



# Application Lifecycle Management Solutions in the Cloud:

Crucial Features for  
Adding Value to the  
Software Development Process

By  
Steven Hamilton  
Software Architect  
TOMOS Software



*"Cloud computing is fundamentally about re-engineering the world's computing infrastructure, to enable game-changing -- even life-changing – applications."*

*- Willy Chiu, Vice President, IBM High Performance On Demand Solutions<sup>1</sup>*

## **Introduction**

In the software development field, the rewards that can be gained from an organized Application Lifecycle Management (ALM) process have been thoroughly documented;<sup>2-3</sup> however, what do ALM solutions that are based in “the cloud” bring to the table for a project? What features should these tools incorporate? More specifically, where is the return on investment (ROI) found by having these features available to the different areas of a software development project?

This paper will detail the most desirable options and features in a hosted ALM application and explain where the ROI for each exists. Some of the following items will undoubtedly be the types of features that any ALM solution should incorporate, but they become “must-haves” when the application is hosted. The features in question are listed below in order of importance.

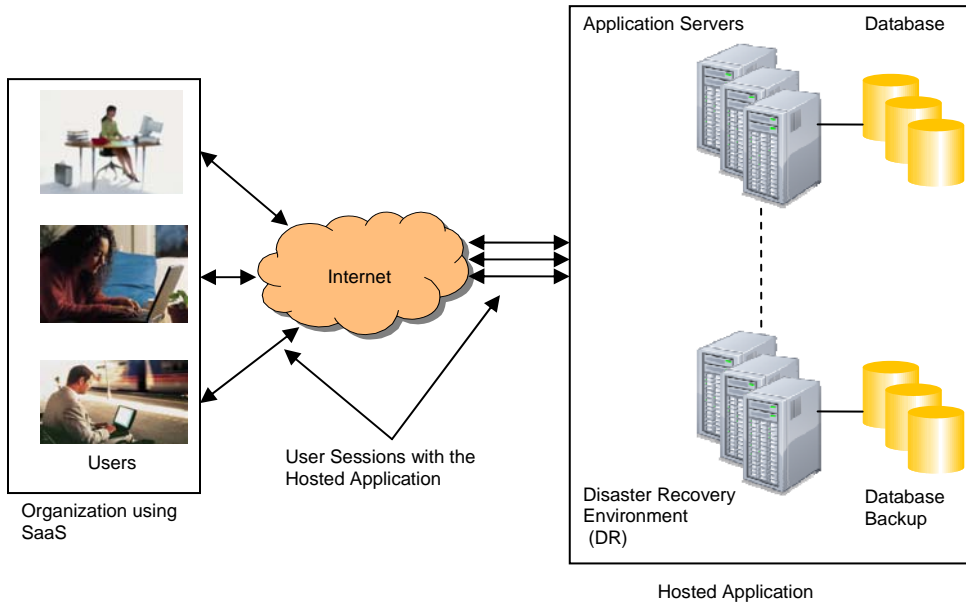
- Appropriate Tool for Application Lifecycle Management
- “Right-sized” Tool Cost
- Integrated Software Quality Methodology
- Security
- Support
- Roles
- Data Integrity and Application Availability
- Traceability, Reporting, and Audit Trails
- Minimal Ramp-up Time
- Integration with External Tools

## **The Cloud Explained**

A brief explanation of “The Cloud” and some examples of its use in today’s market should provide some background for the discussion that will follow. Cloud computing includes several different subsets (Utility Computing, Web Services, Managed Service Providers, Software as a Service (SaaS<sup>4</sup>)) but their common thread is the availability of a resource over the Internet, usually on a large scale, with the benefit of little to no knowledge of the inner workings of said resource for the subscriber.

Many examples of subscribed services such as this have existed in our lives for many years, such as the telephone or cable television, both of which provide a service to the user for a fee without the user needing to know the inner workings of a telephone switching system or physically make connections to cable wires in the street. SaaS applications, such as Salesforce and Google Apps, provide important enterprise-level services in much the same way.

Figure 1. Hosted SaaS Application Diagram



The benefits of utilizing services in this way can be considerable. Instead of hiring an entire development team, building an environment, and maintaining it in order to create an application to fulfill a business need, an application is provided from outside the organization that uses it for a fee. The application is made available to the users in the form of a Web Application that is created and maintained by an external source. The hosted application is accessed via the Internet (see figure 1) by the users and their organization. Knowledge of the technical inner workings of the hosted application (such as servers, databases, and failover systems) is not a concern of the purchasing organization. For this discussion, we will be discussing the desired features of hosted applications whose main purpose centers on software quality and Application Lifecycle Management.

## The Features and the Value Added

### Appropriate Tool for Application Lifecycle Management

There is an old adage that describes this feature perfectly – “always use the right tool for the job.” This adage holds true in many instances, and here, it continues to be a truthful statement. Although hard statistics regarding the usage of software quality management tools are not readily available, many software development teams are still using common spreadsheet and word processing programs to organize their resources. These resources include (but are not limited to) test cases, requirements, test execution statistics, and defects – and are all housed in formats that are only marginally useful when attempting to improve software quality.

Using spreadsheets or other non-targeted applications for organizing an ALM effort may be considered beneficial at first, but consider the following table (See Table 1) that outlines the pros and cons associated with this method:

Pros
<ul style="list-style-type: none"> <li>• <b>Minimal ramp-up time</b> – Spreadsheet software and text documents are familiar items for the users, requiring little to no time before they can be used for ALM projects.</li> <li>• <b>Familiar application</b> – Users that possess a comfort level with a particular application are known to be more productive. For instance, this minimizes the time spent searching through manuals and online forums answers to problems, allowing that time to be available for the project instead.</li> <li>• <b>Small to no cost</b> (cost absorbed previously) - Since spreadsheets and word processing documents have become ubiquitous items in nearly all organizations, the cost of purchasing these programs will have been absorbed previously, leaving no cash outlay to use them.</li> </ul>
Cons
<ul style="list-style-type: none"> <li>• <b>Lack of process or methodology</b> – Documents alone do not contain any inherent structure or strategy for improving software.</li> <li>• <b>Disorganized</b> – Since spreadsheets and text documents don't have an implicit organizing structure, keeping them in order and useful can be a full-time job.</li> <li>• <b>Limited reporting capabilities</b> – Even when documents and spreadsheets are organized into an impeccable fashion, obtaining useful information from this collection of documents can be close to impossible. Without targeted and informative reports, decision-making for the ALM project can be difficult</li> <li>• <b>No technical support</b> – While word processing programs may not be susceptible to many technical issues, support might not be available when a problem does arise, and this support will definitely not be targeted toward software quality goals the way a specific ALM tool might be.</li> </ul>

Table 1. Pros and Cons of using Spreadsheets and Other Applications for ALM

While there are benefits to using non-targeted ALM solutions, the downsides of using this method are substantial. Minimizing the negative items on this list should be the main goal of companies that provide hosted ALM solutions, and if these three items can be mitigated, the advantages (access to a methodology, organization, reporting, technical support, and more) can be obtained with a minimal downside.

### “Right-sized” Tool Cost

Costs are an important consideration for the stakeholders of a project when a software quality effort is considered. It is safe to say that the budget available for any ALM solution is always a percentage of the total cost of the project. Consequently, when considering a tool, its cost should be directly proportional to the size of the project. Small to mid-size software projects will need to consider lower cost ALM tools because of their limited resources. Project management that can find a hosted ALM application that delivers both high quality and a robust set of features for a low cost can realize greater software quality for a lower cost. The cost-savings benefit for larger software projects that can obtain the same low cost software quality tools can be extremely high.

### Integrated Software Quality Methodology

*“The most recent results for the [mini-survey] on the CSST Technologies, Inc., Web site indicate that 40% (102 out of 258 respondents) see software testing methods/process implementation as doing the most to facilitate their testing work.”<sup>5</sup>*

A proven test methodology can sometimes be the defining difference between implementing a successful software quality improvement project and wasting time, money, or other valuable resources. To say it in a simpler fashion; having a viable plan makes nearly any project easier to implement, more resilient to the unexpected (contingency efforts are usually included in project planning), and increases the chances of success. If a methodology is integrated within the ALM tool, adhering to this methodology becomes significantly easier. Moving away from the methodology during the course of the development process (either in error or for other reasons) becomes more difficult, resulting in



resource savings that would otherwise be lost for the project if project members misinterpreted the methodology or used an unproven process.

## Security

Based on a poll of business technology professionals, the top concern of professionals, with regard to cloud computing, is security (scoring 4.3 out of 5, with five being the highest).<sup>6</sup> Applications that once only existed and functioned from within the confines of the organization that utilized them are now accessed solely via the Internet, leaving professionals that access them with a feeling of insecurity.

Cloud based applications need security to be in place on multiple levels – the application itself must be tested for code-based security vulnerabilities, the hosting environment must have adequate monitoring and firewalls in place, and the data centers must be secured as well. A secure application will allow the project stakeholders to take full advantage of all the benefits of cloud-based applications while mitigating the risk associated with their primary fear of these offerings.

## Support

Cloud computing is a service, and like any other service, requires a competent and aggressive support structure to be in place for it to function properly. Problems that arise in the hosted application need to be addressed by the provider and they need to be addressed quickly. Recovering from issues and addressing customer problems equates to a ROI of time savings for the application user if problems arise. As an added value, some hosted application support will be willing to work closely with subscribing users to identify areas that can be customized for the user or to help resolve defects that have made their way into the hosted application.

## Roles

Proven business management strategies such as Six Sigma have well-defined roles at their core. Including well-planned and well-defined roles such as these within a cloud-based ALM application can provide benefits similar to those available from a proven methodology. Roles keep team members focused on their assigned tasks and reduce wasted effort that is the product of overlapping responsibilities. Role-based access to resources also prevents accidental and unauthorized actions from occurring outside the individual user's area of responsibility. A simple example would be a business analyst versus the developer; the business analyst gathers and authors requirements for the application and the developer writes build notes based on her work. Neither role should be able to modify the work of the other resource.

## Data Integrity and Application Availability

Data integrity in this instance means that the customer's data remains available and intact when it is stored in the cloud. Application availability translates to a simple idea; whenever and wherever the application is needed, it is available. Some measures to look for in a hosted ALM solution that equate to data integrity and application availability would be

- **Managed Database Backups** – An organized process that backs up and protects subscriber information housed within the hosted application's database
- **Hardware Redundancy** – Cloud offerings should provide backup systems for all pertinent hardware, minimizing the loss of productivity incurred by the subscriber if a hard disk (or another piece of hardware) breaks or is found to be defective.
- **Multiple Hosting Sites** – Several independently powered locations (separate power grids) for the hosted application to be offered from as a secondary back-up in case a disaster should arise and prevent the primary hosting location from functioning.
- **Service Level Agreements (SLAs)** – A contract between the provider and the user that specifies the level of service expected during its term. SLAs are used by vendors and customers as well as internally by IT shops and their end users. They can specify bandwidth availability, response times for routine and ad hoc queries,



response time for problem resolution (network down, machine failure, etc.) as well as attitudes and consideration of the technical staff.<sup>7</sup>

If features such as these are available to the cloud-user, they can help prevent data loss and foster time savings if recovery of data is ever a necessity. Organizations that provide SaaS offerings and have these features available will be able to quickly recover services and safeguard data when disaster strikes.

### **Traceability, Reporting, and Audit Trails**

All software quality artifacts such as test cases, requirements, and build notes are certain to change over the course of a project. As changes are introduced to these items, a record of where they once were and where they exist now serves a necessary purpose for the user. Having traceability available within the hosted ALM application means that changes to software quality assets can be clearly seen and tracked within the hosted management application. This can help prevent old issues and problems formerly found in the software quality effort from re-occurring. Project managers can prevent history from repeating itself and resources will not be wasted attempting to perform a task that was unsuccessful in the past over again.

Cloud applications that provide reporting functionality allow project managers, team leaders, and stakeholders to gain a better understanding of the progress of the software quality project as it moves forward. Reports can provide this understanding by furnishing the reader with clear, concise, “real-time” information about the project so that informed decisions can be made. Reporting functionality should be available to application users in the form of an easy to use interface that allows information in the generated report to be filtered and organized. This filtering is necessary so that a report can be targeted for the current task or decision at hand.

If traceability and reporting are available as features for the software development project, changes in ALM assets housed within the hosted application can also be more easily scrutinized when necessary. This creates an audit trail that can be examined by regulatory personnel, which will help to shorten audit times and helping to adhere to regulatory rules.

### **Minimal Ramp-up Time**

As mentioned previously in the section that discussed using appropriate tools for the software quality improvement process, training the users to work with specialized ALM software in a hosted environment was identified as a problem. Tools that require instructional classes or lengthy time commitments for ramp-up are not beneficial to the software quality effort. A tool that allows the user to begin using it for project-oriented tasks in a short span of time, with a minimal training commitment, provides value in the form of additional hours devoted to project tasks instead of training. Hosted ALM systems that promote self-training or training via short tutorials mean less money spent getting the users up to speed with the tool and more time ensuring that a quality product is being delivered to the user.

### **Integration with External Tools**

Cloud based tools that provide pathways for data to flow easily in and out of the hosted ALM application mean several bonuses for the user. Primarily, the key value is in the time savings from converting existing software quality documents created prior to moving into the cloud to generate artifacts within the cloud-application. Spreadsheets could be fed to the new hosted application and information within the documents can be used to create test cases or build notes. Integration also means that time and money will be saved if moving out of the current cloud-based software quality system will fulfill the needs of the business at a later date.

## **Conclusion**

Application Lifecycle Management applications, like many other enterprise applications, are making their way into the realm of cloud computing. Identifying the benefits and risk of individual hosted offerings can be complicated and difficult, even as these benefits continue to grow with time. By considering some, if not all of the features described here when selecting a software quality management tool that is based in the cloud, project stakeholders can see a ROI from project time savings, improved use of resources, elimination of redundancy, and increased information from the software quality effort.



## **About the author**

**Steven Hamilton** has spent several years as a Test Engineer with RTTS (Real-Time Technology Solutions,) a software quality company catering to Fortune 500 firms. He has worked on software quality engagements taking part in all areas of the application lifecycle from early planning and requirements gathering, through test execution, and defect analysis/remediation.

Steven also works as a Software Architect with TOMOS Software ([www.tomos360.com](http://www.tomos360.com)), helping to create modern, intuitive, and agile software quality solutions for software development projects of all sizes. TOMOS is a hosted offering (Software as a Service) that provides tools for the improvement of all levels of the Application Lifecycle Management process without many of the negatives usually associated with these tools.

Previously, Steven spent three years as the principle of an IT service company attending to the needs of both residential and business users in the areas of networking, telephony, and computer hardware.

Steven holds a Bachelor of Science degree in Computer and Information Science from Brooklyn College. He can be reached at [steveh@reachsimplicity.com](mailto:steveh@reachsimplicity.com)

---

<sup>1</sup> IBM (August 1, 2008). *IBM Expands its Cloud Computing* [Press Release]. Retrieved from <http://www.ibm.com/news/us/en/2008/08/01/z194896g63799p60.html> on 31-Jan-09

<sup>2</sup> Kim Querner and Michael Lundblad (2007). *Software Quality Management from IBM: A Closed-loop Fitness Program* [whitepaper]. Retrieved from <http://www-01.ibm.com/software/rational/offerings/quality/> on 31-Jan-09

<sup>3</sup> National Institute of Standards and Technology (NIST), 2002. *The Economic Impacts of Inadequate Infrastructure for Software Testing* [whitepaper]. Retrieved from <http://www.nist.gov/director/prog-ofc/report02-3.pdf> on 31-Jan-09

<sup>4</sup> Galen Gruman, InfoWorld (2008). "What Cloud Computing Really Means". Retrieved from [http://www.infoworld.com/article/08/04/07/15FE-cloud-computing-reality\\_1.html](http://www.infoworld.com/article/08/04/07/15FE-cloud-computing-reality_1.html) on 31-Jan-09

<sup>5</sup> Daniel J. Mosley and Bruce A. Posey, *Just Enough Software Test Automation* (New Jersey: Prentice Hall, 2002), 4.

<sup>6</sup> Michael Biddick, InformationWeek (2008). "Time to Think About Cloud Computing". Retrieved from <http://www.informationweek.com/news/software/hosted/showArticle.jhtml?articleID=211300562> on 1-Feb-09

<sup>7</sup> PCMag Encyclopedia (2009). Retrieved from [http://www.pcmag.com/encyclopedia\\_term/0,2542,t=SLA&i=51448,00.asp](http://www.pcmag.com/encyclopedia_term/0,2542,t=SLA&i=51448,00.asp) on 25-Mar-09