



## **Creative Shorts: The business value of Release Management**

*Quality management for deployment and delivery*

---

### ***A Creative Intellect Consulting Shorts Report Series (ALM)***

In this report, from our Creative Shorts series on Application Lifecycle Management, we put forward the business value of Release Management and its role in supporting the ALM process.

**Bola Rotibi, Research Director, Creative Intellect Consulting**

**Ian Murphy, Research Analyst, Creative Intellect consulting**

**June 2011**

## Shorts Overview

Businesses have a significant challenge to deliver applications reliably, timely, and against an agreed Service Level Agreement. The ones that succeed have the biggest returns from their IT investment as well as an ability to respond rapidly to changing priorities and operational models. This is important because businesses don't make any money from their applications until they are successfully running in production.

Release Management links developers and operations. It owns the processes that determine when software is ready for release and how and where it will be deployed. But it is not just about new software, Release Management also has a significant role to play in patch management and the stability of all software releases and updates. In a world where software deployment is increasingly being automated, Release Management must be taken seriously.

### *The demand for better Release Management*

- **Business adaptability and competitive advantage:** Companies invest significant sums of money into their IT estate in order to gain a competitive advantage over their competition. A substantial part of that investment goes into the tuning of software and the introduction of new features. To realize the value of the investment the new software needs to be delivered seamlessly and without impacting other systems. This requires business processes that span multiple divisions inside the IT department and provide a framework to ensure that changes add value and don't degrade existing systems.
- **Planning for change reduces business interruption:** It is uncommon for updates and changes to software to occur without some business interruption. To minimize the impact, organizations need to have effective processes to understand any potential disruptions. Among the many things covered by the processes are scheduling updates at times when disruptions can be minimized, being willing to restore and try again should changes take longer than expected, and a requirement for test deployments to identify unexpected issues. Without planning, a simple patch update can force the reboot of a critical server and have an impact on the entire business. If planned right, the business should never know anything has happened and end-users should see the addition of new features with the minimum of disruptions.
- **Complexity and more effective change control:** Even though most organizations have in place a release management process for deploying their applications into production, it is often resource intensive, time-consuming with little automation support. All this can make it a challenge for the IT organization to support change quickly and take advantage of business growth or innovation opportunities when they arise. One of the reasons that many companies do not have effective release management is that they believe it is too complex to fix. There is also the fear of fixing a process that although might be wildly imperfect, does get change deployed.
- **Correct tools reduce the complexity of release management:** To be effective, Release Management has to cross both organizational and operational boundaries. However, this is not necessarily the problem. More often than not, the challenge is one of wrong tools and ineffective communication. Many companies rely on email, spreadsheets, and word processing to track changes rather than acquiring proper tools. The use of change control, collaboration, code repository, deployment automation, and project software, along with workflow processes to schedule releases, reduces inadequate paper trails and ensures that the process is transparent and available to all the relevant parties.
- **Working with other teams:** All too often, Release Management teams are not an intrinsic part of the various teams that they need to work with. Release Management should be part of the whole development team so that they understand what is happening in development. They should be part of the operations teams, especially the change management teams responsible for the updating, patching,

and delivery of applications so that they have immediate visibility of what is happening on the ground. They also need to be part of the domain expert team that is consulted whenever changes are being planned or considered.

- **Quality improvement, cost containment and time to market in the face of rising user expectations:** Only when Release Management is fully integrated into the entire application lifecycle, from planning to development, operations to deployment, patching to change management, can Release Management unlock the ability to deliver high quality service level agreements for software and services.

To meet the speed of change and the need to effectively deliver software, organizations must develop Release Management teams to reduce the risk of operational failure and provide the governance and processes that IT and the business requires.

## ***Release Management in a nutshell***

Release Management is the part of an Application Lifecycle Management (ALM) process responsible for the management of software released. This includes new software, software updates, software patches, and the development of manageable, repeatable and quality-driven policies and processes that allows the IT department to meet the needs of the business. Unlike other parts of the ALM process, release management spans users, developers, and operations giving it a unique view across the entire business.

The purpose of Release Management is to ensure that a consistent method of deployment is followed across the whole application lifecycle and that the likelihood of incidents and complications as a result of rollouts are kept to a minimum. Only tested and accepted versions of hardware and software should be released for installation at any time. The foundations for good Release Management support are:

- Management and control
- Integration
- Process and workflow automation support
- Repository store for released code
- Intuitive and user friendly interface portal

The release process is a good indication as to the strength of the upfront delivery processes and the overall reliability of the ALM process. A well implemented process that is strongly interconnected will support rapid release cycles.

## ***Guidance strategy for Release Management***

Effective development and deployment of Release Management requires the correct processes, tools, and strategies. This way companies will create a transparent, audited environment allowing release managers to focus on their core responsibilities while providing information to all the stakeholders from users to IT operations.

### **People and processes:**

- I. **Manage ALL environments:** It is not enough to just manage the production environment. Test environments also need to be managed to establish a set of baselines and processes that are central to all software deployments. This is not just about developers but must apply to any member of the operations team testing software patches and updates.

2. **Take control of test acceptance:** Software testing is often squeezed when developments run late. Release Management should ensure that no software is allowed into production without having been properly tested.
3. **Employ a secure path to production with a common repository:** Develop a common process that ensures all release ready software goes into a common repository with the appropriate links to dependent artefacts (e.g. sign-off documentation) held in other federated systems. Only ready-to-be-released software from the repository should be deployed. This provides substantial auditing and quality control.
4. **Common processes for all release types used:** Not all releases are the same. Most organizations will categorize or typify their releases as being one that is either a major, minor, patch or an emergency. Release managers should have a common process for all the different release types they employ.
5. **Set change limits:** The larger and the more numerous the changes and updates that are included in each release, the more complex the testing and the greater the risk of failure. To reduce this risk, set a limit on the size and number of changes to make it more manageable. Fewer changes on a more frequent, but still controlled scheduled, improves quality and delivery.
6. **Integration is vital:** No ALM process is an island since they all interconnect, influence and have an impact on the other. An efficient and successful release process is indicative of effective execution of the preceding ALM processes and the downstream operations with strong handover and integration policies at the interfaces. The later you are in the ALM process, the more expensive it becomes to rectify problems and the greater the impact to quality management and delivery schedules. Release Management considerations need to be addressed at important junctures in the ALM process: from user requirements, to software design, and through to the operational deployment, maintenance, and management.

## Tooling strategy:

Without the right level of tools linked to effective processes, it is impossible to efficiently control Release Management.

1. **Get the right tooling support:** Spreadsheets and document files might work when you have relatively few changes, but the more complex the environment and the more changes you have, the harder it becomes to manage and improve the process. Collaboration tools, processes, workflows, release code repository, and release automation software provide a more automated and transparent approach.
2. **Automate where you can:** Many parts of the release management process can be automated. For example, if you are using Agile processes, use test driven development to automate the software development, testing, delivery, and deployment. Deploy application-centric release automation tools to automate the last steps of release. This improves release time and quality, reduces release failures, and enables you to release more during each release window.
3. **Integration is a necessity:** As pointed out earlier, Release Management requirements need to be considered at key stages of the ALM process. A robust integration strategy and framework is needed to ensure cross tool interaction and collaboration.
4. **Focus on ease of use and engaging user friendly interface portals:** Collaborative tools provide useful dashboard interfaces to make it easier to show everyone what is happening and when. This transparency reduces a lot of the communication overload on Release Management teams

## Business engagement:

Effective Release Management for software is as critical as the planning for a new product. It is essential therefore that there are clear management and business engagement goals, around which Release Management can be measured.

1. **Collaborative engagement for quality release:** Beating the competition is more than about getting there first, it's about getting there right. Hard dates must be given to the Release Management team in advance so that they can effectively plan releases to meet the business demand.
2. **Expose the dangers of cutting corners:** One of the biggest challenges for Release Management is making sure that the quality goals are met. Make sure that there is authority in the Release Management process and team to reject untested or poorly tested products. All it needs is one major system failure such as a website that cannot cope with demand and the costs will be felt by the business. The business needs to be made aware of the dangers of the pressure to let through potentially damaging software.
3. **Proactively monitor performance and use the tools and portals provided:** One of the advantages of having a collaborative environment using dashboards to provide status information is that you can spot problems early in the cycle. Exposing the indicators that reflect the health of the release management process and linking them to important business metrics or service delivery goals will help provide the business with clearer insight for action.

Although it is clear that tools, process, and people are all critical factors to consider to efficiently control Release Management, it can still be challenging to get it right. We believe the guidance strategy we've outlined in this paper will help you minimize the risks and delays around Release Management and increase your chances for success.