COLUMN | **Lifecycle Management Matters**
What Developers, Admins, and Architects Need to Know

Application Lifecycle Management for Everyone

Lifecycle Topics for Developers, Development Topics for Administrators

Karl Kessler (karl.kessler@sap.com) joined SAP AG in 1992. He is the Product Manager of the SAP NetWeaver foundation – which includes SAP NetWeaver Application Server, Web Dynpro, ABAP Workbench, and SAP NetWeaver Developer Studio – and is responsible for all rollout activities. Currently, Karl's focus is on how SAP NetWeaver powers SAP ERP and SAP Business Suite software and on lifecycle management in particular.

With this issue, SAPinsider introduces "Lifecycle Management Matters," the next generation of Karl Kessler's "Under Development" column. This column will continue to introduce new development tools and technology, accompanied by "big picture" roadmap issues. But it will also, by design, broaden the conversation to connect developers, administrators, and system architects by covering both technology and strategy at every phase of the application life cycle.

With the growing complexity of system environments, everyone needs a clear understanding of how their contribution fits into the larger system landscape now and in the future, and how IT roles interrelate. As Karl puts it, "We're all in the same boat now."

Highlighting the dramatic changes in IT since the first issue of SAPinsider, this installment explores how those changes impact development — and anyone responsible for creating, updating, and managing applications.

For those of you who are long-time readers of this column, you've witnessed almost a decade of evolution in SAP's development tools and technology – from SAP R/3 and the Web-enablement of SAP solutions to the SAP NetWeaver technology platform, SAP ERP, and most recently, SAP's enhancement package strategy.

This evolution has meant that development itself has changed. With IT systems becoming increasingly complex and heterogeneous, and with exponential growth of integration points, developers can't afford to simply dive into tools and tasks without a sense of the larger landscape strategy. Developers need to ask the right questions, challenge conflicting requirements, and, ultimately, add value to the solutions they create or oversee. They also need to understand the impact of their work on the overall IT landscape.

As a result, this column's coverage has rather naturally expanded into what has traditionally been seen as the world of system administrators: **application lifecycle management (ALM).** Case in point: Previous columns covering SAP's enhancement package strategy have laid necessary groundwork for the switch framework.[1] Completing a task – "click here to switch this dictionary data element," for example – is one thing, but knowing the task's ramifications on your current landscape requires a broader understanding of the life cycle of your solution portfolio (see the sidebar on the next page).

From "Under Development" to "Lifecycle Management Matters"

This column's new name, Lifecycle Management Matters, reflects the changes and demands that have arisen, and is designed to bridge the gap between developers and administrators. Opening communication channels among everyone who has a role across the application life cycle – from the initial blueprint, to custom development and transport, to monitoring, performance, and security – will give IT a better opportunity to simplify processes, consolidate landscapes, reduce instances, and still maintain operations.

[1] See, for example, "Innovation Without Disruption: A Deep Dive into SAP's Enhancement Package Strategy for SAP ERP" in the January-March 2009 issue of *SAPinsider* (sapinsider.wispubs.com) and "Industry Solutions Are Now Integrated into the SAP ERP Core: How the Switch and Enhancement Framework Makes It Possible" in the October-December 2008 issue.

"Big Picture" Issues for Developers: Enhancement Packages and Namespaces

With custom-built solutions or changes, the goal is not only to meet the stated requirements – you also are increasingly responsible for minimizing your work's impact on the system and avoiding conflicts with the overall system. This means going beyond the purely technical level to understand the story behind the new solutions or changes.

With the switch framework, for example, developers have a straightforward view from the ABAP Workbench and can clearly see whether a development object is assigned to a switch. When you edit a program from the Workbench, you'll see a function group, ABAP class, dictionary data element, and so on. With one click, you can assign it to a switch. Or, you'll see a development object from SAP and notice that it is switched.

However, without grasping the underlying enhancement package strategy, you'll only see the nuts and bolts of certain infrastructure elements on a technical level. Without the proper understanding of why objects were designed and what they are used for, you'll have a hard time properly assessing their full technical advantage or heading off significant problems. Consider these questions, for example: Why is a collection of objects switched? What are the semantics behind it? Should you create your own switches?

Another example is the namespace concept, which is built into every tool of the ABAP Workbench. You could approach this in a technical way, assigning each development object with a company name, combining multiple solutions, and rolling out custom-built solutions to your subsidiaries with a very strong, consistent naming approach. But you need to understand the proper story and semantics behind it to avoid conflicts when delivering your solutions or when solutions are merged in a productive environment.

By folding ALM coverage into discussions of technical tools, you'll see the whole strategy the tools support and get a good look at the entire landscape – even if you're developing for just one system.

In this first article, I'll start with the idea of the application life cycle itself – how SAP defines it, and what it means for developers and their non-developer peers.

Developers need to understand the impact of their work on the overall IT landscape.

Application Lifecycle Management: What It Means and Where You Fit In

SAP defines application lifecycle management as the set of tools, processes, and methodologies to run SAP and non-SAP solutions in a managed landscape, following the IT Infrastructure Library (ITIL) standard. SAP chose the ITIL standard because it helps align six well-defined phases – Requirements, Design, Build and Test, Deploy, Operate, and Optimize – with particular tasks during the entire application life cycle.

The idea is simply this: ALM provides a concrete strategy for managing the life cycle of your application or solution based on a common framework, and lays out clear standards to help you manage quality during design, implementation, and operation. This may fuel the misconception that ALM is for system administrators only. Yet, for developers working with SAP systems, this means that custom-developed code, like external applications and SAP solutions, will be centrally monitored for quality management and systems integration.

The ALM strategy is also designed to make transparent to administrators and developers the phases, roles, and common touch points for different processes, as well as an overall tools roadmap at each phase. **Figure 1** illustrates the roles and tasks involved in one ALM process, Solution Implementation, and which phases this process encompasses. (For an overview of all of the major ALM processes, see the online version of this article at **sapinsider.wispubs.com**.) I'll explore this Solution Implementation example in more detail later in the article.

In a simplified world, a phase would consist of processes that are only executed in that phase. However, due to the complexity of customer solutions and landscapes, processes typically span multiple phases – and thus involve various roles. **Figure 2** shows the ALM processes, including our Solution Implementation example, and how they span various phases.

ALM Tools Everyone Should Understand

SAP first used the terms "application lifecycle management" and "ALM" at SAP TechEd 2009 in Phoenix to structure tracks around our key lifecycle management tools: SAP Solution Manager, ITIL-compliant ALM processes, and SAP NetWeaver lifecycle and infrastructure capabilities.

In the SAP context, ALM involves running a solution landscape consisting of SAP, non-SAP, and custom solutions that are centrally managed by SAP Solution Manager.

SAP solutions and SAP Solution Manager are based on the SAP NetWeaver foundation and integration platform. SAP NetWeaver offers built-in lifecycle management capabilities, such as local monitoring and software logistics capabilities, and SAP Solution

Manager acts as the central console to start and control all ALM processes (see the "ALM Tools" sidebar on the next page). This gives you a complete picture – a single source of the truth – of your solution landscape and central access to all relevant ALM data and events. Many of the tools used in the ALM processes are available from within SAP Solution Manager work centers that provide role-based access for all who are involved in executing a process end to end.

How ALM Highlights the Interdependencies of IT Roles

With an understanding of the tools involved in ALM, let's dive deeper into the Solution Implementation

✔ **NOTE!**
The phased approach to ALM is an approximation of the various tasks and steps that are performed during an application's life cycle.

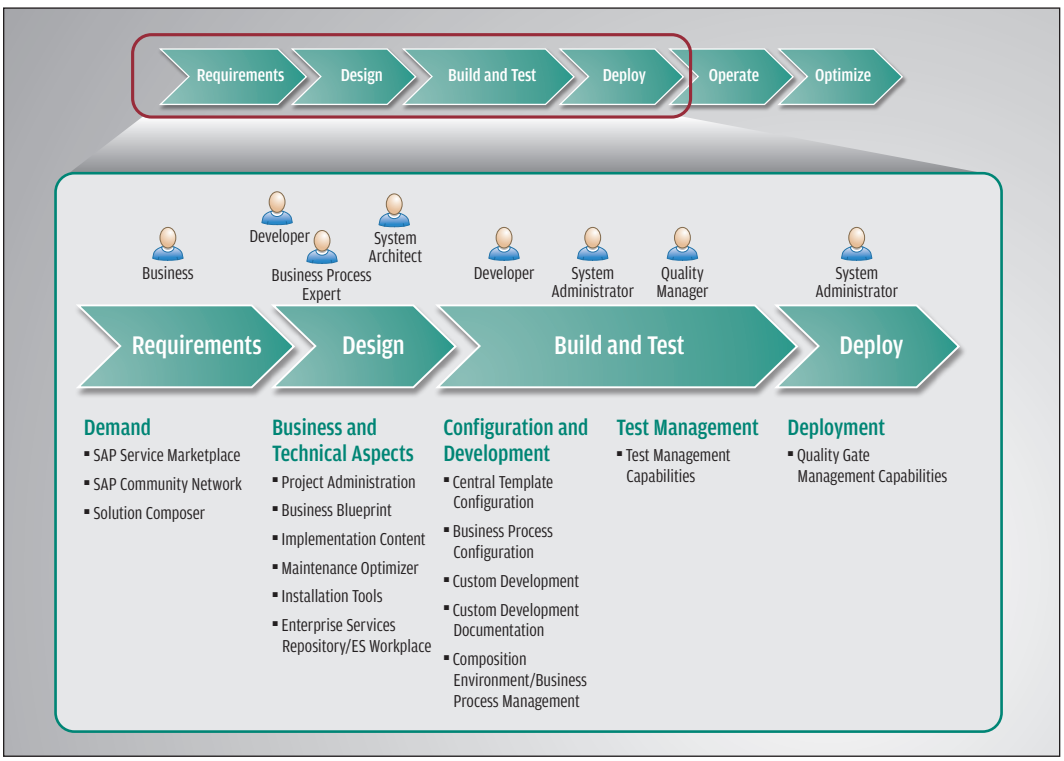

FIGURE 1 ◀ The tasks and roles associated with the Solution Implementation process, which spans several phases of the ITIL standard

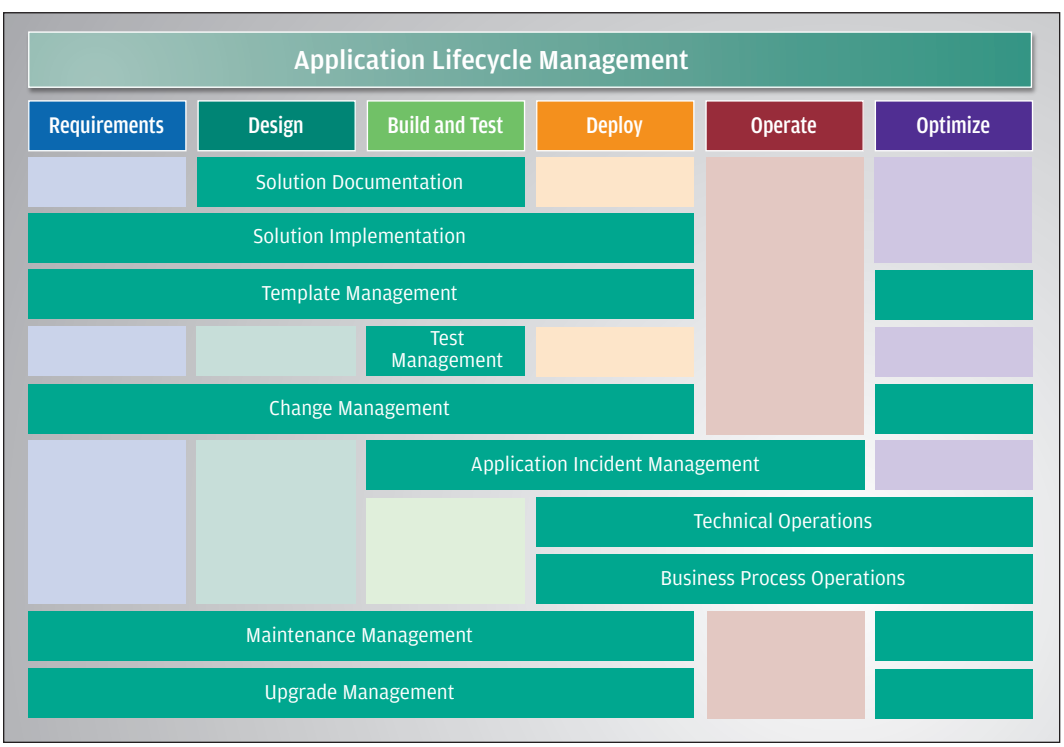

FIGURE 2 ◀ Most ALM processes span several ITIL phases

ALM Tools from SAP

- SAP Solution Manager
- SAP NetWeaver
  - ▶ Local monitoring capabilities
  - ▶ Software logistics capabilities
  - ▶ SAP NetWeaver Development Infrastructure
  - ▶ SAP NetWeaver Adaptive Computing Controller 7.2
- SAP Test Data Migration Server
- SAP IT Service Management
- SAP Resource and Portfolio Management
- Custom development management cockpit
- Solution extensions, such as the SAP Quality Center application by HP, the SAP Central Process Scheduling application by Redwood, and the SAP Enterprise Modeling application by IDS Scheer*

*See "8 Must-Have Tools for Your ALM Toolkit" by Kishore Bhamidipati on page 73 of this April-June 2010 issue of *SAPinsider* (sapinsider.wispubs.com) for more information about these solution extensions.

Business Benefits of a Rationalized Lifecycle Management Strategy

The major goals of an ALM strategy include:

- **Accelerated innovation.** IT teams and business owners need to discover and evaluate the value that innovation can bring into their landscapes while minimizing downtime.
- **Operation at a lower cost.** TCO is an ongoing challenge. Service-level agreements have to be met, which can only be achieved by standardizing, unifying, and automating operations.
- **Quality.** ALM is designed to assure quality management of your systems' functionality, performance, availability, and security.

process shown in Figure 1 to better understand the relevant — and interrelated — roles and tasks.

Consider a company's installation of enhancement package 4 for SAP ERP. During the Requirements phase, its implementation team analyzes and retrieves functionality and release information from SAP Service Marketplace or SAP Community Network, based on the requirements gathered from the business. The enterprise architects may use the Solution Composer tool to design a custom solution that meets the company's specific needs based on SAP's predefined process content.

Now in the Design phase, the implementation team uses various project administration tools to create a business blueprint for the project, laying out a plan for how the administrators will install, configure, and activate an enhancement package. An administrator uses the Maintenance Optimizer inside SAP Solution Manager to compute all the required downloads, software components, and archives to be installed in the solution landscape.

In the Build and Test phase, the implementation team selects the required business functions for activation. The team can use the Switch Framework Cockpit within SAP Solution Manager to execute the activation. The quality assurance team tests the enhanced business processes to ensure the quality of the innovation.

Then, in the Deploy phase — the last phase this process touches — administrators can import the activated business functions into the production system using the Switch Framework Cockpit and the SAP NetWeaver platform's change and transport system (CTS).

This is just a quick sketch of the full process and some of the more prominent tools involved, but it demonstrates the elegance of the model, providing clear guidelines with common best practices and a proven, standards-based methodology.

The ALM Journey Begins

ALM, aligned with the ITIL standard, sets up a framework for accelerated innovation, reduced TCO, and improved quality assurance (see "Business Benefits" sidebar). But with ALM, this is just the start of the journey. The current ALM blueprint and methodology has evolved over time, and new topics and tools will be added. Ultimately, ALM is designed to reduce costs, simplify and consolidate landscapes, reduce instances, and still maintain operations, even as business processes and organizations are consolidated in response to tough, changing business environments.

For more detailed information on ALM, visit **http://sdn.sap.com/irj/sdn/alm**. You can also link to additional resources from the online version of this article at **sapinsider.wispubs.com**. ■

✔ **NOTE!**
While tools such as the ABAP Workbench and SAP NetWeaver Developer Studio are not technically part of ALM, they are critical to the life cycle of any development, customization, and change and transport project.

Additional Resources... ...from SAPinsider

- "Benefit from the Upgrade Support of the Enhancement Framework" by Thomas Weiss (*SAP Professional Journal*, Volume 12, Update 1, **www.SAPpro.com**)
- *SAP Solution Manager, Enterprise Edition* by Marc O. Schäfer and Matthias Melich (SAP PRESS, **www.sap-press.com**)