DavidChappell
& Associates

WHAT IS APPLICATION LIFECYCLE MANAGEMENT?

DAVID CHAPPELL

SPONSORED BY MICROSOFT CORPORATION

Defining application lifecycle management (ALM) isn't easy. Different people (and different vendors) take quite different perspectives. Still, ALM is important, and so understanding what it encompasses is also important.

It's common to equate ALM with the software development lifecycle (SDLC). Yet this simple approach is too limiting; ALM is much more than just SDLC. In fact, an application's lifecycle includes the entire time during which an organization is spending money on this asset, from the initial idea to the end of the application's life. To be both accurate and useful, our view of application lifecycle management should take an equally broad perspective. Anything else just isn't right.

THE THREE ASPECTS OF ALM

ALM can be divided into three distinct areas: governance, development, and operations. Figure 1 illustrates this, showing each of these three aspects on its own horizontal line.

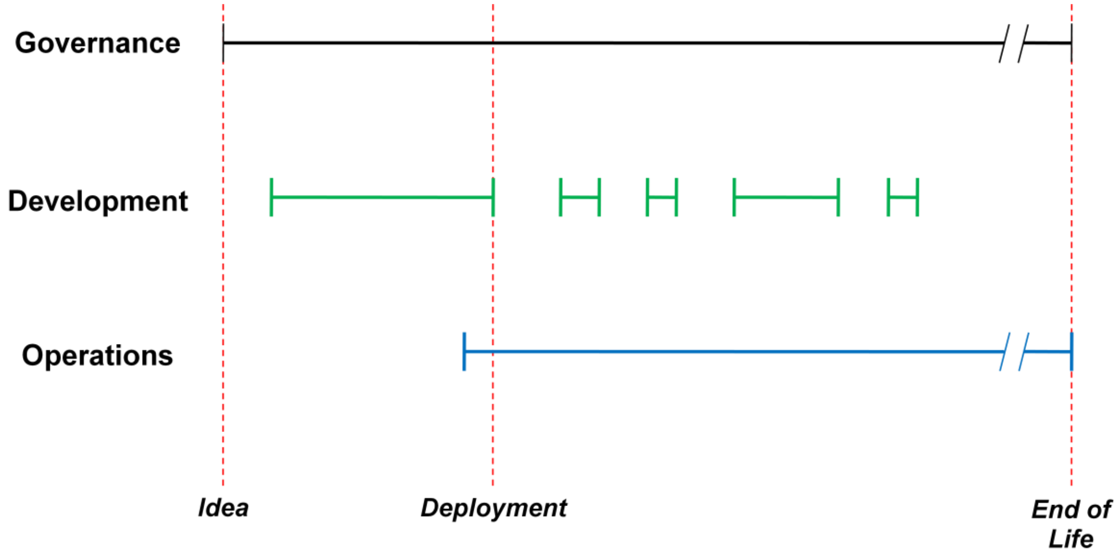


Figure 1: ALM can be viewed as having three aspects.

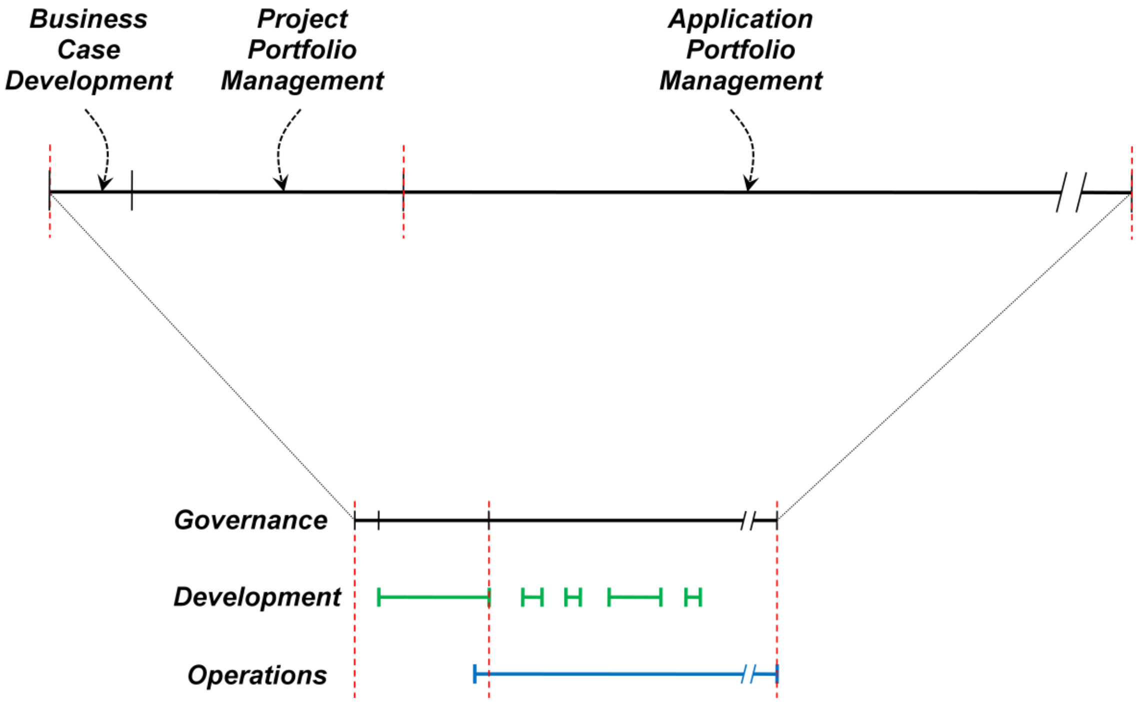
Like a human life, an application's lifecycle is demarcated by significant events. It begins with an *idea*: Why don't we build something that does this? Once the application is created, the next big event is *deployment*, when the application goes into production. And finally, when it no longer has business value, the application reaches *end of life* and is removed from service.

Like a human life, an application's lifecycle is demarcated by significant events.

Governance, which encompasses all of the decision making and project management for an application, extends over this entire time. Development, the process of actually creating an application, happens first between idea and deployment. For many applications, the development process reappears again several more times in the application's lifetime, both for upgrades and for wholly new versions. Operations, the work required to run and manage an application, typically begins shortly before deployment, then runs continuously until the application is removed from service. Each of these three areas is important, and so each is worth examining in more detail.

ASPECTS OF ALM: GOVERNANCE

In ALM, the purpose of governance is to make sure the application always provides what the business needs. Figure 2 gives a close-up view of ALM governance, providing a bit more detail on what it entails.



Business Case Development
Project Portfolio Management
Application Portfolio Management
Governance
Development
Operations

Figure 2: Governance extends over the entire application lifecycle.

The first step in ALM governance is business case development. As Figure 2 shows, this analysis happens before the development process begins. Once the business case is approved, application development starts, and governance is now implemented through project portfolio management. In some organizations, this is simple: A project manager might be attached to the development team, or one of the technical people on the team might take on this role. Other organizations use a more formal approach, relying on a centralized project management office to enforce established procedures. And with some processes, such as Scrum, the development team might not even have a formal project manager. However it's accomplished, some kind of project management is all but unavoidable.

Once the completed application is deployed, it becomes part of the organization's portfolio of applications. An application is an asset like any other, and so the organization needs an ongoing understanding of its benefits and costs. Application portfolio management (APM) provides this, offering a way to avoid duplicating functions across different applications. APM also provides governance for the deployed application, addressing things such as when updates and larger revisions make business sense. In fact, examining the APM section of the Governance line in more detail would show that it contains business case development and project portfolio management for each of the revisions to the application shown on the Development line.

Governance is the only thing that extends throughout the entire ALM time span. In many ways, it's the most important aspect of ALM. Get it wrong, and you won't come close to maximizing the application's business value.

ASPECTS OF ALM: DEVELOPMENT

While equating ALM with the software development process isn't accurate, development certainly is a fundamental part of every custom application's lifecycle. Figure 3 takes a closer look at this aspect of ALM.

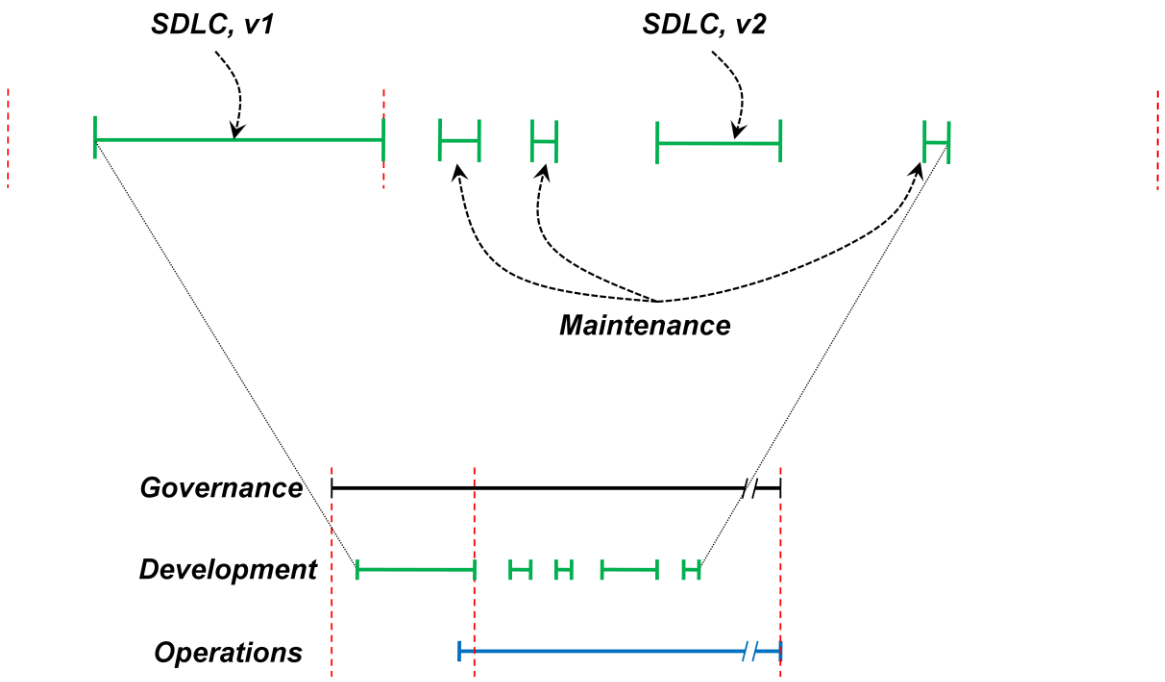


Figure 3: Development occurs in the first part of an application's lifecycle, then happens periodically as the application is updated.

Once the business case is approved, the software development lifecycle begins. If we expanded the SDLC parts of the Development line shown in the figure, a modern process would probably show software development as a series of iterations. Each iteration would contain some requirements definition, some design, some development, and some testing. This iterative style of development isn't always appropriate—some projects are still better done using more traditional methods—but it's becoming the norm in many areas.

Once the SDLC process for version 1 of the application is complete, the application is deployed. For most applications, however, deployment doesn't mark the end of development. Instead, the application needs periodic updates, as shown in the figure, and perhaps one or more full SDLC efforts to create new versions, as in this example. For some applications, the money spent on these updates and new versions can exceed the cost of the original development by a significant amount.

Once again, notice the role of SDLC in the overall ALM process. As Figure 2 shows, this aspect is certainly important, but it's far from the whole story. Viewing ALM as synonymous with SDLC is just wrong—it leads to a misunderstanding of what's really required to be successful in this area.

ASPECTS OF ALM: OPERATIONS

Every deployed application must be monitored and managed. Figure 4 shows some of the important aspects of this operations process.

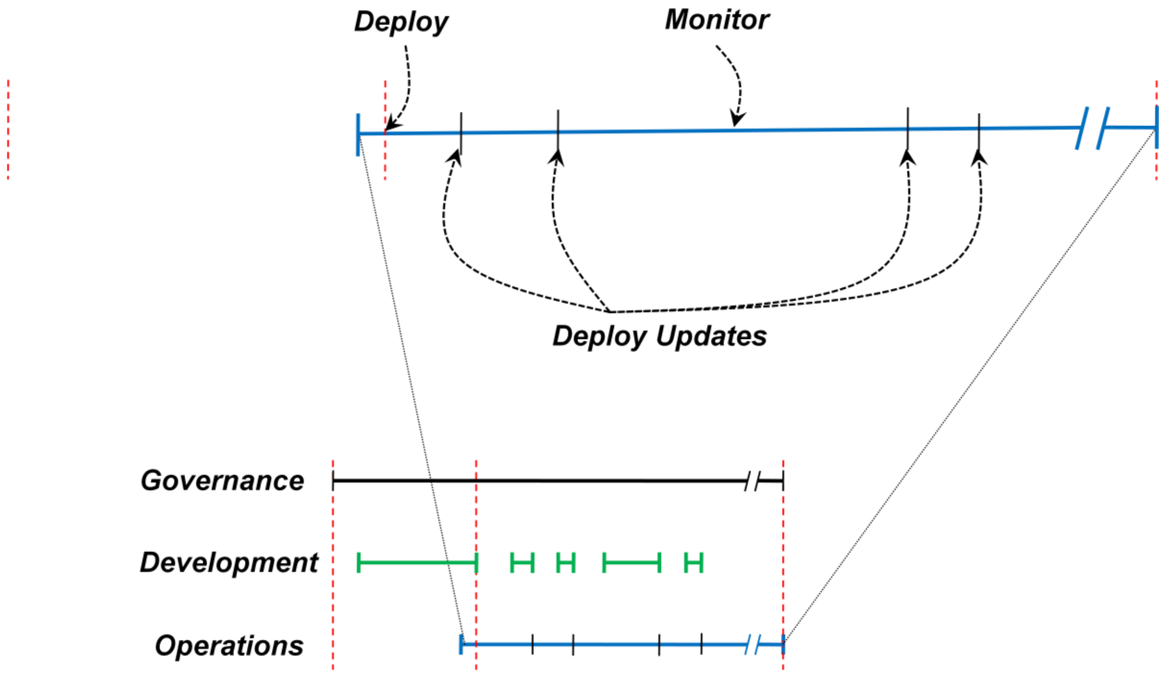


Figure 4: Operations begins shortly before an application is deployed, then continues until the application is removed from service.

As with Governance, the Operations line is intimately connected to the Development line. For example, planning for deployment likely begins shortly before the application is completed, and the act of deployment itself is a fundamental part of operations. Once the application is deployed, it must be monitored throughout its lifetime. Similarly, each update to the application must be deployed once it's completed, as the figure shows.

TOOLS FOR ALM

The three aspects of ALM—governance, development, and operations—are tightly connected to one another. Doing all three well is a requirement for any organization that aspires to maximize the business value of custom software. But this isn't an easy goal to achieve. Each of the three is challenging to get right on its own, and so getting the combination right is even more challenging.

The right tools can make this easier. A number of vendors today provide tools that are horizontally integrated, i.e., tools that work together well on one of the three lines. For example, Microsoft's Visual Studio brings together a range of tools supporting several aspects of the development process. Yet tools should be integrated not just horizontally but vertically as well, helping organizations make connections across the three lines. For instance, project management tools should be connected to development tools, which in turn should have connections to the tools used for operations.

Tools should be integrated not just horizontally but vertically too.

These connections are beginning to appear. Visual Studio, for example, can connect with Microsoft's Project Server to help project managers get up-to-date information on what developers are doing. There's still plenty of room for improvement, however, and no vendor today offers a set of ALM tools with full vertical integration across all three lines.

CONCLUSION

ALM is much more than just writing code. All three aspects—governance, development, and operations—are important. Think about a project that gets the initial governance aspects wrong, for example, perhaps by not understanding the business needs or failing to get the right stakeholders involved. No matter how well the organization does development and operations, this project won't provide much business value. Similarly, a project that targets the right problems using a first-class development process might ignore operational issues, such as providing enough resources to run the application reliably. Once again, the business value this investment provides won't be as large as it should be. Taking a broad view of ALM can help organizations avoid problems like these.

Maximizing the value of the applications we create means doing all three aspects of ALM well. Achieving this goal isn't easy, especially when today's ALM tools aren't as well integrated as they could be. Yet there's no way around it: Taking a broad, holistic view of ALM is essential for improving this critical business process.

ABOUT THE AUTHOR

David Chappell is Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technology.