Business agility revealed!

How IT can enable business change with application lifecycle management

To truly enable business change, IT managers must examine their approach to planning, developing, deploying, and operating software applications.

Table of contents

Introduction—partners in business change

Whatever strategy and plans a business had last year, the current economic climate brings a new reality: Businesses must act boldly to meet the demands of changed markets and economics. And while there was a time when the role of IT was to "keep the lights on," today every business change is enabled by applications. Business agility means IT agility. This puts IT on the front line of business change, elbow to elbow with line-of-business managers.

Coincidentally, the last wave of major application refresh—deployed in the specter of Y2K and during the dot-com boom—has run its course, and major applications now require modernization or replacement. The promise of modernization is a more nimble and efficient enterprise. But when every dollar counts, how does the business ensure IT investments produce the right business outcome?

To truly enable business change, IT managers must examine their approach to planning, developing, deploying, and operating software applications. That's what this white paper is about—an expanded application lifecycle that begins with business strategy and carries through business results so you can bring on the next generation of applications while better aligning IT efforts with the changing business.

The challenges of application modernization

Application modernization touches almost all aspects of IT. It turns local, dedicated teams into virtual, distributed ones. It reshapes applications from stove-pipe software to composite "systems of systems." It enriches user experience and company brand via Web 2.0 and rich Internet applications. And it changes release management from singular launches to multi-application "release trains."

Figure 1: Today's applications are designed for business change.

Yesterday's technologies, teams, applications	**Today's** technologies, teams, applications
Designed to last	Designed to change
Tightly coupled	Loosely coupled, modular
Integrated silos	Compositions (of services, of applications)
Code-oriented	Process-oriented
Rigid sequential development	Interactive and iterative development
Cost-centered	Business-oriented
Homogeneous	Heterogeneous

These trends offer lower cost of ownership, flexible, responsive architectures, the ability to leverage specialty skill sets from around the globe, quicker development cycles with lower risk, and more streamlined IT portfolios. However, the path to realizing these new benefits is not without hurdles.

New challenges to application fundamentals

Sharing and reuse mean many applications will share and reuse defects if they reach production. Risk increases so quality assurance (QA) becomes more critical than ever. QA must learn to manage requirements, testing, and release not within a single project or application but across the whole set of services and applications that share components. New questions must be answered:

- Do we have the right tooling (dashboard views, automation, etc.) to manage Agile projects?
- How do we know the impact a defect or enhancement will have on our "system of systems"?
- With so much sharing of components, how can we be sure applications will perform?
- So much of the business is now exposed through Web applications—is it secure?

Heightened dependencies among projects and teams

Since applications are no longer self-contained, projects are no longer self-contained. Functionality, performance, and security decisions made within a project affect IT services throughout the enterprise. Architectural policies must be aligned to business strategy and then communicated and enforced across IT. As a result, such decisions must be made with an enterprise view and based on business architecture. IT must ask:

- Does our architectural strategy support the business strategy?
- How do we plan and govern the way application building blocks (services) are created, used, and maintained?
- How do we ensure enterprise reuse of components without killing the nimbleness and entrepreneurship of project teams?
- Once in production, how do we manage change? How do we assess the impact of failures, notify the necessary parties, and ensure a coordinated response?

Greater need for a shared business-IT view

In a recent survey by the Economist Intelligence Unit, 41 percent of IT managers said half or fewer of IT initiatives had a positive impact on the business. Implementing a complete application lifecycle is about improving that. IT managers must work directly with line-of-business managers to determine:

- Who is asking for what IT investment? What is the status of those investments?
- Are there gaps or redundancies in the IT portfolio?
- If the business changes suddenly, where and how must the IT portfolio change?
- Do we understand the requirements the same way?
- How do we modernize without interrupting current business processes?

If application modernization holds the promise of improved business agility, it is up to IT managers to deliver on the promise. “Business-IT alignment” and “the business outcomes of IT,” topics usually discussed over oak tables, must make their way into every IT discussion and decision. Application lifecycle management (ALM), properly conceived, enables just that—the means to ensure better business outcomes.

Meeting new-era challenges

With the benefits and challenges of modernization understood, the question before IT is: How do we respond? Each challenge demands that IT managers look beyond the purely functional tasks he or she managed in the past to see them in the context of the whole enterprise. And each challenge demands a specific IT response.

Figure 2: IT response to challenges

Challenge	IT response
Create a shared business-IT view.	Implement strategic control points.
Manage heightened dependencies among projects and teams.	Understand and manage the complete application lifecycle.
Maintain application fundamentals.	Ensure functionality, performance, and security.

The foundation—ensure functionality, performance, and security.

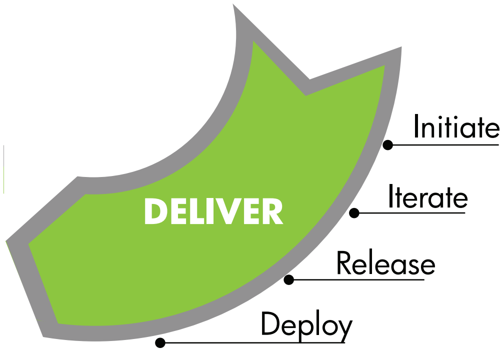
We'll start with application fundamentals. The need for application performance, security, and functionality hasn't changed. But modernization has wrought new challenges in delivering these fundamentals.

Automated testing, for example, is imperative to keep up with short-cycle, Agile methodologies. Dashboard views help managers keep distributed teams on track. Risk assessment must consider the potential impact of defects on all the business systems that might use a shared service or component. As applications implement Web 2.0 concepts, the projected user load becomes difficult to predict, so performance testing must establish the performance parameters of the application, and IT operations must monitor carefully for performance thresholds. Since more of the business is exposed through Web applications, implementing and verifying security is more critical. Development and QA must verify that services and applications conform to security policies, and security vulnerabilities must be treated as defects.

Understand and manage a complete application lifecycle.

In development and QA we have learned to manage our tasks in terms of the software development lifecycle (SDLC)—a set of processes that lead us through the tasks of application planning, design, development, and launch. (See Figure 3.)

Figure 3: The software development lifecycle



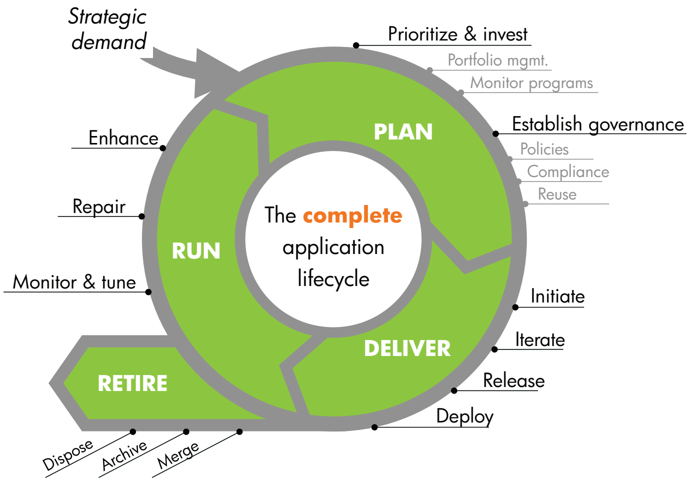
There are a few reasons why an SDLC-centric view isn't enough. This narrower view tends to ignore the strategic business drivers—the decisions to put money behind automating or improving one process versus another—that gave rise to the application in the first place. Moreover, the SDLC-centric approach forgets that an application really only begins life after going live. No matter how long the delivery phase, launch isn't an end but a beginning—of future enhancements, of adjustments based on usage, of changes in market demand, etc. Organizations that think primarily in terms of the software development lifecycle too easily fall into the temptation of "chucking it over the wall"—from the business to application development, from application development to operations.

The properly modernized enterprise knows that beyond the software development lifecycle, there is the broader view that addresses more significant issues:

- Why the business should invest in this application rather than others
- How it changes the way the business operates
- The results is it expected to achieve
- Who is using it (once deployed) and how
- If its availability and performance make it useful to business users
- If it is benefiting the business in the way that was expected
- When the application has reached the end of its useful life

Expanding our thinking to a complete application lifecycle—rather than just a software development lifecycle—gives us a framework for answering these questions. This expanded view encompasses the phases for planning, developing, running, and ultimately retiring the application. (See Figure 4.)

Figure 4: The complete application lifecycle



Business strategy drives the investment plan of a company. Application strategy—based on business strategy—provides an investment plan for applications and the people, tools, projects, and programs required to produce and operate them. Application strategy also determines the architecture, governance policies, processes, and metrics IT will use to carry out the strategy and assess its ongoing success. Application development includes the plan-design-develop-test-launch stages on which development and QA organizations are traditionally focused. The decisions we make here must implement the strategy, and they must support the needs of the operations phase. And operations is more than just keeping it running. IT operations staff must implement systems to measure service levels, to isolate problems when they occur, and to know which business services are threatened if a component fails or a software or infrastructure change goes awry. Finally, older or redundant applications must be retired on a regular basis. Otherwise the portfolio bloats and maintenance and hardware costs increase.

By thinking about the complete lifecycle of an application, not just the SDLC, an IT organization gets a true picture of the application. The final step is to understand how to align with business stakeholders throughout the lifecycle.

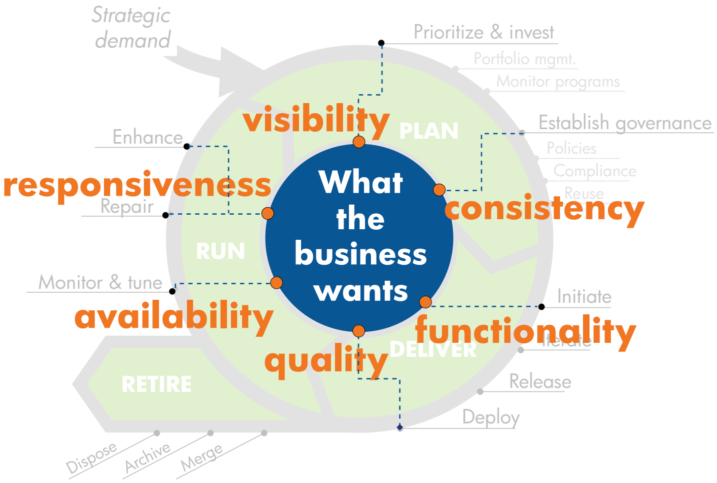
Implement strategic control points.

Developing an effective partnership with business managers requires their involvement at key points in the application lifecycle. These are strategic control points—the points where business stakeholders see and affect application decisions, including whether ongoing investment is warranted. From the perspective of IT, strategic control points are those in the application lifecycle where buy-in of the customer (i.e., the business) is imperative.

Figure 5 shows the control points and how they overlay the application lifecycle. Let’s survey them briefly by lifecycle stage.

Plan – Control points in the planning phase help manage the IT investment to properly reflect business needs and priorities. IT and business managers work together to identify business demand and to manage the IT portfolio to meet the demand. Portfolio management helps assure resources are allocated appropriately and projects are mounted in direct support of business needs. In this phase, IT managers must also establish architectural governance policies that help assure that development activities properly implement the strategy.

Figure 5: HP application lifecycle management implements strategic control points.



Deliver – The delivery phase contains traditional development and quality assurance activities. But now, they operate within the confines of strategy and governance that have been established for the business, for IT, and for individual projects. Control points address the rigorous identification and management of requirements and verify that developed software fulfills the requirements for functionality, performance, and security.

Run – Control points in the operations phase let IT assess whether applications meet availability and performance goals (service levels) in production, and they help IT manage the change process for both applications and infrastructure. When failures occur, IT operations should prioritize restorative actions based on the business impact of failures. And since many service disruptions are caused by changes, IT must understand what business services might be affected before changes are initiated.

Retire – This final phase leans on control points from the planning phase, ensuring that the portfolio is kept as trim as possible, while considering the application data. Analysis of the data will dictate what should be kept and merged with current applications, what can be destroyed, and what should be archived in order to meet long-term compliance, reporting, and e-discovery requirements.

HP solutions for application lifecycle management

The rise of Agile

More companies are turning to Agile practices to reduce development cycles and improve IT efficiency. In 2009, in almost every major survey, more than 70 percent of enterprises reported at least some projects employing Agile methods. While the migration to Agile is unlikely to be total, it is accelerating.

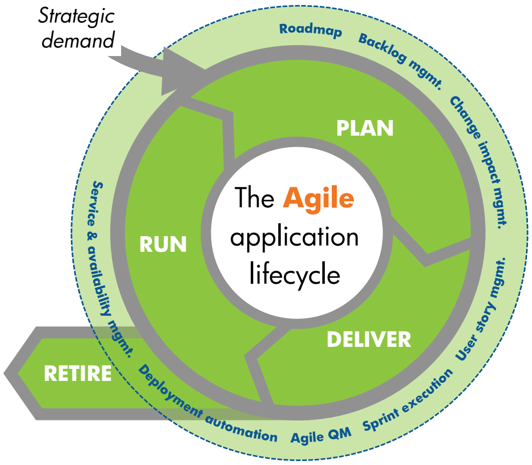
In its textbook form, Agile removes many of the theoretical aspects of application delivery, proving application behaviors sooner rather than later, providing better adaptability to change, and driving defect discovery into earlier development phases. Not surprisingly, however, it's unusual to find Agile applied in textbook form. It's far more common for an organization's developers to embrace a subset of Agile practices while business analysts, testers, and others look on with uncertainty. This presents a challenge, because the fastest development in the world is of little use if the resulting application is misaligned with business expectations, doesn't function properly, lacks interoperability, can't perform under load, or is fraught with security holes.

HP solutions provide for the broader perspective of Agile delivery, not just Agile development. The key is to make sure that project velocity doesn't compromise application quality and consistency. This requires:

- A quick, comprehensive means for **change impact analysis**. This means knowing how change to a given shared component (such as a reusable service) may affect the dependent applications. It also means comprehensive traceability, from requirement through development and test to defect management. In short, it requires a single picture of the artifacts affected by any change.
- A lightweight but thorough strategy for **requirement (i.e., user story) capture and management**.
- Sprint (or iteration) tests that validate all work to date, not just work from a given sprint. Given the brevity of sprints, this means **test automation** of both system and unit test. There should also be regular "hardening" sprints that validate **non-functional conditions including performance and security**, as well as **technical standards compliance**. Without this, issue discovery is delayed to the final weeks, when there is too little time for resolution, like in older waterfall models.
- Cross-functional team engagement and communication. This means not only **standard Agile instrumentation** (burn-up and burn-down charts, etc.) but also, critically, an effective means for storing, versioning, and **sharing artifacts**—particularly in larger enterprises where team collocation is not always viable.

Although ALM, properly conceived, should be methodology-neutral, the HP ALM strategy includes an overlay for Agile. The aim is better responsiveness across the complete application lifecycle, from request for change through release.

Figure 6: This represents the complete lifecycle from an Agile perspective.



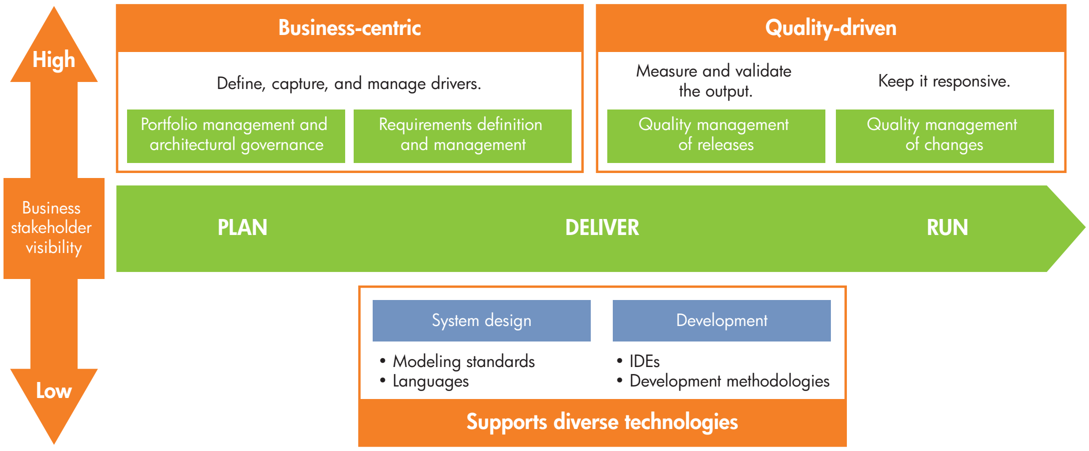
Built for IT, with the business stakeholder in mind

How is the HP solution for ALM distinct? Our industry-leading products and services are focused on better alignment between business and IT and better return on each IT dollar spent—a focus, in short, on better business outcomes. Where core development activities are concerned, HP provides broad integration to a variety of developer tools such as modeling and version control systems to enable open support of heterogeneous development environments.

HP differentiation rests on the following principles:

- **Support heterogeneous environments *and* delivery methods**. HP offers the broadest support for more than 70 application environments (.NET, Java, SAP, Oracle, etc.). Similarly, HP solutions are sufficiently flexible and feature-rich to support traditional sequential, iterative, or Agile methods or—most commonly in large enterprises—some combination of these methods.
- **Address the complete application lifecycle**. The task is to link the outcome of IT to the strategy of the business. Doing this requires IT to provide formalized processes for incorporating business strategy into IT planning and for linking every phase of the application lifecycle to achieve the strategy goals. HP solutions provide the coverage to do this.

Figure 7: The HP ALM solution focuses on what matters to the business.


High
Business stakeholder visibility
Low
Business-centric
Define, capture, and manage drivers.
Portfolio management and architectural governance
Requirements definition and management
Quality-driven
Measure and validate the output.
Keep it responsive.
Quality management of releases
Quality management of changes
PLAN
DELIVER
RUN
System design
• Modeling standards
• Languages
Development
• IDEs
• Development methodologies
Supports diverse technologies

- **Enforce strategic control points**. Strategic control points are the few, critical points where precise control has a direct effect on the outcome. HP solutions center on these control points and give IT the tools to steer the application lifecycle to meet business goals and to achieve the IT agility businesses need.
- **Focus where the business focuses**. The largest challenge for IT as we modernize applications is to achieve the alignment needed to enable better business outcomes of IT. We provide HP solutions that manage control points in the areas that are most visible to the business while maintaining compatibility with leading solutions for development methodologies and tools.
- **Accelerate the time to value**. The era of the IT mega-project is past. IT agility comes from identifying and implementing the programs and tools that return value to the business quickly. HP solutions are modular, allowing IT to solve the most pressing problems first, and integrated, increasing value when multiple solutions are implemented. Moreover, with the HP ALM solution customers have the option of choosing software as a service (SaaS), allowing bottom-line value without any additional investment in infrastructure.
- **Build on a heritage of quality**. HP has been providing software solutions to help IT organizations solve the hard problems for almost 20 years. We are the industry leader in automated software quality, application performance validation, distributed performance and availability management, and network management.

Learn more

How effectively is your IT organization aligning the investment of human and financial resources with the needs and goals of the business? How will you undertake the application modernization tasks ahead? How will you provide the IT agility needed for business agility? To learn more about HP solutions for ALM and discover how HP can help your organization link IT strategy to business results, go to **hp.com/go/alm**.

Technology for better business outcomes

To learn more, visit www.hp.com/go/alm

4AA2-5000ENW Rev.1, January 2010