

Key Issues for Application Development, 2009

Matthew Hotle

Application development organizations are under increasing stress. New technologies and deployment mechanisms, skills shortages, and nontraditional software "development" make the challenges of leading a development organization much greater than in the past.

ANALYSIS

It's rare to speak with a CIO or vice president of applications who has a simple life. Whereas, in the past, organizations would describe themselves as "package shops" or "custom-made development organizations," nearly every application organization now has a blend of "traditional" deployment styles and a bit of software as a service (SaaS), business process management (BPM) and end-user development thrown in, then mixed with multiple sourcing methods. The choices and trade-offs involved in managing these organizations drive the complexity of leading an application organization, as well as the complexity of creating research in the area traditionally known as application development.

Key Issues and the "Story Line"

The high-level key issues for 2009 in application development are traditional; however, as we go through the major story lines, it will become clear that their application will be anything but.

Key Issue: What skills will be necessary to build and deploy applications, and where will they reside?

This key issue incorporates three of our key stories. First is the issue of the skills shortage that development organizations are facing. Next is the continued use of external resources, whether onshore, "nearshore" or offshore. Finally, we find that development isn't just for developers any more. It's really "deployment," done by developers, package installers, integrators and even business people. There will be emerging needs for more-effective planning, organizing and controlling of development activities.

Key Issue: What technologies will support application deployment?

Whereas, in the past, the development tools' space was defined by which vendors filled out their product portfolios with representative tools around requirements management, design, code, and debugging and testing, the vendor landscape has changed during the past several years to move to application life cycle management. This requires a more framework-dominated approach, instead of a point-solution aggregation. Furthermore, the complexity of development activities will place a higher premium on coordination with project management and operations management. Tools also must support the current scenario of more-distributed teams that may be made up of employees, consultants and outsourced resources. A key challenge is how these teams work with nondevelopers/IT staff, and there is a split between tools for development versus tools for packages. Also, tools are evolving from being project-focused to being asset-focused or enabling reuse across projects.

Key Issue: What methods and processes are required for application development and deployment?

Methods and processes remain anathema to many, if not most, organizations. However, as service-oriented development of applications (SODA) becomes more prevalent, methods will be required, ranging from complex, model-based development to simple, iterative and/or agile methods. In fact, as depicted in the story lines that follow, agility is one of the critical must-haves for development organizations. Even where traditional development holds sway, cost-conscious organizations will raise their estimation of "just enough process" to deal with the varieties of development sources, processes and tools.

The aforementioned key issues will frame the research, but there are several story lines that Gartner will also address in 2009:

- **Development isn't just for developers anymore.** In fact, development is done by many different groups in the IT organization and the business. We call this the "disintegration" of the more-traditional development organization, which was typically organized around a group of people who supported the major packages and another group of people who did coding, around the packages or in a stand-alone fashion. That's still true, of course, but there are more options available up and down the development stack (integration competency centers, for example) and horizontally, across business and the IT organization (for example, BPM and end-user development, in which literally no IT resources are consumed in some cases). Someone has to understand how all this fits together, and we believe the traditional role of the application development organization will need to expand to include multiple deployment methods.
- **Cost cutting.** Given the current economic conditions, many, if not most, development organizations (along with their businesses) face the need to manage budgets even more tightly. In some cases, they're looking to alternative deployment methods (for example, buy versus build or SaaS versus buy). In other cases, companies are deferring a portion of new development. In still other cases, these organizations are making choices involving cutting services that the business needs. Gartner research will help frame the cost-cutting discussion in development organizations.
- **Service-oriented architecture (SOA)/SODA.** SOA is a set of mainstream technologies and architectural patterns. SODA is how development organizations "encounter" SOA, and use the architectural and technology choices to build services that support the process and workflow designs of business architects and analysts. The role of application architect to "right size" the design of services for agility and reuse is emerging. There's much discussion about what "just enough" or "good enough" SOA and SODA are, and about how to choose the necessary level of rigor.
- **Agility.** There are two types of agility discussed. First is the "capital A": agile methods, such as scrum or extreme programming. These are formal methods that stand alone. Second is the "small m, small a": more agile, where organizations are attempting to use current methods in a more agile manner or, perhaps, in iterations. Both techniques are appropriate; it's just a matter of selecting the right method at the right time.
- **Human resources.** It'd be easy to refer to this as just "skills," but there's a lot more to it than that. The first issue that organizations face is a shortage of qualified candidates to do what they're doing, and, given today's budget issues, getting those resources in the right place at the right price. Next, organizations face an aging workforce, many of whom will soon retire and take their technical and business knowledge with them. This has a major effect on which applications will be viably supported in the long term, leading to key application portfolio decisions. Finally, there is a new set of skills on the horizon that will be critical, and most of them have to do with "nontraditional" programming, such as process modeling or testing. It will be crucial to make human resource planning a key competency for future development organizations.
- **Sourcing.** Traditionally, sourcing has referred to "outsourcing" (that is, ceding control of an application or set of applications to an external third party) or to a "factory" (sending all code to a third party, while retaining the early and latter phases of a project). In addition, much of the choice to externally source was driven by labor arbitrage. Today, these choices are more complex, and sourcers are moving upward and downward in the "stack" of project phases, in an attempt to earn more work from traditional development organizations. Choosing what to externally source, when to do so, and to whom are critical choices for development organizations.

Drill-Down Key Issues

For references to individual areas of coverage in the application development space, and references to specific research, see:

"Key Issues, Application Development Tools, 2009"

"Key Issues for Application Life Cycle Management Processes and Tools"

"Key Issues for Software Quality and Testing, 2008"

"Key Issues for Web and Cloud Application Development, 2009"

"Key Issues for Application Management and Governance, 2009"

"Key Issues in Application Architecture, 2009"

REGIONAL HEADQUARTERS

Corporate Headquarters

56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters

Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters

Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters

Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters

Gartner do Brazil
Av. das Nações Unidas, 12551
9º andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509