

# SDLC & IDLC™

## Integrating Development and Operations

By: Harris Kern's Enterprise Computing Institute

Most mature infrastructure organizations use a Release Management process to maintain the integrity of their production environment. This is a simple but far-reaching statement in its implication to the scope of the Release Management process. The scope of the process can vary dramatically depending on the extent of other processes already implemented within the organization. Consider what it means to maintain the integrity of the production environment. First, it means achieving some initial level of integrity with superior levels of reliability, availability, and serviceability (RAS). Once that is accomplished, the environment must be “locked down” so that uncontrolled changes cannot negatively impact RAS. Finally, it means allowing changes, which are inevitable, to occur in the production environment but only in a controlled fashion.

Typically, the source of those changes is the infrastructure support group itself, new projects implementing new systems (hardware and software), and the software development team making modifications and changes to existing applications to support their clients. Any or all of these “changes” can have an impact on the production environment. This implies, then, that the release management process is very closely related to the change management process and the Application Development organization's software development life cycle (SDLC).

Referring back to the original statement, the scope of the release management process depends to a large extent on how well developed and far reaching these other related processes are practiced within your organization. If you have no change management process or a change management process that is geared primarily toward software development, then the Release Management process must pick up the change management slack and manage infrastructure changes. If your software or application development organization operates in an ad-hoc fashion and doesn't do thorough requirements analysis, design, testing, etc., then either Change Management or Release Management must make sure that the ad-hoc nature of their work doesn't negatively impact the production environment. Finally, infrastructure organizations are often their own worst enemy, modifying components of the infrastructure on the fly or with only minimal analysis and testing, essentially in an ad-hoc fashion. To maintain the integrity of the production environment, the Release Management process may have to account for all of these possibilities.

Having worked with many very large and prestigious companies, you may be surprised to know that we have yet to find one with both a strong Change Management process and a strong systems development lifecycle. Don't feel bad if you're in this boat. In order to achieve the goals of Release Management, we have implemented the IDLC process to cover those short-comings and you may have to do the same. It is interesting to note that if you implement some functions or steps as part of your Production Acceptance process to cover a short-coming in the existing change management process, those functions will eventually

migrate to where they belong, that is in this case, the change management process. Once they (these functions) have been in use for a while and the organization understands what they are and how they're used, it eventually dawns on people that this particular component of the PA process should be handled by change management, or in the software development group's SDLC. That's highly desirable so let the function migrate to the other process because you have enough to do. Further, let it evolve to that point and don't attempt to force the migration.

What is Production Acceptance if your organization has a mature change management process and a mature SDLC? If you're lucky enough to be in this situation, then production acceptance becomes the infrastructure organization's Infrastructure Development Lifecycle (IDLC) and the interfaces to change management and the SDLC. Just like the software or application development organization, the infrastructure organization needs a development lifecycle. The IDLC is focused on the infrastructure lifecycle as opposed to the software lifecycle. The infrastructure lifecycle has to cover project initiation, planning, analysis, design, construction, testing, deployment, close-out, and on-going operations, just like the software lifecycle. Most SDLCs have components that attempt to cover these functions for the infrastructure, but they don't do it very well. The only reason they do it at all is because infrastructure organizations don't have their own. That has to change. Infrastructure organizations have evolved far beyond "operations", as they used to be viewed. Today they're full blown development organizations that plan, design, implement, manage, and "operate" extremely complex environments, independent of the applications that are hosted in that environment. Infrastructure organizations have to evolve so that they can think holistically about the entire environment they manage, the applications they host, and the flexibility and reliability they must provide. They can no longer be driven by or be a victim of the software or applications development function. The infrastructure organization develops and manages an environment that hosts numerous applications for numerous customers. Infrastructure organizations must think and act as service providers. The very foundation of a service provider is a mature infrastructure development lifecycle and the services required to move applications and changes into that environment. This is Production Acceptance.

The infrastructure organization owns the production environment and the responsibility for RAS. To prevent changes to that environment from impacting reliability, you define the criteria for making and accepting changes to that environment. Now go one step further: instead of just telling someone else what they must do, the criteria they must meet to make a change or put something new into the production environment, provide that to them as a service. Now you're acting like a service provider. Think like a service provider and Production Acceptance becomes more clear. Think of yourselves, your infrastructure organization and the infrastructure you are responsible for, as a separate business at a separate location hosting applications for company XYZ (in this case it's your company). You have a service contract or a service level agreement in place guaranteeing XYZ certain levels of performance. The developers at XYZ want to introduce a new application into your environment. What do you do to make sure they don't cause a problem? This is your business, you've made guarantees, and even though XYZ's own developers may cause the problem, XYZ will blame you. Production acceptance becomes somewhat more clear.

One thing you can do is provide them with development assistance, at least as far as in helping to identify the infrastructure requirements. Then you can design an infrastructure solution for them. Then you can help them jointly test the application in the designed environment. Only after passing all the tests, you help them implement the solution. These are all services you provide that allow you to be proactively involved in the Systems Development Lifecycle (SDLC). You have ensured that your environment and organization are ready for deployment and ready for continued operations. You have also ensured that you can continue to meet your service level commitments. You can't be a victim when you have defined the criteria and proactively ensured that it has been met.

Production Acceptance is a process that allows you to act like an internal ASP by proactively managing the development, operations, and ongoing changes to the production environment. The process provides the criteria for modifying the production environment and defines the sequence of steps and activities to ensure that the criteria is met. The Production Acceptance process is the foundation for providing services to your customers, those that utilize your production environment, and becoming an internal application system provider. The process is tightly coupled with the change management process, the configuration management process and the software development lifecycle. Production acceptance, like all processes, introduces a certain amount of rigor so that it is possible to achieve repeatability and predictability, while providing for the flexibility required to run the business.