DZone Refcardz

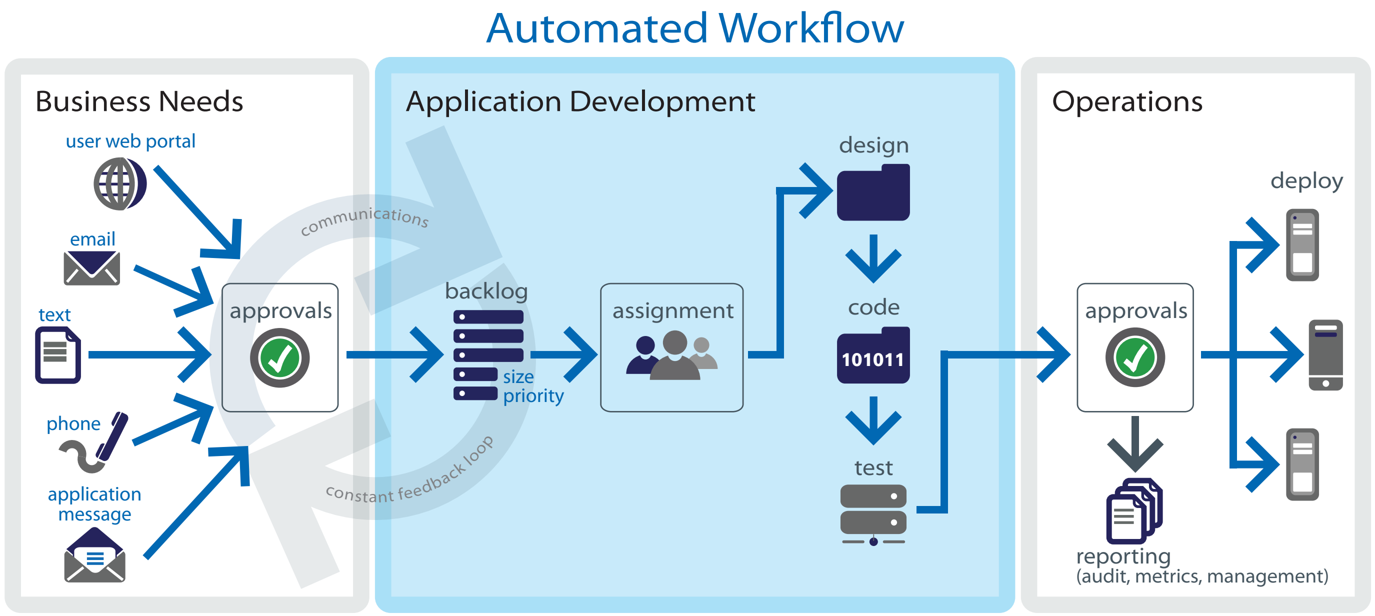
brought to you by... Aldon
Change. Collaborate. Comply.

CONTENTS INCLUDE:

- What is ALM?
- End-to-End Application Lifecycle Management
- Change Request Tracking
- Inventory Control
- Process Automation
- Deployment Automation and more...

Getting Started with Application Lifecycle Management

By Daniel Magid



Application Lifecycle Management comprises all the functions that must be performed to appropriately automate, control, document and facilitate everything that occurs from the time the need for an application change arises until the time a solution is delivered into production. Whether you are working in a traditional waterfall environment or an agile environment or something in between, Application Lifecycle Management provides the tools and process infrastructure to support you in achieving your goals.

WHAT IS ALM?

Application Lifecycle Management (ALM) encompasses the automation, control and tracking of all activity involved in delivering software solutions for resolving business challenges and taking advantage of emerging opportunities. When appropriately implemented, ALM simplifies the lives of everyone involved in the change process, improves communication, automates regulatory compliance and supports the organization in rapidly providing high quality solutions to the highest priority business requirements.

END-TO-END APPLICATION LIFECYCLE MANAGEMENT

Automated ALM can simplify and improve the working environment for everyone involved in application change by:

- Automating repetitive daily tasks like builds, unit tests, promotions and deployments
- Automatically tracking activity and generating audit compliance reports
- Routing changes through the appropriate workflow and notifying staff members of task assignments
- Providing an easily accessible central repository for all change related communication
- Allowing access to change backlog from existing IDEs and version control tools
- Helping to keep the backlog of work visible and organized
- Keeping track of application artifacts (programs, documentation, test cases, etc.)
- Managing access to application artifacts

- Ensuring all the right application parts end up in all the right places throughout the change lifecycle

For organizations transitioning to Agile methodologies like Scrum or Kanban, ALM provides the backlog management, communication and reporting capabilities that are necessary in order to truly realize the promise of Agile.

In this Refcard, we will examine the components of ALM systems, the challenges they are meant to address and how they can be used effectively. We will begin with the basics and then discuss how ALM applies to different management methodologies like Waterfall and Agile.

Comprehensive end-to-end ALM addresses these important areas:

- Change Request Routing and Tracking (What is it we are to do, who is to do it and in what sequence?)
- Inventory Management (What artifacts need to change? Who can change them?)
- Process Automation (How does a change flow through the lifecycle?)
- Deployment Automation (How do changes get to their ultimate destination?)
- Compliance Reporting (How do I keep the auditors happy?)

CHANGE REQUEST TRACKING

Whether you are working in a Waterfall environment, an Agile environment or something in between, the application change process always starts with a business requirement. Someone discovers a bug in the existing application or requests an enhancement or needs an entirely new system. At that point, they need to communicate their need to application development in order to pursue a solution. Many of the problems in managing application change begin with faulty processes for gathering and prioritizing change requests.

Request Capture

Change Requests must be captured at the source — directly from the user(s) who require the change. Business analysts and developers will need to take that information and work with the end user to define exactly what the system must do to fulfill the need, but it is important never to lose sight of what the user said that they want. As the change process progresses, there will undoubtedly be new requirements that are discovered and modifications to the original request. The ALM System makes these easy to track. This continually up-to-date information should drive the design of the solution and the creation of the tests that will be used to determine when a change is 'done' — i.e. whether it meets the need the user described in an appropriate way.

It is incumbent on the users to ensure that they identify the business value of the requests they submit in order to drive the prioritization process. The ALM system can automatically route the change request to the appropriate end user team members for review and value assignment, gather the signoffs, record the responses and then, if approved, forward the request to application development.

The ALM system:

1. Provides a simple method for entering requests
2. Ensures they contain the necessary detail
3. Notifies people who must be involved in the review
4. Gathers and records approvals
5. Routes the request through the appropriate workflow
6. Records all communication regarding the request to ensure nothing is lost
7. Provides a method to generate reports regarding activity
8. Keeps the request accessible to all authorized stakeholders

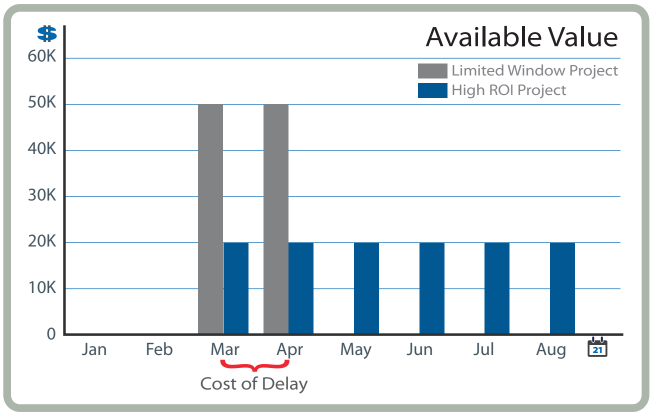
These are all necessary steps in most compliance and best practice standards for ensuring that the appropriate information is tracked and that nothing is 'lost'.

Request Prioritization

Once a Change Request has entered the system, it must be prioritized so the application development staff knows which things to do first. Inadequate prioritization is a major contributor to misalignment between application development deliveries and business needs. It is the responsibility of the business users to adequately communicate those priorities to application development. The ALM System can record business value information and priorities and make them visible to all the appropriate team members and stakeholders. High performance application development organizations use the ALM system as a tool to constantly review their change backlog to ensure they are working on the changes that provide the greatest business value.

The ALM System can be a useful tool in creating prioritized lists of changes based on business value. It can provide users with the capability to display a filtered list of backlog items based on whatever criteria the user finds helpful. Users can then move items around in an ordered list until they are in descending order by value with the most valuable changes on top. The ALM System provides visibility to the prioritized list for all authorized, interested stakeholders.

More sophisticated methods of prioritization involve assigning specific economic values to change requests. The most common method is to assign an ROI value to the project. However, the problem with ROI is it does not consider the value of timing. There may be a project in the backlog with a very high ROI relative to other projects, but that ROI is



A two-month delay will only cost us a fraction of the high ROI project value, while we will lose all of the value of the Limited Window project.

available whenever the project is delivered. Conversely, there may be a project with a lower overall ROI, but the value it provides might be highly dependent on when it is delivered. In that case, the second project has a higher Cost of Delay. In that case, application development might want to deliver the project with the higher Cost of Delay first (see chart on page 2). Whichever method you choose, the prioritization justification should be visible in the ALM System so that everyone is aware of how decisions are being made. Once the change has been delivered, actual values can be recorded so that the decision making process can be validated.

If a choice must be made between two projects, one with a very high ROI and the other with a smaller ROI but with a higher Cost of Delay, the choice should be the project with the higher Cost of Delay. In the example above, there are two projects. Each will take two months to implement. However if the March delivery date is missed, the entire value ($100,000) of the Limited Window Project will be lost while only two months of return ($40,000) of the High ROI Project will be lost. In scheduling and managing development, it is critical to understand and record the Cost of Delay. (See Donald Reinertsen's book "The Principles of Product Development Flow" for an excellent discussion of the importance of tracking Cost of Delay.)

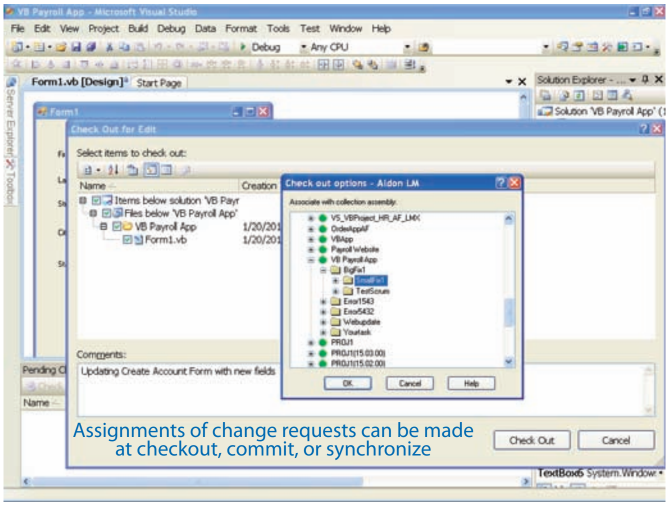
Hot Tip

At Aldon we ran into this situation and our ALM system made it immediately visible. We were implementing a restructuring of our database that would provide a platform for a variety of new enhancements to the product. Once those enhancements were delivered, we anticipated a large ROI for the project. At the same time, our salespeople were telling us that we could win a greater percentage of our current opportunities if we would make some changes to our user interface. The ROI for those changes was smaller overall than the architectural change, but the window of opportunity for winning the current deals was relatively short. We decided that the cost of delaying the big project by a few months was relatively minor while the cost of delaying the changes required by our prospects was much higher. We chose to deliver the interface changes first and delay the restructuring. By recording the relative value information in the ALM system, everyone understood the justification for our decision.

Request Tracking

The ALM system allows request tracking to occur automatically as a by-product of working together on a Change Request, so developers need not be burdened with time-consuming activity documentation requirements.

Once a Change Request has been prioritized and approved for work, the ALM System routes the change through its appropriate workflow and notifies users when they have a task to perform. It records signoffs and state changes so that important metrics like cycle time, velocity and wait-times (bottlenecks) can be recorded and managed. Developer activity is automatically logged as they perform operations like check-in, build and unit test. By automatically gathering this information as part of normal day-to-day activities, the ALM System can significantly reduce the administrative documentation burden placed on developers by audit and regulatory compliance requirements.



Comments: Updating Create Account Form with new fields

Assignments of change requests can be made at checkout, commit, or synchronize

Integration with the developer's preferred IDE allows them to keep the ALM system up-to-date without being saddled with a lot of additional administrative work. In the example above, a developer associates the changes they are making to an ALM task directly through Visual Studio.

Developers can associate their changes with a Change Request directly through their IDE or version control system at the time they Commit, Check-out or Synchronize. Development change activity can then be tracked and reported upon with very little administrative impact on developers.

By tracking the Change Request from initiation to completion, the ALM System will be able to identify for the development team and other stakeholders how long it takes changes of different sizes to make it through the system. These 'cycle times' can then be used to make deliveries more predictable.

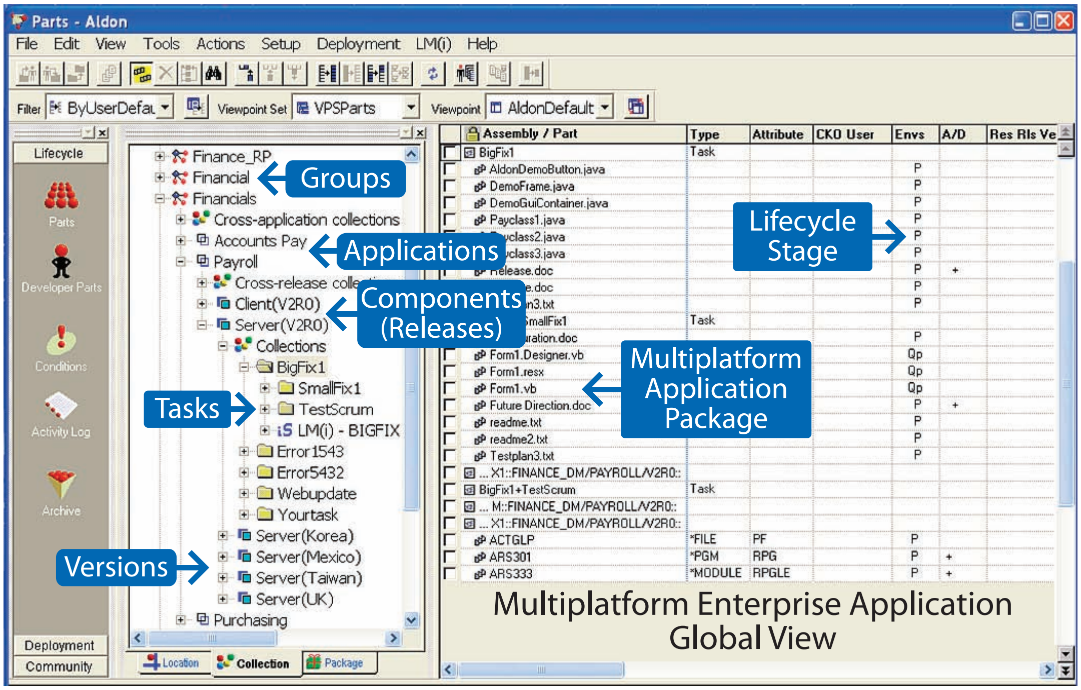
Later, the activity logs can be used for compliance reporting. Auditors will be able to use the ALM system to view exactly what happened from the time the request arose until the solution was delivered into production. They will be able to easily verify that the change process met the audit requirements.

INVENTORY CONTROL

All applications have an inventory of related application artifacts. These include files of program source, build results, build scripts, test scripts, documentation and many other types of files. Managing and tracking these files can be especially difficult in today's integrated, multi-platform application environment. A single application might have application logic residing on a mainframe with a mainframe database while using Windows clients and supporting Unix or Linux Web Application Servers for a web-based interface. Changes to any of those environments might need to be synchronized with changes to the other environments.

As developers we might only be concerned with one part of that architecture (the Windows client, for example). However, we might be impacted by changes to the other environments.

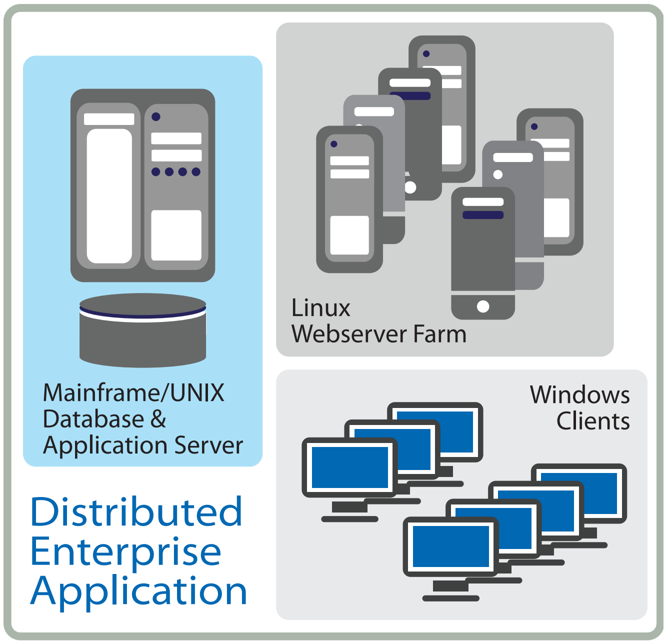
The ALM system can ensure that we are aware of those cross platform dependencies. It can also provide us with a business oriented view of our environment so we can see all work-in-progress for an application, regardless of the targeted platform. The ALM system does this by maintaining a Meta-Data inventory of the current location, and status of every application artifact regardless of its location, and making that information easily accessible and viewable.



Assembly / Part	Type	Attribute	CKO User	Envs	A/D	Res Rls Ve
BigFix1	Task					
AldonDemoButton.java				P		
DemoFrame.java				P		
DemoGuiContainer.java				P		
Payclass1.java				P		
yclass2.java				P		
yclass3.java				P		
elease.doc				P	+	
e.doc				P		
n3.txt				P		
SmallFix1	Task					
ration.doc				P		
Form1.Designer.vb				Qp		
Form1.resx				Qp		
Form1.vb				Qp		
Future Direction.doc				P	+	
readme.txt				P		
readme2.txt				P		
Testplan3.txt				P		
... X1::FINANCE_DM/PAYROLL/V2R0::						
BigFix1+TestScrum	Task					
... M::FINANCE_DM/PAYROLL/V2R0::						
... X1::FINANCE_DM/PAYROLL/V2R0::						
ACTGLP	*FILE	PF		P		
ARS301	*PGM	RPG		P	+	
ARS333	*MODULE	RPGLE		P	+	

Through the ALM inventory views, users can see application artifacts and work in progress based on their business application structure. They simply need to know which task, application or application version they are interested in and they can see where the associated changes are within the application lifecycle across all of their operating environments. There is no need to know all the file/directory/library structures of the underlying platforms or to constantly switch machines to see what is going on.

Although developers will typically work through their preferred application development tools, the ALM System can provide them with a more global view that reveals the business application structure and shows where all the application parts are and how they are related across all platforms. In this view, users can see all the parts related to an application, a component of an application, a version of an application or an application task even if those files are stored on multiple platforms and run in many different operating environments. They can also see where those files are in the application lifecycle. The ALM System will notify them if a change in one area will impact files in other areas.


Mainframe/UNIX Database & Application Server
Linux Webserver Farm
Windows Clients
Distributed Enterprise Application

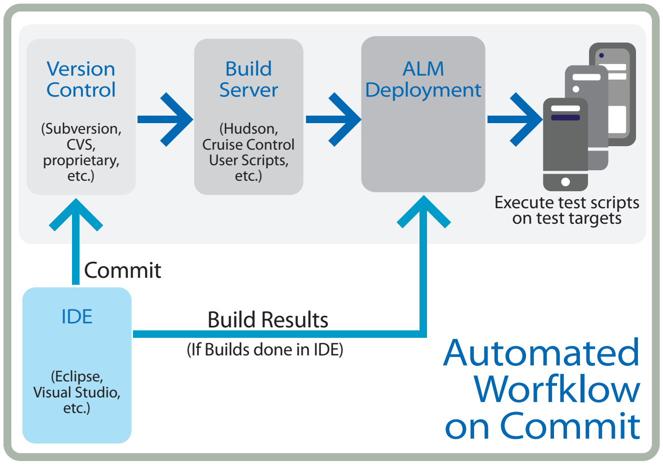
Most modern enterprise applications span platforms and operating environments.

The ALM System also provides access control over the artifacts in inventory. Authorization to change application artifacts can be limited based on the user's role in the organization, the applications in which they are authorized to work, the stages of the lifecycle in which they are authorized to make changes and a variety of other criteria.

PROCESS AUTOMATION

The goal of the ALM System is not to force a particular process on the organization but rather to automate and enforce the processes the organization wants to implement. The ALM System must do its work with as little administrative impact on the developers as possible. Consequently, most ALM systems allow developers to access their functions directly through the developer's preferred IDE. As developers commit, check-out, check-in or synchronize their changes with their source version repository, the ALM system gathers the changed artifacts and moves them through the appropriate lifecycle. If you are in a Continuous Integration environment, upon check-in, the ALM system can kick off a build and then deploy the build results to the appropriate test environments and even initiate unit tests and regression tests.

The ALM system will promote changes through whatever lifecycle stage has been defined to it (e.g. Development, to Integration Test, to Quality Assurance, to User Acceptance testing and then on into production), ensuring that the versions of all the files that were tested are the ones that move into production. It can be configured to support and automate a variety of Agile or traditional methodologies. The ALM system will gather approvals to ensure the appropriate separation of duties has been enforced, and then deploy the files to the appropriate machines for execution. At each stage users simply need to tell the system that their process is complete and the change is ready for the next step. The system automates the movement of the related artifacts, eliminating manual, administrative effort.


Version Control
(Subversion, CVS, proprietary, etc.)
Build Server
(Hudson, Cruise Control User Scripts, etc.)
ALM Deployment
Execute test scripts on test targets
Commit
IDE
(Eclipse, Visual Studio, etc.)
Build Results
(If Builds done in IDE)
Automated Workflow on Commit

If you are working in a Continuous Integration environment the ALM system can automatically kick off the build and deploy process immediately upon a commit to version control.

In an Agile environment, the ALM system can ensure that the changes flow through each of the lifecycle stages within the time-boxed iteration. If Lean-based techniques are being used, the ALM system can limit the number of Change Requests that can be moved to Work-in-Progress or a particular stage of Work-in-Progress based on user-defined boundaries.

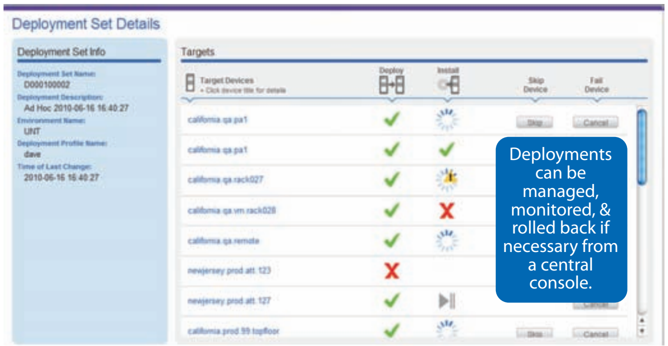
Since process improvement is a key element of any successful application development methodology, the ALM system should have flexible rules definition so that the process can be changed and improved as users learn about what works and what does not work.

The main purpose of the ALM system is to eliminate the need for users to worry about the process. The process definition is encoded into the system and it ensures that everything happens as planned. If the process needs to be modified, the ALM system should provide simple mechanisms to make those changes. Importantly, all the process documentation is embedded within the system. Auditors can generate reports of the defined process without relying on manually maintained (and generally poorly maintained) process documents.

DEPLOYMENT AUTOMATION

At each stage of the lifecycle, completed changes must be deployed to the appropriate target machines for testing or production. A single change in an enterprise environment might require synchronized updates to a variety of clients and servers. Specialized installations scripts might need to execute in order for the change to be implemented successfully and, in case of failure, there must be a mechanism to recover.

The rules regarding deployment for each stage of the lifecycle can be defined to the ALM system. That way, users simply need to identify what needs to be moved. The system knows where the artifacts are and where they need to go. The deployment can then be monitored and managed via the ALM System.



Deployment Set Details

Deployment Set Info

Deployment Set Name:
D000100002
Deployment Description:
Ad Hoc 2010-06-16 16:40:27
Environment Name:
UNT
Deployment Profile Name:
dave
Time of Last Change:
2010-06-16 16:40:27

Targets

Target Devices • Click device title for details	Deploy	Install	Skip Device	Fail Device
california.qa.pa1	✓		Skip	Cancel
california.qa.pa1	✓	✓		
california.qa.rack027	✓	⚠		
california.qa.vm.rack028	✓	✗		
california.qa.remote	✓			
newjersey.prod.att.123	✗			
newjersey.prod.att.127	✓	▶‖		Cancel
california.prod.99.topfloor	✓		Skip	Cancel

Deployments can be managed, monitored, & rolled back if necessary from a central console.

ALM AND AGILE

In addition to the benefits described above, the ALM system can be indispensable in implementing Agile or Lean processes. For teams that are working in Agile or Lean environments, the ALM system can provide the infrastructure for collecting metrics on velocity, flow, queuing and cycle time. These are necessary measures to truly deliver on the value promised by Agile methodologies.

In a Scrum environment, the ALM system can track changes as they are completed and generate Sprint or Release burndown (or burnup) reports. At the end of each Sprint it can track the number of story points delivered by each team. Over time, this will allow the teams to establish their delivery 'velocity'. That velocity becomes a critical component in creating predictable estimates of future deliveries.

In Kanban or other Lean environments, the ALM system can be used to establish limits on work-in-progress — improving the flow of changes. The time that changes spend in queues can be measured and improved upon. This will allow for the removal of bottlenecks and improvements in cycle time. Over time, predictable cycle times for changes can be established.

COMPLIANCE REPORTING

Regardless of which methodology an organization chooses to use to manage application maintenance and development, it is still subject to audit and compliance requirements. Meeting those requirements can place a heavy administrative burden on application development. The ALM system can dramatically reduce that burden and free development staff to focus on delivering great software.

Embedded process documentation — since the workflow process has been defined to the ALM system, up-to-date process documentation can be generated on demand directly from the system.

Activity reporting — the ALM system is automatically logging every activity, so at any time, auditors can see exactly who did what in association with a change request. Actual code changes can be tied to change requests so they can even drill down to see specifically what each developer changed.

Process automation and enforcement — users define the approved workflow to the system and then the system automates and enforces that process. Auditors can query the activity logs to ensure that the appropriate steps were followed.

Separation of duties — auditors can view user roles and functional authorizations to ensure that only the appropriate people can perform specific operations (e.g. developers or testers can request a move to production but a change control administrator must approve the request before anything can move forward). They can then review the activity reports to see that those rules were enforced.

Service level reporting — Does IT have service level agreements with business end users? If so, the ALM system can report on service level compliance and violation.

Importantly, this information is being gathered as a by-product of normal day-to-day activities reducing the impact of compliance efforts on product deliveries.

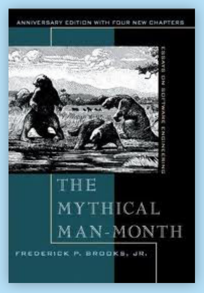
END TO END LIFECYCLE MANAGEMENT

ALM systems make it easy to automate and track everything that happens from the time the need for a change arises until the time the solution is delivered to the end users while reducing administrative burdens on developers. Users have an easy way to enter requests and developers have a powerful way to review and work with those requests. At every stage of the process, notifications are sent out regarding assignments and all activity is tracked. Auditors have a simple way to generate comprehensive reports and queries on activity and process compliance — reducing the need for development staff members to provide those reports. ALM systems automate administrative procedures and make information about backlogs and work in progress available to all authorized stakeholders. With an ALM System in place, developers can focus on supporting the business with high quality solutions.

ABOUT THE AUTHOR

Daniel Magid, Chief Technology Strategist for Aldon and a recognized authority on helping leading organizations achieve compliance through build release management and application lifecycle management solutions. He has written a variety of articles for leading IT publications and is a regular speaker at technology conferences. With Magid's guidance, Aldon has received broad industry recognition as the clear leader in the process-driven change management.

RECOMMENDED BOOK


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.

No book on software project management has been so influential and so timeless as The Mythical Man-Month. Now 20 years after the publication of his book, Frederick P. Brooks, Jr. (best known as the "father of the IBM System 360") revisits his original ideas and develops new thoughts and advice both for readers familiar with his work and for readers discovering it for the first time.

BUY NOW

books.dzone.com/books/mythical-man-month

DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more.
"DZone is a developer's dream," says PC Magazine.

DZone, Inc.
140 Preston Executive Dr.
Suite 100
Cary, NC 27513

888.678.0399
919.678.0300

Refcardz Feedback Welcome
refcardz@dzone.com

Sponsorship Opportunities
sales@dzone.com

ISBN-13: 978-1-934238-75-2
ISBN-10: 1-934238-75-9
50795
9 781934 238752
$7.95

Version 1.0