June 25, 2010

Application Upgrades: How To Make Upgrade Decisions When Business Value Proves Elusive

by Paul D. Hamerman
for Business Process Professionals
For Business Process Professionals

June 25, 2010

Application Upgrades: How To Make Upgrade Decisions When Business Value Proves Elusive

by Paul D. Hamerman
with Connie Moore and Ralph Vitti

EXEcutIVE SUMMARY

Business owners of on-premises enterprise application packages face an ongoing dilemma: whether or not to invest in upgrades to the current release. While most business and IT execs put off upgrades as long as possible to avoid costs and minimize business disruption, some choose to move forward by staying current with the release path. When done infrequently, upgrades represent a major ownership cost spike that usually requires rigorous internal justification. Both business process stakeholders and IT applications delivery partners often struggle to make a compelling case to fund an upgrade project. To make complex upgrade decisions easier, Forrester’s Total Economic Impact™ (TEI) framework takes into account the costs and project risks, along with business benefits, risk avoidance, and flexibility. More strategically, additional options — increasingly more readily available — make long-term costs of ownership simpler, more transparent, and possibly lower. These options include software-as-a-service (SaaS), cloud-based infrastructure-as-a-service (IaaS), application managed services (AMS), and third-party support.

TABLE OF CONTENTS

2 Upgrades Remain The Wild Card In Applications Cost Of Ownership

2 Upgrade Adoption Patterns Indicate Reluctance To Move Forward

5 Customers Wrestle With Upgrade Pros And Cons, Scenarios, And Timing

8 Use TEI As A Framework To Justify Upgrade Decisions

RECOMMENDATIONS

13 Consider Four Alternatives To Level Out The Upgrade Cost Spike

WHAT IT MEANS

14 Business Owners Demand More Transparent And Predictable Costs Of Ownership

NOTES & RESOURCES

Forrester spoke with dozens of clients via inquiries and analyzed product road maps, release adoption patterns, and maintenance policies of the leading applications vendors.

Related Research Documents

“The State Of ERP 2009: Market Forces Drive Specialization, Consolidation, And Innovation”
November 2, 2009

“The ROI Of CRM Application Upgrades”
March 6, 2009

“Application Upgrades: When And Why”
December 18, 2006

© 2010, Forrester Research, Inc. All rights reserved. Unauthorized reproduction is strictly prohibited. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change. Forrester®, Technographics®, Forrester Wave, RoleView, TechRadar, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. To purchase reprints of this document, please email clientsupport@forrester.com. For additional information, go to www.forrester.com.
UPGRADES REMAIN THE WILD CARD IN APPLICATIONS COST OF OWNERSHIP

Cost of ownership is a universal concern among business process professionals responsible for managing enterprise applications. The ownership cost concerns range from the relatively fixed and predictable vendor maintenance fees, internal staffing, and infrastructure support to the less predictable cost of application upgrades. Application upgrade costs remain the wild card in the ownership cost picture because:

- **New application releases occur infrequently.** The pulse of releases for enterprise application packages beats, historically, once every two years. Some vendors have attempted to accelerate the pace of enhancements, with mixed success.\(^1\) In comparison with SaaS application packages, where updates get propagated to all customers multiple times per year, on-premises application releases move at a glacial pace.

- **Release adoption stays optional.** In the world of licensed on-premises software, the application upgrades are optional, at least until release support deadlines become a factor (see the next section for more on that). The optional nature of upgrades means that neither IT apps professionals nor business stakeholders plan and budget for them on a regular basis; therefore the upgrade costs are usually not factored in as part of the ongoing costs of ownership.

- **Upgrade costs vary widely, making it difficult to estimate.** Typically, enterprise application upgrades involve significant IT projects that often require outside assistance. Estimating the cost of an upgrade involves a number of variables, including the level of customization, whether prior releases were skipped, the extent of new functionality to be deployed, the stability of the new release, the impact on integration with other systems, etc. Elaborate planning and justification are usually required by the finance department and IT leadership to fund major upgrade projects.

UPGRADE ADOPTION PATTERNS INDICATE RELUCTANCE TO MOVE FORWARD

Upgrading enterprise applications to a new release ranked the third-highest priority among software initiatives in Forrester’s most recent survey of enterprise software decision-makers, following rationalizing and consolidating enterprise applications and updating or modernizing legacy applications.\(^2\) Application upgrades were rated as “very important” by 16% of the survey respondents and “important” by 38% in supporting current business goals. The survey result is consistent with Forrester survey results from the past several years, in which approximately 15% to 20% of enterprise companies undertake major application upgrades within a given year.

Finance and accounting applications (32%), enterprise resource planning (ERP) (27%), customer relationship management (CRM) (27%), and industry-specific applications (27%) receive the highest levels of attention in upgrade and expansion plans (see Figure 1). The attention to finance and accounting software reflects the mission-critical nature of the business processes supported
by these applications and its virtually universal use as a core system of record for businesses as well as governments. Finance and accounting applications require significant levels of regulatory compliance, which also explains the propensity to upgrade more frequently than some other types of applications. Finance and accounting form a core component of ERP systems as well, which are seeing similar levels of upgrade and expansion activity.

**Figure 1** Finance And ERP Lead Upgrade And Expansion Activity

“**What are your firm’s plans to adopt the following business applications?”**

<table>
<thead>
<tr>
<th>Application Type</th>
<th>Don’t know</th>
<th>Not interested</th>
<th>Interested, but no plans</th>
<th>Implemented, not expanding</th>
<th>Expanding/upgrading implementation</th>
<th>Planning to implement in the next 12 months</th>
<th>Planning to implement in a year or more</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance and accounting software</td>
<td>3%</td>
<td>5%</td>
<td>47%</td>
<td>3%</td>
<td>32%</td>
<td>5%</td>
<td></td>
</tr>
<tr>
<td>ERP software</td>
<td>6%</td>
<td>12%</td>
<td>10%</td>
<td>31%</td>
<td>9%</td>
<td>27%</td>
<td>5%</td>
</tr>
<tr>
<td>CRM software</td>
<td>8%</td>
<td>13%</td>
<td>29%</td>
<td>4%</td>
<td>10%</td>
<td>7%</td>
<td></td>
</tr>
<tr>
<td>Software that supports an industry-specific process</td>
<td>12%</td>
<td>15%</td>
<td>9%</td>
<td>25%</td>
<td>4%</td>
<td>4%</td>
<td></td>
</tr>
<tr>
<td>Order management software</td>
<td>7%</td>
<td>22%</td>
<td>8%</td>
<td>33%</td>
<td>18%</td>
<td>9%</td>
<td></td>
</tr>
<tr>
<td>HCM software</td>
<td>10%</td>
<td>17%</td>
<td>15%</td>
<td>29%</td>
<td>18%</td>
<td>7%</td>
<td>4%</td>
</tr>
<tr>
<td>PLM software</td>
<td>13%</td>
<td>29%</td>
<td>22%</td>
<td>15%</td>
<td>9%</td>
<td>8%</td>
<td></td>
</tr>
</tbody>
</table>

Base: 455 North American and European enterprise software decision-makers responsible for packaged applications (percentages may not total 100 because of rounding)

Source: Enterprise And SMB Software Survey, North America And Europe, Q4 2009

**Mature Application Packages Suffer From Upgrade Lag**

Upgrade adoption patterns are similar for the leading enterprise application packages, including Oracle’s PeopleSoft, Siebel, and E-Business Suite products, SAP’s ERP suite, Lawson’s S3, etc. Typically, 5% to 10% move to the latest release, 40% to 50% of customers stay on the release prior to the latest one, and 40% to 50% remain on older releases, including 10% to 20% on releases that are no longer fully supported by the vendor. For SAP specifically, the enterprise applications market leader, 46% of companies surveyed are on ECC (ERP) 6.0, which became generally available in 2006 (see Figure 2).
The release adoption patterns indicate that customers on older releases have a lower propensity to upgrade, for several reasons: They lack the resources to invest in the upgrade project; high levels of customization make an upgrade impractical; and upgrades become more difficult when releases are skipped (due to schema and architectural changes, etc.). They also recognize the diminishing value of their vendor maintenance agreement (if their release is still supported at all) and may opt out of renewing rather than paying for unused upgrades and minimal support.

### Why Your Enterprise Application Vendor Wants You To Upgrade

On-premises enterprise application vendors maintain release life cycle policies that are part of the vendor’s overall maintenance agreement. Leading enterprise application vendors (e.g., Lawson, Microsoft, Oracle, SAP) require customers to upgrade within five to seven years to avoid loss of support or increased support costs via extended maintenance programs. At the same time, vendors tout the benefits of the enhancements that are included in each release, hoping that the enhanced capabilities will resonate with customers.

From a vendor perspective, a customer that upgrades is a more profitable customer, pure and simple. Upgrading increases the likelihood of customer retention and opens additional revenue opportunities.

- **Upgrades promote future maintenance renewals.** In the current market for enterprise applications, maintenance accounts for approximately 50% of total revenues. This revenue stream is both highly profitable and growing, while license revenues have declined during the past two years. A customer that upgrades to the current release is more likely to renew its
maintenance agreement to take advantage of patches, compliance updates, technical support, and future enhancements, versus a customer on an older release that is potentially at risk to sever the maintenance contract or switch to another vendor.

- **Vendor lock-in is increased.** The architectural evolution of enterprise applications emphasizes an increasingly robust and complex middleware stack, which extends to analytics, business process management tools, and in some cases, the database and hardware platforms. Customers should be wary that new releases of the application software may contain technology architecture elements that require additional license fees or restrict flexibility to adopt third-party or open source middleware and database platforms.

- **Vendors can upsell more products, seats, and services.** Upgrading to the latest application release enables the vendor to sell additional products that often work with the latest release but not prior releases. This may include new application modules, analytics products, or new user interface technologies. Adopting a more current release, therefore, creates a new set of application license revenue opportunities for the software vendor. Upgrades also create service revenue opportunities for the vendors and their partners.

### CUSTOMERS WRESTLE WITH UPGRADE PROS AND CONS, SCENARIOS, AND TIMING

Business process owners and applications professionals cite a variety of valid reasons to upgrade, as well as equally legitimate reasons to defer upgrades, depending on the business’ priorities and requirements (see Figure 3). These reasons, detailed below, reflect key upgrade drivers and inhibitors as communicated by our clients during numerous inquiry and advisory discussions.

Business process and applications professionals may choose to upgrade to:

- **Get relief from vendor-imposed support deadlines.** Under maintenance programs from leading vendors (e.g., Oracle and SAP), vendors may no longer support older releases or may support them at significantly higher costs (i.e., 10% to 20% more than the current support fees). Upgrading may provide some measure of cost relief and risk avoidance.

- **Use new release enhancements.** Business process improvements and elimination of homegrown extensions or workarounds may result from adoption of new releases of the application software.

- **Reduce customizations.** Upgrades provide an opportunity to reduce the level of customization in application packages to the extent that the new release addresses the gaps for which the customizations were developed.
• **Take advantage of compliance updates.** Vendors often package compliance updates separately from new versions; however, upgrades are often required to maintain regulatory compliance related to taxes, regulatory reporting, and other issues.

• **Maintain flexibility to migrate to future releases and new technologies.** Staying on the upgrade path provides the flexibility to adopt new technologies related to the application architecture, as well as to adopt future releases featuring major usability improvements. Clients ask Forrester for insights, for example, about upgrade strategies in the context of Oracle's upcoming next-generation product, Oracle Fusion Applications.

• **Mitigate technology obsolescence risks.** Obsolescence risk is inherent in older releases, such as when database and hardware platforms supporting the applications are no longer viable or IT support skills for the application become scarce. The risk of downtime and business disruption as a result of platform obsolescence and diminished internal support can drive applications customers to move to the newer releases.

---

**Figure 3 Weighing Upgrade Pros And Cons**

<table>
<thead>
<tr>
<th>Why upgrade?</th>
<th>Why not upgrade?</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Get relief from vendor-imposed support deadlines</td>
<td>• Business case won’t justify cost and resource requirements</td>
</tr>
<tr>
<td>• Use new release enhancements</td>
<td>• Lack of compelling release enhancements</td>
</tr>
<tr>
<td>• Reduce customizations</td>
<td>• Level of customization makes upgrades difficult</td>
</tr>
<tr>
<td>• Take advantage of compliance updates</td>
<td>• Avoid technology stack lock-in</td>
</tr>
<tr>
<td>• Maintain flexibility to migrate to future releases and new technologies</td>
<td>• Application to be replaced</td>
</tr>
<tr>
<td>• Mitigate technology obsolescence risks</td>
<td>• Plan to reduce ownership costs via in-house or third-party support</td>
</tr>
</tbody>
</table>

Source: Forrester Research, Inc.
Conversely, business process professionals also have sound reasons for upgrade deferral or avoidance that relate to upgrade costs, IT priorities, and application strategies. Companies often choose to defer upgrade projects or to not upgrade at all because:

- **The business case won’t justify cost and resource requirements.** The most common reason that we hear for deferring an upgrade is the lack of a convincing business case. Sometimes, however, the perceived lack of benefits may be a result of incomplete analysis of the business process opportunities. IT applications professionals find it difficult to articulate the upgrade case based on IT benefits alone. In such cases, business process owners must step up to assist in the analysis, which may also benefit from outside assistance.

- **There is a lack of compelling release enhancements.** Business application enhancements are more compelling earlier in the life cycle of the product and less so as the package reaches an equilibrium or declining stage. Nevertheless, the value of release enhancements remains a function of whether or not the enhancements meet business requirements of specific customers.

- **The level of customization makes upgrades difficult.** Application packages, such as ERP, frequently do not fully address business requirements and are often customized. If the customizations are extensive and pervasive, the package will be difficult to upgrade, especially when the customizations are not well documented or compartmentalized from the core package.

- **Of a desire to avoid technology stack lock-in.** Fewer and fewer application packages remain agnostic with respect to underlying middleware and database technology, and options tend to diminish over time. In some cases, upgrading to a new release will impose certain architectural constraints, or “lock-in”, to the extent that it may make the upgrade unpalatable.

- **The application is scheduled for replacement or retirement.** One of the most clear-cut reasons for not upgrading is that the application is no longer viable and will be replaced or retired, usually within a few years. Legacy systems and products from vendors that have faded into insignificance are candidates for replacement, rather than upgrade.

- **They are planning to reduce ownership costs via in-house or third-party support.** As a cost reduction strategy, business process owners can choose third-party or in-house maintenance, which result in nonrenewal of the maintenance contract, thereby precluding future version upgrade opportunities.

**Four Upgrade Scenarios: Technical, Minor, Major Functional, And Strategic Upgrades**

Application upgrade project scenarios range from technical upgrades with minimal impact on the business to transformation upgrades that change business processes and responsibilities as well as improving IT efficiency. Although companies use many types of upgrade scenarios, four common conceptual scenarios tend to dominate (see Figure 4).
Figure 4 Application Upgrade Scenarios: From Minimal Disruption To Business Transformation

<table>
<thead>
<tr>
<th>Upgrade scenario</th>
<th>Objectives and business process impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1: Technical upgrade</td>
<td>Improve system performance and update technical architecture with minimal disruption to the business process environment.</td>
</tr>
<tr>
<td>Scenario 2: Minor functional upgrade</td>
<td>Improve application usability and business process support. The business users will notice the update, but business processes will not change significantly.</td>
</tr>
<tr>
<td>Scenario 3: Major business process impact</td>
<td>Increase the functional scope of the implementation, extend the system to more users, innovate business processes, reduce customizations, and eliminate shadow systems.</td>
</tr>
<tr>
<td>Scenario 4: Strategic transformation</td>
<td>Consolidate multiple application instances on the latest software release, merge in acquired business units, centralize IT application support, and implement shared services for business processes. Under this scenario, the upgrade itself is secondary to objectives for increasing the application value proposition and lowering costs of ownership.</td>
</tr>
</tbody>
</table>

Release Maturity And Business Cycles Affect Upgrade Timing

A number of factors have an impact on the upgrade decision from a timing standpoint. The readiness of the release to be adopted is an obvious consideration. A new version typically requires six to 12 months to mature sufficiently before it is ready for mainstream adoption. The vendor’s support schedule for the version being replaced is also a factor.

From a business standpoint, the timing of a business cycle, such as the fiscal year-end or the benefits open enrollment and payroll cycles, may drive the upgrade schedule. The timing may also be influenced by other applications projects and by requirements to update technical platforms supporting the applications.

USE TEI AS A FRAMEWORK TO JUSTIFY UPGRADE DECISIONS

Forrester’s TEI methodology offers a useful framework to assess upgrade decisions. Unlike typical cost/benefit analyses for upgrades, which limit the ability of the organization to measure the full economic impact of the investment, Forrester’s TEI methodology adds the dimensions of risk or uncertainty, as well as future flexibility.9

In calculating TEI for upgrades, the benefits, risk avoidance, and flexibility are offset by upgrade costs and risks (see Figure 5).
Upgrade Cost Estimation Is Relatively Complicated

Estimating the cost of upgrading a package is similar to estimating a package implementation project. It is well understood, and a body of knowledge exists from internal IT professionals and consultants who have been through similar projects. Nevertheless, upgrade costs are subject to risks and uncertainties and are often underestimated.

Application upgrade costs consist of internal and external resources, user training, process change management, hardware, and infrastructure software costs. Normally, vendors do not charge for the version upgrade as long as the customer subscribes to the vendor’s maintenance program, but some releases may require a change in the licensing model, and others may embed infrastructure software requiring additional fees. Customers should be wary of such additional charges and should work together (e.g., via user groups) to protest the imposition of additional software costs linked to an upgrade.

Business process professionals should consider a number of variables when estimating upgrade costs:

- **Customization.** The biggest variable in upgrade complexity depends on the extent to which the application has been customized to meet customer-specific business requirements. Customizations must be carefully reviewed and migrated, and in some cases, the coding must be redone in the new version. If extensive, customizations can be a significant barrier to staying current with software releases.
• **Organizational impact.** Some new releases involve complete overhauls of the user interface, even so far that they require significant retraining. This issue gets magnified when the application is deployed in multiple countries and languages. User assimilation also counts as a cost, and though it often is not budgeted for in upgrades, it should be where the organizational impact is material.

• **Functional scope.** The upgrade cost impact should be determined based on application complexity, including the number of application modules, reports, interfaces, users, and other quantitative characteristics that will support effort-level estimating.

• **Architecture.** Some application upgrades involve platform or architectural changes, either at the customer’s option or mandated by the software vendor. Factor hardware, middleware, and database upgrade costs into the estimates, including procurement and installation costs.

• **Skipped releases.** Companies that delay upgrades more than a few years find that upgrades become more difficult as a result of the need to skip releases to get to the current release. Upgrade paths may require that the interim releases be installed temporarily to get to the current release due to schema changes and other issues. In some cases, skipping a release level involves substantially re-implementing the software, and the effort level could easily be 1.5 to 2 times a normal upgrade.

• **Resource requirements.** The level of consulting support required depends on the level of resources that can be brought to the table by the internal IT and business stakeholder departments. Many companies use consultants to supplement their own staff rather than completely outsourcing the work.

• **Integration.** Carefully inventory the number of system interfaces affected by the upgrade and assess them in terms of complexity. It may be advantageous to convert batch interfaces to message-based interfaces using SOA standards, so format conversions must be assessed.

**Anticipate And Factor Risks Into The Business Case**

The risks associated with upgrades are also relatively similar to implementation projects, but they must also be carefully assessed and estimated. Consider the following risks as cost factors that negatively impact the TEI calculation:

• **Business disruption.** Some upgrades require a period of system downtime that goes beyond a weekend interval. This may have an impact on the ability of the business to function, so it’s important to consider the costs of remedial work and lost business. In addition, the business may encounter a post-implementation business disruption when assimilating a new release — user productivity could decline for a few months, or worse, revenues could be affected. A
sporting goods catalog retailer, for example, experienced significant disruption to order fulfillment (duplicate orders, incorrect orders, service call backlog) as a result of an upgrade to a major new release that was not ready for prime time.

- **Project delays and cost overruns.** Any complex project is subject to potential delays and cost overruns. Assign a higher risk factor to projects with higher levels of complexity, involving multiple applications, more interfaces, skipped releases, and more customizations to migrate.

- **Release maturity and quality.** Major new releases typically contain a significant number of bugs, which will obviously complicate the migration. The bugs will settle down once adoption has progressed beyond a handful of customers, and the vendor will address these issues by one or more “service pack” updates. Don't accept new software releases until the system reaches stability — usually six to 12 months following the general availability (GA) announcement.

- **Affect on other applications.** A challenging aspect of upgrading enterprise applications is determining the impact on other related applications that are not scheduled for upgrade. If, for example, some applications are being upgraded in a loosely coupled suite, the related applications from the same vendor may also need upgrading due to platform and integration requirements, or this integration may need to be reconfigured. This risk factor could lead to collateral damage to the environment in terms of integration and technical compatibility issues.

**Concrete Upgrade Benefits Require Careful Analysis**

The business benefits of upgrade projects often exist in obscurity, especially where the upgrade project ownership rests with IT instead of the business. Benefits analysis must involve a careful analysis of the business process implications, ease of use, and compliance risk avoidance, as the IT benefits alone rarely justify the investment.

- **Process improvements.** In the best-case scenario, the new release of the software will have some clear benefits to improving the execution of business processes. For example, the implementation of self-service functionality that extends processes to end users via the native system can have significant efficiency benefits versus processes that depend on email and spreadsheets to extend processes to internal end users. Business process and applications professionals must engage with business stakeholders to fully analyze potential upgrade benefits related to new process functionality.

- **Ease of use.** New releases of application packages, where user interfaces have been updated to leverage new UI technology (e.g., Rich Internet Applications and collaborative technologies) show significant advances in usability. The usability benefits may have some tangible value in assimilating new business users and potentially increasing the productivity of existing users. In addition, improvements to reporting and online access to information should be assessed for productivity impacts.
• **Reduction of customizations.** New functionality often replaces customizations, so customizations must be carefully analyzed to determine opportunities for reduction. Customization, which can run rampant for products that ship with development tools (e.g., PeopleSoft), may also apply to business requirements that no longer exist. Reduced customization lowers current and future upgrade costs, as well as ongoing support costs.

• **Compliance.** Compliance plays a significant role in a risk-avoidance benefit, particularly in applications that depend on tax updates (e.g., payroll) and in regulated businesses (e.g., utilities). Loss of compliance support by not upgrading can result in higher maintenance costs and increased risks of fines due to noncompliance.

• **Extension of vendor technical support.** Similar to compliance, extension of the vendor maintenance window provides assurance that the application can be repaired if it breaks. The level of risk avoidance depends on the nature of the application and the business disruption that may be caused, as well as the stability of the existing application.

• **Avoidance of increased support fees.** Applications that age into extended customer-specific support are subject to increased maintenance fees of 10% to 20%. Upgrading can reduce the fees by moving back to standard maintenance from the escalated extended maintenance fees.

**Flexibility Adds To The Upgrade Value Proposition**

Flexibility refers the ability to manage change in the future, evolving the solution as business needs change. The impact of flexibility can be relatively high, as changes to the business can be absorbed quickly and at relatively low cost versus older systems that are relatively inflexible and would require extensive customization. Flexibility comes into play in several areas:

• **Improved configurability and process change.** Newer enterprise applications provide flexibility through visual tools that allow for rapid configuration using objects, rather than custom coding. In the context of business processes, tools embedded in the application middleware allow processes to be redefined and modified more easily.

• **Information delivery.** Improved analytics and reporting tools provide significant benefits to end users, particularly for accessing relevant information more easily and with less IT support.

• **Extensibility.** SOA tools included in the application's middleware layer enable standards-based integration with other applications. This category also includes the flexibility to increase that application's functional footprint by using additional modules and partner-based offerings, as well as developing custom add-ons using extension tools.
• **Ability to migrate to future releases.** Upgrade dependencies require a customer to follow a defined upgrade path, so the upgrade adds flexibility for future upgrades. In the case of a next-generation release in the development pipeline (e.g., Oracle Fusion Applications), the upgrade provides the flexibility to migrate.

---

**RECOMMENDATIONS**

**CONSIDER FOUR ALTERNATIVES TO LEVEL OUT THE UPGRADE COST SPIKE**

At best, packaged application upgrades represent a spike in operating costs. At worst, they represent a brick wall that leads to system obsolescence. Consider the following alternative ways to level out the ownership costs and avoid the big-bang upgrade cost spike:

- **Get on the latest release, and stay current.** Under this scenario, commit to stay current on the latest releases of the software, consuming each release once it is determined to be stable. Determine the frequency of new releases, and budget the upgrades as operating expense over the course of multiple years, if necessary — thereby avoiding the debate of whether or not to upgrade.

- **Outsource the care and feeding of your enterprise packages.** Use third-party service partners (or the software vendor itself) to host and manage your enterprise applications, holding them accountable for keeping the software up to date. This scenario may not save you money (it could cost more), but it effectively spreads and stabilizes operating costs over future years. Like SaaS, this arrangement can add transparency and predictability to applications operating costs.

- **Stop upgrading, and reduce costs of ownership and support.** If the upgrades aren’t delivering value, and compliance and obsolescence risks are manageable, don’t upgrade the package. This scenario can be supplemented with third-party support for break-fix and compliance updates, as well as hosting and managed services.10

- **Move to SaaS, where possible.** Investigate SaaS applications where similar (or better) functionality can be obtained. Under the SaaS model, the costs of ownership are simplified, and software enhancements are delivered more frequently and less disruptively as part of the service. SaaS is widely adopted in CRM and human resources management (HRM) applications and is beginning to gain momentum in financial accounting and comprehensive ERP. Whereas SaaS CRM and HRM are adopted in organizations ranging from SMBs to large enterprises, we believe that SaaS ERP will see more traction in the SMB segment until extensibility and integration services become robust enough for larger enterprises to consider this deployment model.
WHAT IT MEANS

BUSINESSES WANT MORE TRANSPARENT AND PREDICTABLE COSTS OF OWNERSHIP

Traditional on-premises application vendors have been too slow to recognize a key customer requirement: to make application costs of ownership (including upgrades) simpler and more transparent. The on-premises application sales process focuses too heavily on the initial costs of acquisition and the vendor’s maintenance contract annuity, and possibly implementation costs, but it rarely accounts for full long-term costs of ownership. In fact, application buyers are often unclear on the long-term costs of ownership when they buy packaged applications, especially the future cost of upgrades. This naivety can come back to haunt business buyers, so always pay attention to long-term costs when evaluating applications.

SaaS and cloud-based IaaS will create a game-changing impact on the application buying process. For traditional on-premises application vendors to remain competitive with new-generation SaaS vendors, they must deliver a full package of services for application life-cycle costs, including software use, implementation, support, hosting/iaaS, upgrades, and managed services. Today, an ecosystem of partners provide these services, which the customer can piece together, but rarely as a comprehensive offering from the software vendor. Whereas multitenant SaaS provides a better deployment model for many customers, this approach is not universally accepted, nor is it widely available for some enterprise applications (e.g., finance and ERP). Vendors can deliver many of the benefits of SaaS with a single-tenant, hosted solution, plus potentially higher-level extensibility, if they package the life-cycle costs of ownership in a more comprehensive fashion.

ENDNOTES

1 SAP went to great lengths to accelerate the pace of upgrades via its Enhancement Package (EhP) program and tools to accelerate adoption, beginning in 2006. The pace of EhPs has slowed to one per year, with the release of EhP 5 scheduled for Q3 2010, and customers have been even slower to adopt.

2 Forrester surveyed 1,007 enterprise software decision-makers as part of the Enterprise And SMB Software Survey, North America And Europe, Q4 2009. The result of the question “How important is each of the following software initiatives in supporting your firm’s current business goals?” is shown in Figure 1 in “Align Your Enterprise Apps Investment Plans With Industry Trends.” See the April 21, 2010, “Align Your Enterprise Apps Investment Plans With Industry Trends” report.

3 Release adoption data has been provided to Forrester by the application vendors. Precision of the data varies among the products.

4 The release adoption percentage estimates are based on customer adoption profile information provided to Forrester by Lawson, Oracle, and SAP, including the data for SAP shown in Figure 2.

5 Lawson’s life-cycle support policy is that the prior release is supported for three years after the new release becomes generally available, with a minimum of four-year support on each release. Microsoft provides
five years of support plus five years of extended support on its Microsoft Dynamics applications. Oracle provides five years of standard support plus three years of extended support. SAP provides seven years of support on ERP 6.0 plus two years of extended support.

6 In 2009, ERP license revenues declined by a whopping 24%, while maintenance revenues grew slightly. Forrester projects that, through 2013, ERP maintenance revenues will grow at 5.8% rate annually, while license revenues will be flat for the period. For Forrester’s market projection for ERP see the November 2, 2009, “The State Of ERP 2009: Market Forces Drive Specialization, Consolidation, And Innovation” report.

7 Forrester compared the application strategies of the two enterprise applications market leaders, Oracle and SAP, and found contrasting styles with respect to database and middleware compatibility but similarity in strategies to move customers to adjacent projects via integration hooks and product road maps. See the October 31, 2008, “Which Has The Better Apps Strategy: Oracle Or SAP?” report.

8 Oracle recently began to introduce new modules (e.g., talent management) and deployment models (e.g., SaaS) to make new products compatible with certain prior releases, enabling customers to adopt the new functionality without upgrading the core ERP applications.

9 Forrester developed an ROI-based methodology for analyzing and evaluating the costs, benefits, and risks of IT decisions, Total Economic Impact™ (TEI). TEI allows the measurement of the effectiveness of an IT decision or project and can be used as a proactive, predictive tool. Forrester has used this methodology to evaluate the feasibility of application upgrade projects as well as to assess the relative paybacks of various upgrade-scenario options. See the August 4, 2008, “The Total Economic Impact™ Methodology: A Foundation For Sound Technology Investments” report.

10 The leading provider for third-party maintenance services is Rimini Street. For overview of this market, see the February 18, 2009, “Third-Party Apps Maintenance Rebounds” report.

For a profile of the market for applications outsourcing services, see the February 8, 2010, “Market Overview: Applications Outsourcing Suppliers Adjust To The ‘New Normal’” report.