waterfall to more informal "code as a craft" methods such as extreme programming (XP). The rise of the waterfall method in all its forms was a natural reaction to the "code and fix" mentality of the 70s and 80s. As system complexity increased, so did the need to control the development process. But often this was at the cost of flexibility and time to market. Rapid application development (RAD) was the first attempt to develop in a more flexible manner but, without explicit control, it soon reverted to code and fix "bad RAD."

The early 1990s saw methods emerge that combined explicit practices with the best of RAD under the umbrella name "Agile" — not a single method, but a set of methods that have similar values and principles. Agile will not be the last time the seesaw tips, but it does represent a major milestone in software development methodologies. Agile methods offer those organizations looking for greater flexibility and shorter cycle times a viable alternative to waterfall.

Tactical Guideline: Adopt a continuous delivery strategy based on delivering and maintaining software products instead of projects.

Waterfall Gives the Illusion of Seeing the Future: Accept the Winds of Change


Were are they?
They said they would be here four months ago.
Are we there yet ?
there ?
Start
We need to plan the whole thing upfront.
ds Of Change
Finish
Come over — the weather's great.
Are we there yet ?
You're sure this is Boston?
Check the plan.

Limiting predictions to short iterations gives us increased predictability and greater agility.

"Give up the illusion of control to find it."

Iterative methods were known as RAD. For most IT organizations, however, RAD equated to delivering bad software faster. Today, more organizations are using an iterative approach to development, which is characterized by the following.

• Iterative efforts are focused on a product first, not a project.

• Each iteration is a sequenced project with time certain (that is, a time box).

• Estimation is based on how many requirements the team can deliver in the time box.

• Each iteration has a critical path that can be tracked via earned-value analysis (EVA), typically taking two to six months.

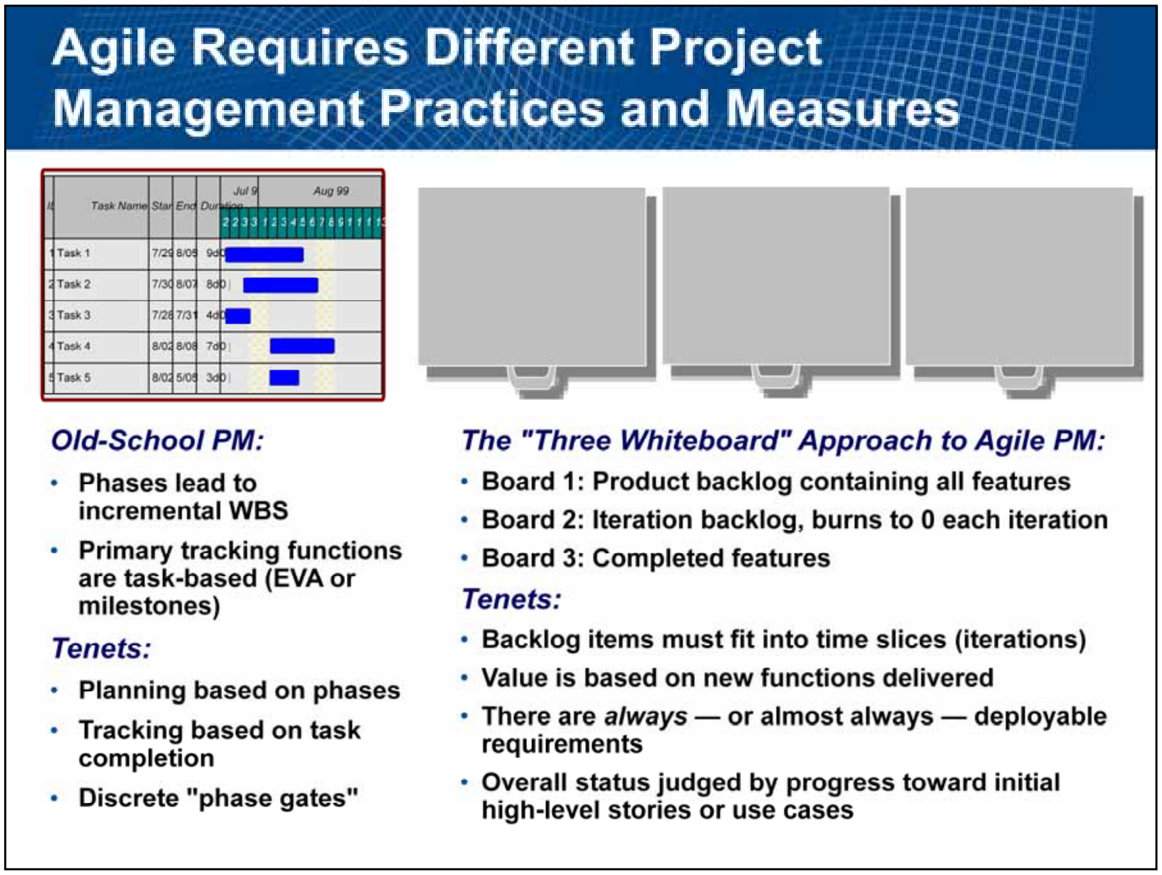
• The product team stays together throughout the life of the product.

• Success and failure of the product are based on business measurements.

• Performance of the team is based on efficiency measurements.

In a waterfall approach, the basic estimation question is: Given this set of requirements, what is the cost, time and effort? In an iterative approach, the question is: Given this time box, how much can we do?

Strategic Imperative: Since agile requirements are at the task level, more-traditional measures of status don't apply to agile.



Agile Requires Different Project Management Practices and Measures

Old-School PM:

- Phases lead to incremental WBS
- Primary tracking functions are task-based (EVA or milestones)

Tenets:

- Planning based on phases
- Tracking based on task completion
- Discrete "phase gates"

The "Three Whiteboard" Approach to Agile PM:

- Board 1: Product backlog containing all features
- Board 2: Iteration backlog, burns to 0 each iteration
- Board 3: Completed features

Tenets:

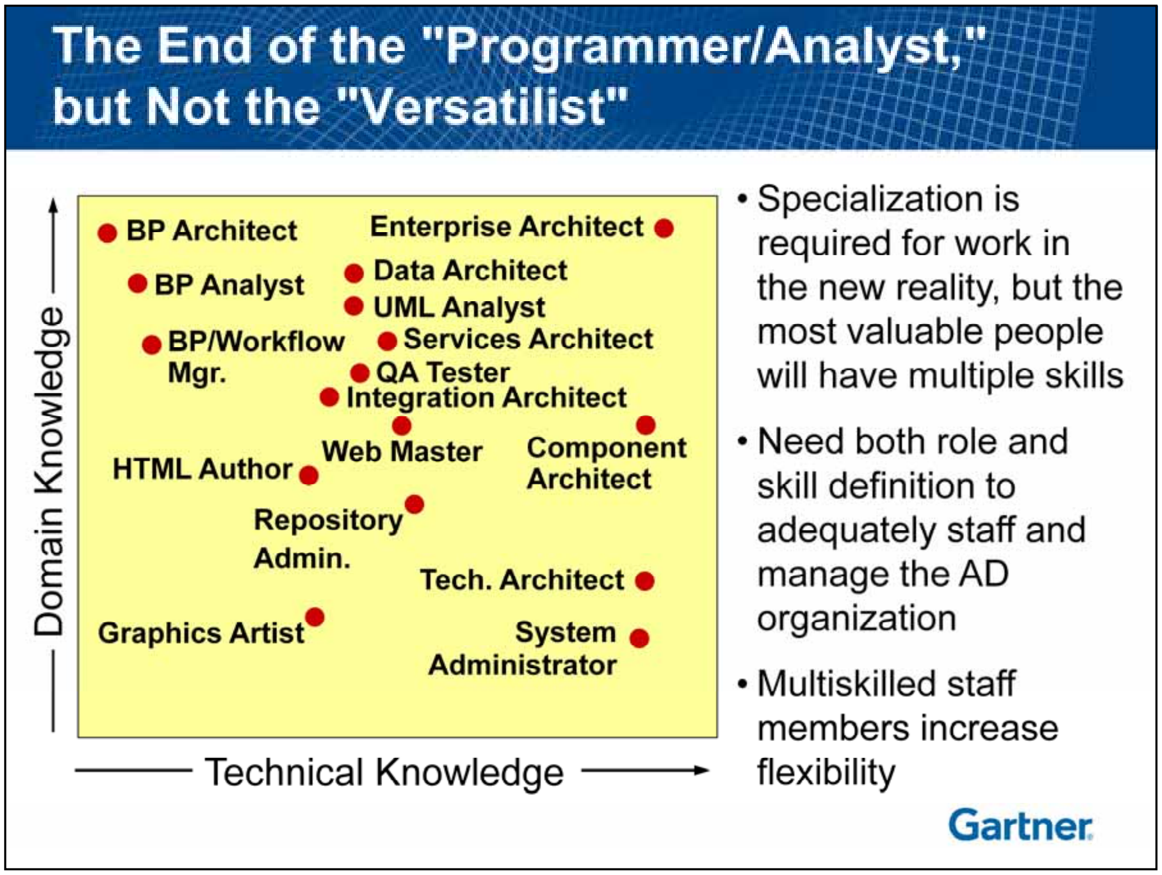
- Backlog items must fit into time slices (iterations)
- Value is based on new functions delivered
- There are *always* — or almost always — deployable requirements
- Overall status judged by progress toward initial high-level stories or use cases

In an agile method, stage gates happen literally every day with the current build. Code and functions are delivered to the base system, tests are run, and businessperson feedback is given. Agile development becomes a microcosm of the build-and-deploy phases. In addition, each time a build is complete, new requirements may be added. So, during each integration new requirements are fleshed out and formally tracked and there is, then, a structure. It's simply so quick that it's a blur.

Unless there's a specific reason, however, phase gates shouldn't be applied to this blur. There should indeed be regular checkpoints for use in funding, but the fact is that, if phase gates are applied to an agile method to make it look like a waterfall method, it is a waterfall method. Agile processes are by nature strongly iterative and rapid. Attempting to apply phase gates makes agile methods into waterfall methods. The one exception to this rule is the final acceptance test, which should be performed once the product team has finished its systems work.

Basic PM methods attempt to control a project based on complicated, dependent tasks. It's crucial to note that a two-week sprint (to use "scrum" terminology) falls within the PMI's guidelines for task planning. Interdependencies are noted on the to-do board, handling the need to integrate these into a longer-term plan.

Strategic Imperative: To be successful in the deployment of Web services, IT organizations must make a paradigm shift involving new IT and business roles, and organizational structures.

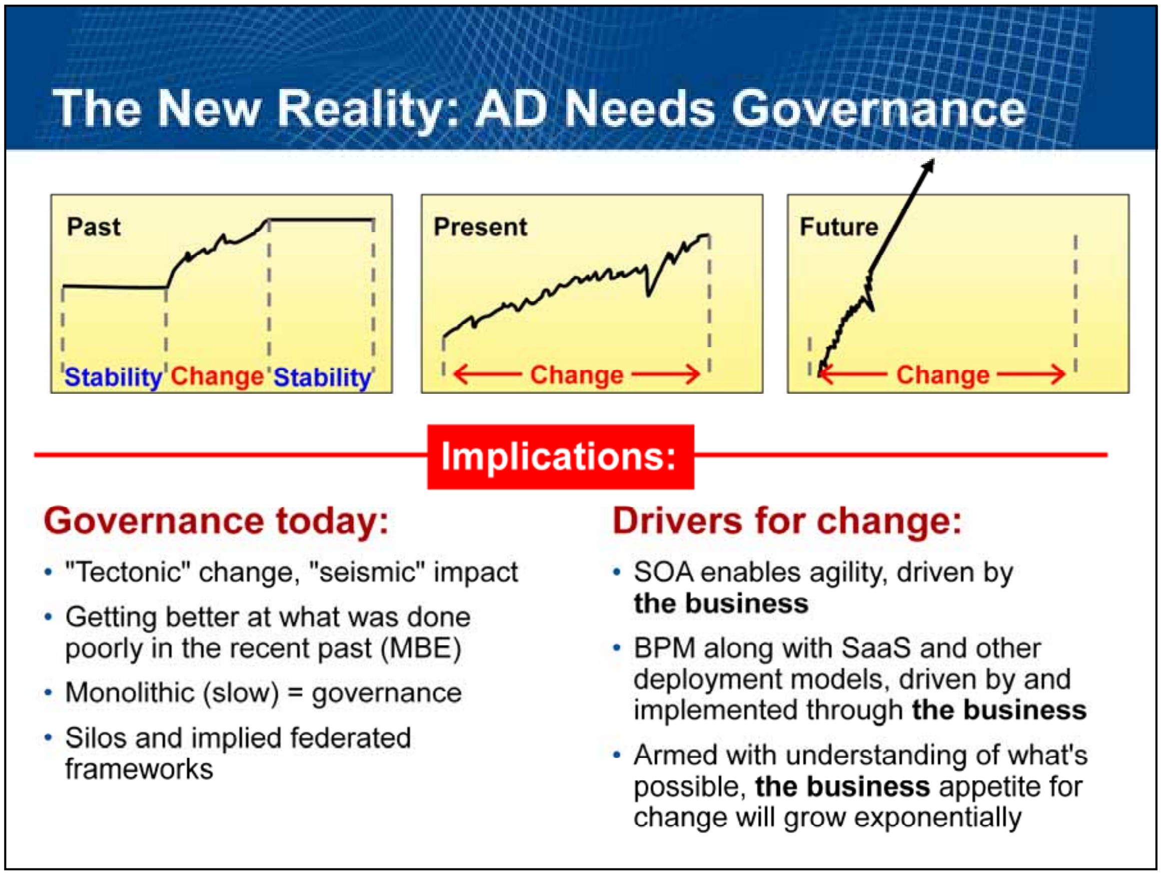


The End of the "Programmer/Analyst," but Not the "Versatilist"

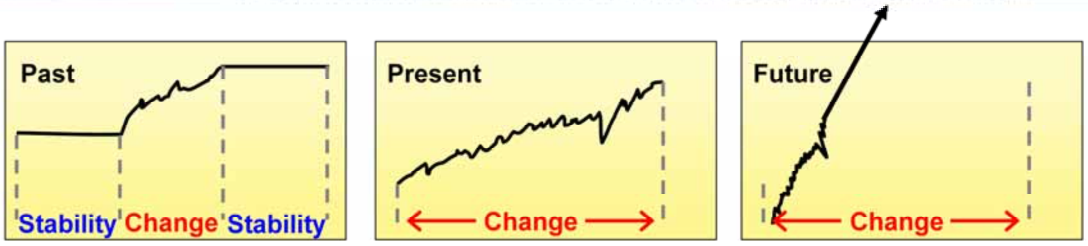
- Specialization is required for work in the new reality, but the most valuable people will have multiple skills
- Need both role and skill definition to adequately staff and manage the AD organization
- Multiskilled staff members increase flexibility

The new generation of business solutions based on SOAs requires a re-examination of roles and responsibilities. There are new roles (such as webmaster, service architect and business process manager). The role of the enterprise architect is becoming increasingly more important with SOAs since the building and deploying of new service-oriented business applications (SOBAs) requires a clear understanding of the needs and constraints across the business, technical and information architectures. There are also changes to established roles (such as the business analyst). The scope of the business analyst has grown. They are now responsible for high-level, cross-organizational business process architectures (now the domain of the business architect), as well as designing new business-unit-specific business processes and being a detailed liaison to development teams. Similar changes are occurring in the IT organization. A transition strategy that includes collaboration with the HR organization, and, probably, external mentoring and training to maximize success in moving BPM efforts forward. Roles must change in the IT organization to focus on service and component development.

Action Item: You may choose to implement new SOA, SODA, and BPM roles and practices differently in your company, but you will find that changes from current organizational structures are necessary to maximize success.



The New Reality: AD Needs Governance



Implications:

Governance today:

- "Tectonic" change, "seismic" impact
- Getting better at what was done poorly in the recent past (MBE)
- Monolithic (slow) = governance
- Silos and implied federated frameworks

Drivers for change:

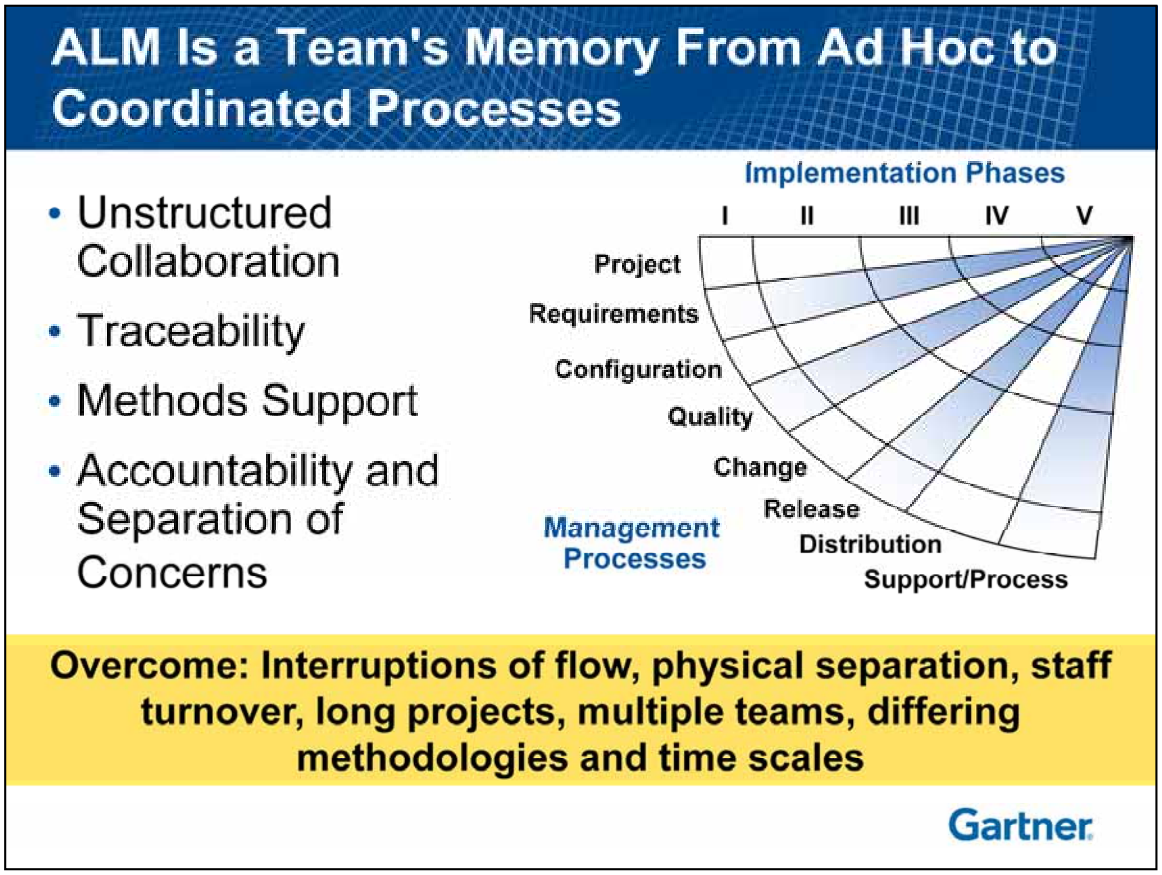
- SOA enables agility, driven by **the business**
- BPM along with SaaS and other deployment models, driven by and implemented through **the business**
- Armed with understanding of what's possible, **the business** appetite for change will grow exponentially

Application organizations can essentially be thought of as the agents that change the business. While commonly considered "tools," applications as they're implemented really embody major behavioral change within the business. Change is messy; change is dirty; change is hard.

The way we currently change the business can be thought of in the same way we think of two major tectonic plates moving against each other. Stress builds up over a period of time, then releases itself in seismic activity — an earthquake. Our applications have been delivered and/or deployed in the same manner…the business senses a need to change and therefore builds stress. Applications organizations work as hard as they can over a period of time to deliver the needed changes. And the earthquake happens.

Things are changing, though. Agile methods, such as BPM, SaaS and SOA, are all examples of ways applications organizations and the business are working together to change the disruptions caused by tectonic change. Most of these methods are not new, but most are new "to us." Thus, they will require some form of policies, procedures and rules to make them work in the short and midterm. In the longer term, organizations will find that their "application governance" and "business governance" processes simply become "process governance."

The organization's governing values are the basis for decision making. CM definitions, ownership and strategy must determined and aligned.



ALM Is a Team's Memory From Ad Hoc to Coordinated Processes

- Unstructured Collaboration
- Traceability
- Methods Support
- Accountability and Separation of Concerns

Implementation Phases
I
II
III
IV
V
Project
Requirements
Configuration
Quality
Change
Release
Distribution
Support/Process
Management Processes

Overcome: Interruptions of flow, physical separation, staff turnover, long projects, multiple teams, differing methodologies and time scales

Gartner

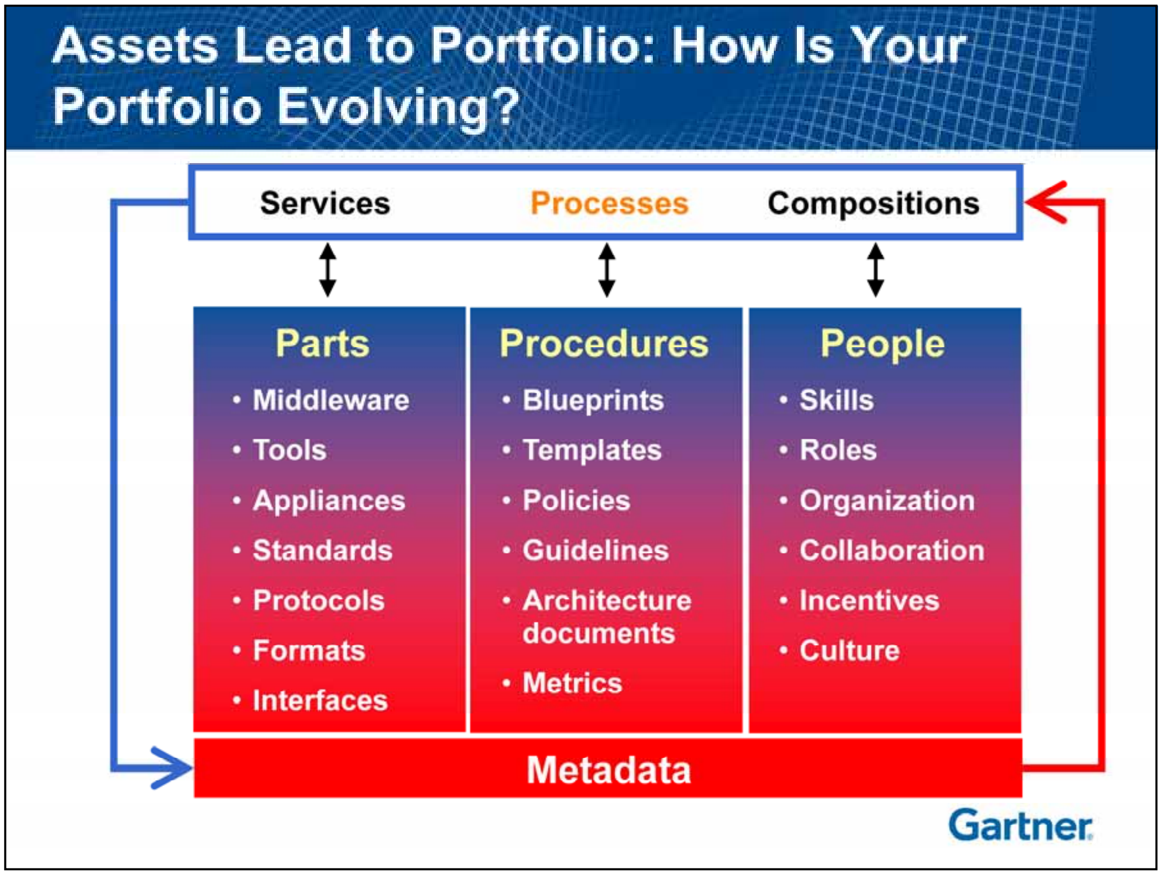
Key Issue: The evolution of AD platforms how will the market change?

Project and portfolio management play critical roles in shaping change management practices. Project plans and checkpoints create the landscape within which the development team performs, and document resource and risk expectations. The business outcomes that lead to the project charter resolve into requirements, technical design and work items. As developer work, their individual activities roll back up into the reporting of project progress.

Once shareholder needs and activities are documented, they can be mapped to a common set of shared processes. A staging plan can be developed, identifying the level of process, tooling and automation that is desired, and if necessary, the stages through which the current processes will be evolved.

If the organization has embraced the same values, chances of success are greatly increased. People are resistant to changes in measurements because they expect those changes to change the balance of decision making power and to ultimately change their reward structures. As is the case with PPM, these organizational change components must be recognized and managed. Governing values are the "truths" that the organization will use to make decisions about CM.

Strategic Planning Assumption: Through 2011, increasing complexity of SOA projects will be managed through an SOA portfolio discipline.

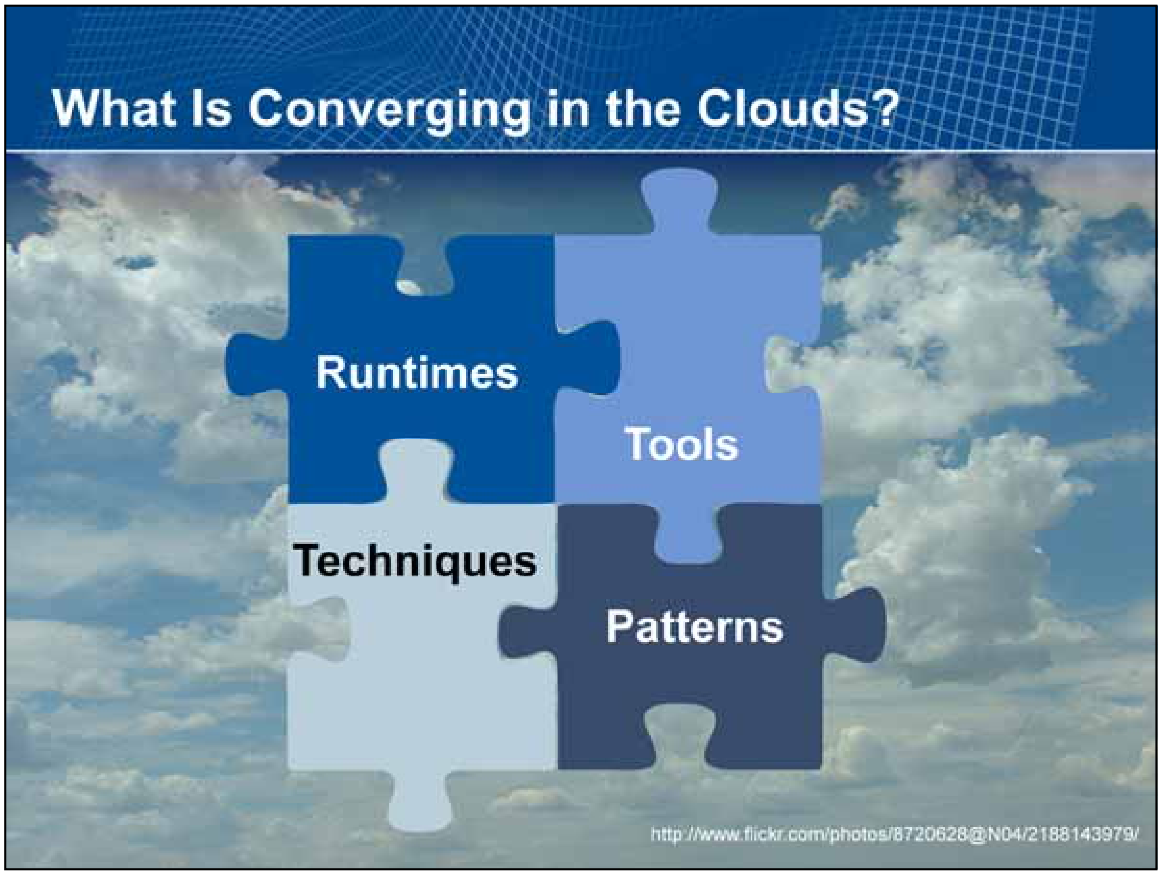


Key Issue: How will SOA Governance affect your SOA Portfolio over the next 5 years?

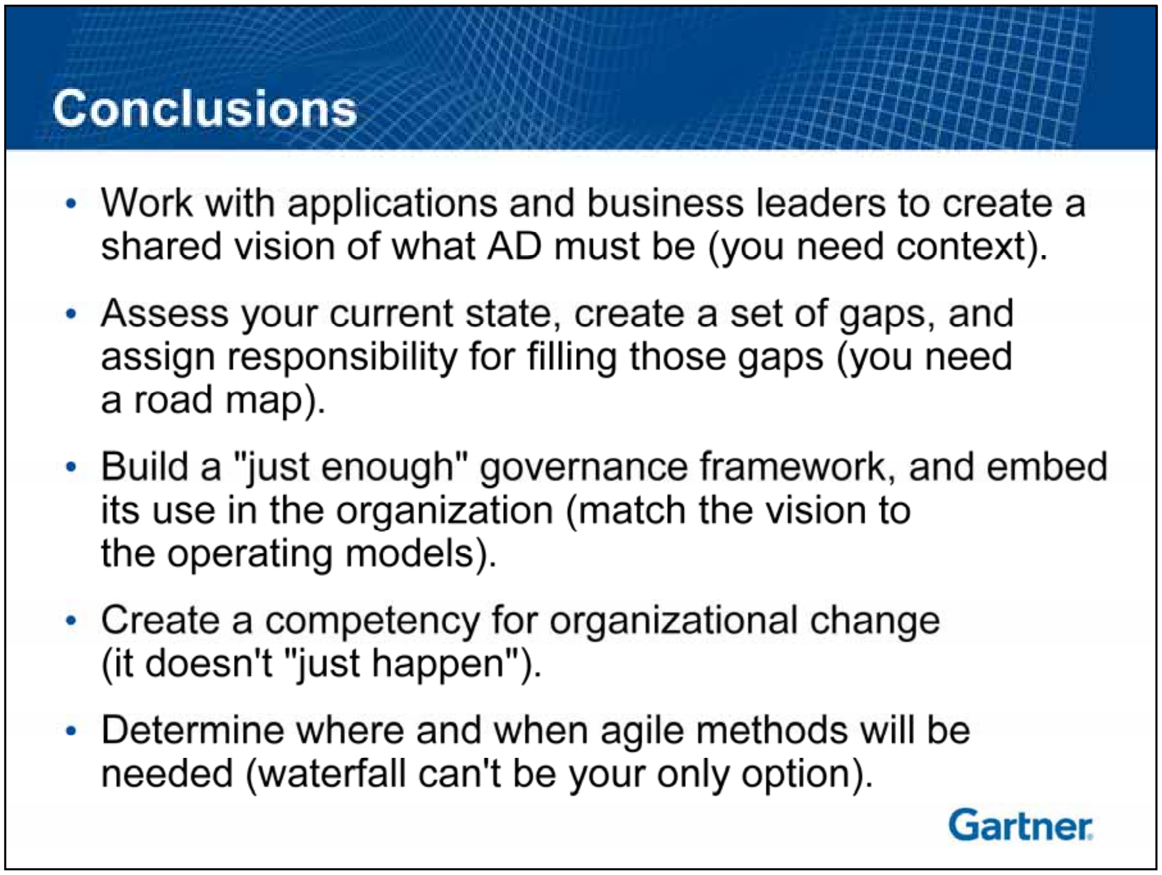
A portfolio is, simply put, a set of capabilities. The composition of your IT portfolio depends on the duties you need to accomplish daily. Anything that helps you achieve your business goals would be included. *Thus, software should be viewed as a portfolio management issue. For this discussion, based on needs expressed by Gartner clients, we use service-oriented architecture (SOA) as an example.* An SOA portfolio is broken down into three major aspects (the three Ps). **Parts:** The physical parts of the SOA, which include the infrastructure, applications, tools and services; in other words, the products, on which many companies focus to the exclusion of the other two aspects. **Procedures:** The actions taken within the SOA, including the policies and guidelines that direct those actions. This includes methodologies, blueprints, reference architectures, standards, practices, roles and responsibilities. This aspect is often neglected in favor of a product-centered view. **People:** The human resources, and the way they're governed and allocated. This includes skill sets and organization, establishment of contracts, sourcing and all the associated tasks of hiring, assignment and distribution, collaboration.

Action Item: Rather than being sidetracked too early in the selection process by a zealous, detailed discussion of a particular technology or platform, in the hope that it will offer a universal panacea, perform a self-evaluation of your SOA portfolio in light of these three parts.

Strategic Imperative: Recognize that in the future, "Web AD" and "cloud AD" will become almost interchangeable terms; future Web development will be focused on cloud computing as a deployment and runtime environment.


What Is Converging in the Clouds?
Runtimes
Tools
Techniques
Patterns
http://www.flickr.com/photos/8720628@N04/2188143979/

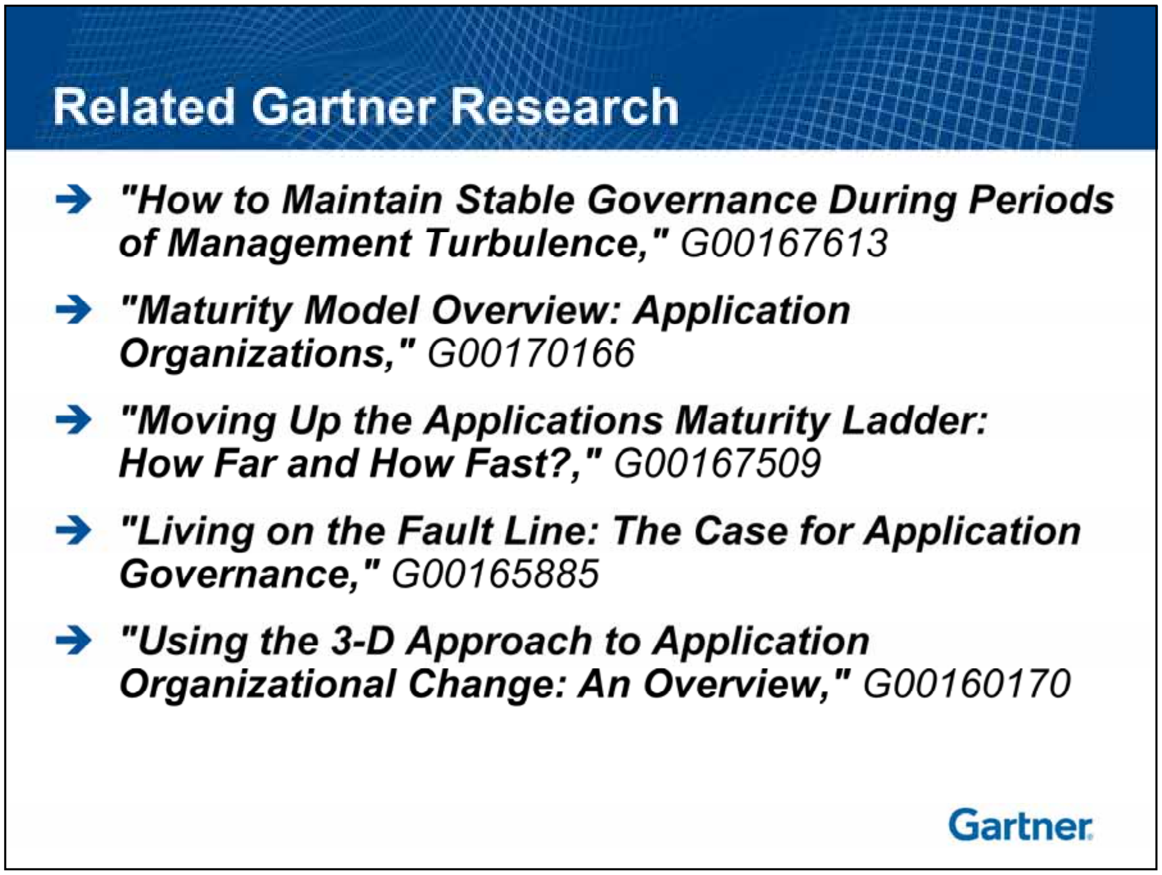
APaaS must offer a runtime to host applications, and some vendors (Google, LongJump, Stax and WaveMaker) already use Java as the runtime environment for developer applications. Microsoft Azure makes .NET runtime available to developers creating Web applications "in the cloud." Other firms (Heroku, Engine Yard, Aptana) offer Ruby on Rails runtimes optimized for cloud computing infrastructures. The dominant runtimes of precloud Web AD will play a role in APaaS. Web developers commonly use an integrated development environment (IDE) to create and modify Web applications at the program level. Developer tools used in building Web applications for APaaS are often similar (or identical) to those used for traditional Web AD. Some APaaS offer reusable components and business services that can be composed into an application, along with custom coding — giving the IDE some capabilities of an integrated composition environment (ICE). Developers "wire together" components offered by the APaaS, which may include business services and UI building blocks, depending on the platform's capabilities and component library. For example, VisualForce, part of Force.com, is a component-based framework for creating user interfaces from such reusable building blocks, and the APIs of the underlying salesforce.com CRM and Force.com platform provide reusable business services that can accelerate development of business applications. SODA and SOBA, already popular in enterprises that have evolved to a SOA-style services portfolio, are a natural fit for cloud AD. In the always-on Internet world, modular, distributed, clearly defined, swappable, shareable services are consumed by new Web applications, necessitating a loosely coupled architecture. Agile practices of Web development are a fit for cloud AD, but the highly productive nature of some APaaSs may mean rethinking Agile practices. Architectural design patterns like model-view-controller (MVC), service facades and others are transitioning from Web to cloud AD — the distributed nature of APaaS-style, service-consuming applications necessitates intermediaries, brokers and caches to minimize failures and provide abstraction that masks complexity.



Conclusions

- Work with applications and business leaders to create a shared vision of what AD must be (you need context).
- Assess your current state, create a set of gaps, and assign responsibility for filling those gaps (you need a road map).
- Build a "just enough" governance framework, and embed its use in the organization (match the vision to the operating models).
- Create a competency for organizational change (it doesn't "just happen").
- Determine where and when agile methods will be needed (waterfall can't be your only option).

Gartner



Related Gartner Research

- ***"How to Maintain Stable Governance During Periods of Management Turbulence,"*** *G00167613*
- ***"Maturity Model Overview: Application Organizations,"*** *G00170166*
- ***"Moving Up the Applications Maturity Ladder: How Far and How Fast?,"*** *G00167509*
- ***"Living on the Fault Line: The Case for Application Governance,"*** *G00165885*
- ***"Using the 3-D Approach to Application Organizational Change: An Overview,"*** *G00160170*

Gartner