Statement of Work for
Application Lifecycle Management (ALM) Tool

1. Purpose

The purpose of this Request for Proposal (RFP) is to invite prospective vendors to submit a proposal to supply an Application Lifecycle Management (ALM) system for the Washington Metropolitan Area Transit Authority (WMATA) Department of Information Technology (IT). The RFP provides vendors with the relevant technical, performance, application, and architectural requirements of the system.

2. Background

2.1 WMATA

The Washington Metropolitan Area Transit Authority (WMATA or Metro) is the transit provider for the National Capital Region, providing safe, clean and reliable service to both residents and visitors. Our customers include more than a third of the federal government workforce and millions of tourists who visit the Nation's Capital every year. Metro operates the second largest heavy rail transit system, sixth largest bus network and fifth largest paratransit service in the United States.

Metro was created by an interstate compact in 1967 to plan, develop, build, finance, and operate a regional transportation system in the national capital area. Metro began building its rail system in 1969, acquired four regional bus systems in 1973, and began operating the first phase of Metrorail in 1976. The newest leg of the rail network opened on July 26, 2014. Today, there are 91 Metro stations in service within a 118 mile network. Metrobus serves the nation's capital 24 hours a day, seven days a week. Metrorail and Metrobus serve a population of 5 million within a 1,500-square mile jurisdiction. Metro began its paratransit service, MetroAccess, in 1994; it provides about 2.3 million trips per year.

To support customer operations, WMATA maintains more than 1,100 rail cars and over 1,500 buses. The rail right of way and related equipment, the system's 91 stations, the 24x7 electrical power grids, and other infrastructure elements, from plumbing to tunnels to bridges, are maintained by WMATA employees representing a range of crafts and working out of rail yards and bus garages located across the service area. WMATA's total workforce of approximately 11,000 is largely unionized.

WMATA is set up with a board of directors that appoints a General Manager as CEO to supervise the day-to-day operation. The following figure presents the organization chart of Metro.

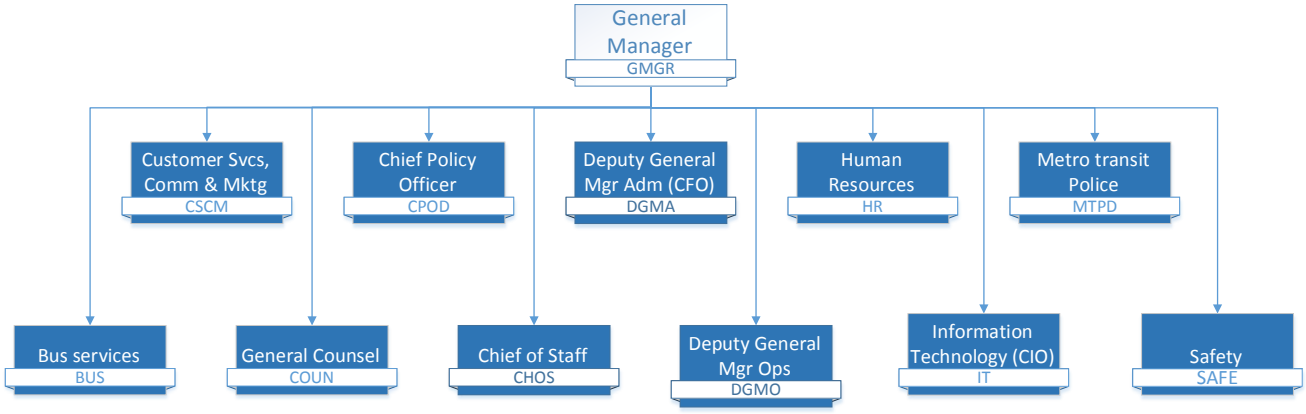

Figure 1: WMATA Organization Chart

2.2 Department of Information Technology

This Request for Proposal (RFP) is issued by WMATA's Department of Information Technology (IT). IT provides information technology and telecommunication services to support WMATA's strategic goals including safety, customer service, transparency and decision support. IT implements solutions throughout WMATA; promotes compatibility, integration and interoperability; and develops and enforces information technology policy and standards throughout the Authority. The scope of IT's responsibility extends to the infrastructure that includes such elements as voice and data communications, the enterprise storage area network (SAN), transport backbone connectivity, and the operations to support these items.

The IT department is led by WMATA's Chief Information officer (CIO), who is supported by the following offices:

- Application Development & Operations (APPS): responsible for developing, configuring, deploying and maintaining major software systems used throughout the Authority. The office also provides technical support to users of those applications.
- Business Process Re-engineering (BPR): responsible for align information technology solutions with operational needs.
- Data Center & Infrastructure (DCI): responsible for designing, building, maintaining, and monitoring a robust, state-of-the-art IT infrastructure to enable Metro to deliver safe, reliable, and effective transit operations.
- Enterprise Architecture (EA): responsible for the blueprint and roadmap for aligning business strategy with IT by identifying best practices, standards, tools, processes, and templates to assist in the creation of WMATA's enterprise architecture.
- Program Management Office (PMO): responsible for supporting the financial management and procurement processes of the IT department.
- Metro IT Security (MITS): responsible for ensuring a safe, secure, and resilient digital environment, as well as cost effective Cyber Security products & services.
- Networks & Communications (NCS): responsible for maintenance of all Voice and Data communications used throughout the Authority. All Voice and Data including infrastructure

and cabling falls under this group, including the network connectivity, telephone services, copper and fiber optic cabling and cellular services.

- Quality Assurance (QA): responsible for ensuring quality in IT department's operations, such as the establishment and implementation of quality policy, objectives, responsibilities, to include technical support for configuration management, software requirements, management/testing, documentation, training, current and future IT and business process documentation, and audit responsibilities.

One of IT's key roles is the design, development, integration, testing, deployment and support of the 100+ applications and services that are used for day-to-day WMATA operations, many of them mission critical, such as those in support of customer service, financial management, inventory control, telecommunications provisioning, bus and rail scheduling, and data warehousing. Some examples of these systems are: Maximo for network and communications work order and asset management, NextFare, an automated fare collection system, PeopleSoft for finance and human capital management, Trapeze, a proprietary transportation scheduling and work management application, Advanced Information Management (AIM), a proprietary train operations and monitoring system, and Safety Measurement System (SMS) for safety case management that is built in-house.

These systems are sourced from a mixture of wholly in-house development, in-house integration and configuration of vendors' packages (such as Oracle's PeopleSoft, IBM's Maximo, and Trapeze transportation solutions), and wholly COTS solutions (such as Amano McGann Parking System). The following lists some of the pertinent characteristics of WMATA's IT environment:

- AIX and Windows based development and productions environments;
- Citrix Web-based infrastructure for serving client-server applications from anywhere across the internet;
- Microsoft SQL Server Database Management System (DBMS) used by mission-critical applications and services;
- Cloud and virtualization capabilities using VMware;
- Hardware consisting of IBM pSeries and Dell PowerEdge servers.

2.3 IT Strategic Plan

The IT Strategic Plan is part of an ongoing process for aligning the IT project portfolio with WMATA's strategy, line of business plans and external drivers. As such, the strategic plan provides the baseline against which IT manages its project portfolio to ensure that all projects are consistent with the department's goals and objectives that are also aligned with Metro's overall business strategy. The scope of the IT strategic plan is from FY15 through FY17. The strategic plan identifies the following IT goals that are pertinent to this RFP: IT must provide services, applications, and systems that

- Support configuration management;
- Support timely decisions;
- Support effective/responsive continuity of service planning;
- Implement effective customer relationship management (CRM) / analytics;

- Make use of current, cost effective technologies;
- Make effective use of existing resources; and
- Are extensible and adaptable.

2.4 Problem Statement

One of the obstacles to achieving the above strategic goals is that there is no standardized tool for application development and support across all the functional areas of life-cycle management. Currently, different types of ALM tools (or partial components of an ALM) are used by the many IT project teams for different purposes. The following table provides a sample:

Table 1: Sample of Tools Used By WMATA Projects.

Tool	Req Mgmt	Defect Mgmt	Source Code Mgmt	Build Mgmt	Release Mgmt	Test Case Mgmt	Test Execution Mgmt	Load/ Perf Testing	Functional /Unit Testing	Project/ Portfolio Mgmt
Bugzilla		X								
FogBugz		X								
DELL STAT			X							
Quality Center	X	X				X	X			
Word	X									
Excel									X	
Visio	X									
balsamiq	X									
Atlassian GreenHopper	X									
Atlassian JIRA	X	X			X					

Tool	Req Mgmt	Defect Mgmt	Source Code Mgmt	Build Mgmt	Release Mgmt	Test Case Mgmt	Test Execution Mgmt	Load/ Perf Testing	Functional /Unit Testing	Project/ Portfolio Mgmt
TestRail		X				X	X			
VAULT			X	X						
SUBVERSION			X	X	X					
TortoiseSVN			X							
Microsoft Visual Studio Team Foundation Server			X	X						
Atlassian Bamboo				X	X		X			
ORACLE PeopleTools				X	X					
hp Loadrunner								X		
Se									X	
specflow Cucumber for .NET									X	
N unit									X	
R# ReSharper									X	
ca Clarity PPM technologies										X

Based on the above table, two issues are apparent:

1. Multiple tools are currently being used for the same functionality. For example, six different tools are used for requirements management; five tools are used for defect management; and five tools are used for source code management.
2. Tools are used to manage portions of the lifecycle but not all of it. For example: Although HP tools are being used to capture defects and handle testing, they do not appear to be used for linking to related requirements.

3. Scope of Work

3.1 General Objectives

With this RFP, WMATA seeks to procure an ALM tool that all the IT project teams can use for development and maintenance of their software products for their complete life cycle. The IT Quality Assurance (QA) Office has proposed that the ALM tool in concert with standardized processes serve as a common framework specifically for supporting the software lifecycle of WMATA IT projects.

The selected ALM tool suite will also serve as WMATA IT's integrated application management system for requirements, defects and testing. It will feature a built-in change control board for approval tracking and configurable metrics dashboards for management at the project, release, baseline, folder and user level. The framework will be used for managing requirements, quality, document and software changes, as well as configuration management activities for all projects across the Authority.

The selected ALM tool suite shall be based on a robust architecture that provides "out-of-the-box" features and functions for requirements, change, configuration, quality management, and the software lifecycle. Although we expect that industry's best practices and methodology templates will be built into framework, we also want the ALM tool to be configurable and customizable.

In summary, the procurement of an ALM tool suite to be used as a standard life-cycle management framework for WMATA projects is part of the overall IT strategy plan. Implementing a standard ALM framework offers WMATA IT the opportunity to:

- Improve the enterprise-wide use of common toolset for development and support;
- Enhance developer productivity;
- Increase the efficiency of requirements management practices;
- Increase the efficiency of quality management practices;
- Increase the efficiency of change management practices;
- Increase the efficiency of configuration management practices;
- Get complete traceability starting from requirements to test cases;
- Report to management on standard attributes for all the projects across the Authority;
- Improve the overall quality of projects; and
- Increase the rate of successful project completion.

To realize these opportunities, we are in search for an ALM tool suite that can help us with these four main objectives of the IT Department:

1. Real-time planning: to capture and manage a work plan identifying what goals have been scheduled, who's working on these goals, which goals are done or in-progress. The plan is integrated with the execution of tasks for accurate tracking for all disciplines in the life cycle from multiple perspectives.

2. Governance: to ensure that the developed application suits WMATA's needs, and that updates and larger revisions for deployed applications make business sense;
3. Life-cycle traceability: to track project artifacts and their relationship through the life cycle of an application product, from requirements to tests that become designs that engender software products that are packaged into releases that become deployments.
4. Real-time measurement: to identify metrics that measure and report progress and success as the planned tasks are executed though the life cycle of a product.

3.2 FY15-16 Objectives

The long-term objective is that the implementation of an enterprise-wide ALM framework that will be used across all IT projects to align IT's business processes with the evolution of WMATA's performance measurement program. However, the primary objectives for FY15-16 are as follows:

- Implementing a standard ALM framework with minimal risk to projects that are in progress or near completion. Given the disruptive nature of introducing a new toolset for a project in flight, we have decided that the selected ALM framework will be used primarily for a handful of carefully chosen new projects that shall serve as pilots;
- The new projects selected as pilots for using a standard ALM framework should reflect the prevalent development methodologies used here at WMATA: waterfall, agile, spiral, rapid application development, and prototyping;
- The ALM tool shall be used by the pilot projects in a way that exercises all phases and activities of the life-cycle associated with their chosen development methodology;
- WMATA is in the process of developing a Service Oriented Architecture (SOA) approach to its integration needs. Consequently, the pilot projects are expected to use the selected ALM framework for applying SOA principles to realize cloud enabled solutions, holistic business insight and agility to externalized APIs;
- Using results from the pilot projects, WMATA's IT department shall demonstrate the efficiency gains in time and cost for meeting the needs of its customers;
- Mid to long term goal will be to eliminate the multiple tools currently being used, standardize all projects to use one tool to reduce cost, increase consistency, accuracy and establish one repeatable, measurable process.

The successful Offeror will offer an ALM solution that provides the following:

- Good coverage for the technical requirements specified in Appendix I of this RFP;
- Ease of use/productivity features for all the roles associated with software development: requirements analysts, system engineers, architects, developers, configuration managers, testers, and project managers;
- Conformance to WMATA's security policies;
- Excellent customer service in terms of technical support, documentation, training, and consulting;
- Strategic vision to address potential future needs as well as current objectives.

3.3 Number of Users

WMATA anticipates that as many as 50 (fifty) users will use the ALM tool during the first two years of its deployment. Offerors should assume that there may be up to 45 (forty-five) concurrent users.

3.4 ALM Installation Plan

Initial deployment consists of two environments: development and production, each environment consisting of an ALM application server (Windows), ALM database server, and ALM build and deploy server. Future plans include deploying backup server(s).

3.5 Customer Service and Knowledge Transfer

WMATA requires technical telephone help desk support during normal business hours Monday through Saturday, 7 AM to 6 PM EST. Offerors are also asked to include an option for after-hours support.

WMATA also requires comprehensive training for up to 50 (fifty) users for these software development roles: requirements analyst, system architect, developer, configuration manager, ALM system administrator, integration and testing, and project manager. Offerors are asked to provide specifics of the recommended curriculum for the various roles involved in use of the proposed product, as well as a generic course schedule.

3.6 Location

The ALM software will be installed in the Data Center at WMATA's Jackson Graham Building (JGB) in Washington, DC. The address of the location is:

Washington Metropolitan Area Transit Authority
Jackson Graham Building
600 5th Street, NW
Washington DC 20001

4. Submittal Requirements

4.1 Technical Requirements

Technical requirements are specified in Appendix I and are presented in template format for response. Offerors will respond using the template by checking the appropriate box for each requirement item (Yes, Full Support out of the box; Qualified No, partial support; No support). Offerors may provide explanations and comments, and may submit additional information if they desire, but the template must be completed as indicated.

At a minimum, the ALM solution must provide the following features:

- Develop requirements and user-stories, then decompose into detailed tasks for resource tracking;
- Prioritize and schedule requirements for each release/iteration;
- Assign requirements, tasks and incidents using Agile planning board;

- Provide for full-feature source code repository;
- Integrate with project team's integrated development environment (IDE) of choice, such as Eclipse, Xcode, Visual Studio, and other IDEs;
- Support automated software builds and deployment for continuous integration (CI) and testing (integration with leading CI solutions, such as Jenkins/Hudson) ;
- Track bugs, enhancements, risks and issues linked to changes in source code repository;
- Create, edit and execute test-cases;
- Map tests to requirements to track coverage;
- Create and assign test sets for test execution management;
- Record bugs linked to test steps for full test traceability;
- Manage manual and automated testing;
- Provide personalized dashboards and customizable reporting;
- Provide capability to attach documents, screenshots and URLs to all artifacts;
- Maintain a full audit history of all changes made in the system
- Provide customizable graphs and reports in multiple formats (HTML, MS-Office and Adobe Acrobat);
- Support multiple browser environments including Chrome, Safari, Internet Explorer (IE) or Firefox running on AIX and Microsoft platforms.

4.2 Customer Service and Knowledge Transfer

Offerors will provide information about Customer Service and Knowledge Transfer as follows. Offerors will use the template for Customer Service Evaluation that is provided in Appendix II. Offerors may submit additional information if they desire, but the template must be completed as indicated.

Section Number	Title	Details
4.2.1	Technical Support	• The Offeror must be able to resolve or help WMATA resolve any issues encountered during the implementation and usage of the ALM solution. The Offeror shall at least be able to direct our issues/concerns to the right support team as soon as possible depending upon the severity of the issue raised; • The Offeror must define and describe the service details of the various priority levels of technical support provided; • The Offeror must specify the hours and days of week available, number of qualified technical support resources available, and method of delivery (e.g., telephone, real-time on-line, email); • The Offeror must state whether it offers a dedicated customer point of contact, and; • The Offeror must state whether it is willing to enter into a Service Level Agreement (SLA) for technical support.

Section Number	Title	Details

4.2.2	Documentation	• A comprehensive listing of the documentation topics covered to include, installation, setup and configuration instructions of the
		  o ALM tool suite,
		  o associated databases, if applicable,
		  o webserver application, if applicable,
		  o lightweight directory access protocol (LDAP) server application, if applicable, and
		  o other applications for a complete working ALM solution that enables WMATA users at different locations to share and collaborate on IT projects over the network;
		• Method, frequency and mode of documentation updates.

4.2.3	Knowledge Transfer	• Training availability (offerings and locations convenient for WMATA personnel), • Modes of delivery (e.g., learning center, on-site, web-based, train the trainer); • Ongoing, expert mentoring from a technical account manager, helping WMATA ensure successful ALM solution adoption in alignment with business objectives; and • Comprehensive offerings with respect to a range of users, skills and skill levels required to use the proposed ALM solution successfully.
4.2.4	Consulting Qualification	Offerors' expertise and experience, including bench strength and availability of qualified consulting resources competent to assist with tool implementation and deployment, and to provide guidance and support to projects in meeting the following objectives: • To accelerate successful deployment of ALM solution with pre-packaged implementation and migration best practices; • To ensure that WMATA successfully adopts the ALM solution in alignment with business objectives described in this RFP; • To help define the success criteria for the deployment and adoption of the ALM solution, to include a strategy for measuring progress; and • To optimize the cost of the ALM investment by improving cost predictability of WMATA's IT projects.

4.4 Strategic Direction

Offerors will provide information about their Strategic Direction by submitting an overview document of not more than 15 pages that addresses the following questions regarding their product, its architecture, future direction, and implementation approach. If possible, vendors are requested to reference published material available on the web, in company documents, or in the proceedings of industry meeting.

Section Number	Title	Details
4.4.1	Purchase Model	Will the ALM tool suite delivered to WMATA be licensed based or will it be provisioned via software as a service (SaaS) model? Be sure to explain the licensing and/or SaaS model by providing details about what value-added features/services it includes. If both options are available from the Offeror, what does the Offeror recommend in light of WMATA's business objectives described in this RFP? What are the advantages and disadvantages of license-based versus SaaS?
4.4.2	Out-Year Options	Assuming WMATA has purchased your proposed ALM tool solution, what options do you recommend in the way of maintenance, support, products and services to ensure that the solution continues to be effective after the base year? Please provide options /recommendations for three years.
4.4.3	Key Components and Features	What components are included in the base product, and what are the core features of the product?
4.4.4	Additional Components and Features	What are the additional components, their respective features, and how are they packaged? Are they available now or in the future?
4.4.5	Components-Lifecycle Phase Matrix	What are the components used for the various lifecycle phases of an application? For example, what are the components used for managing Requirements/User Stories, Design, Implementation (development), Verification (integration and testing), Release, and Maintenance?
4.4.6	Recommendations for Quick Start	Given a new development project utilizing Agile with a team of fifteen (15) members consisting of: software architect, engineers, developers, configuration manager, testers, deployment, and project managers, what would Offeror recommend to quickly get started with the proposed ALM system? How should the ALM system be configured? What components to use? What processes involving the ALM solution should be established? Be sure to include other pertinent information that can help the WMATA project team quickly get started with the ALM system.

Section Number	Title	Details
4.4.7	Product Strengths	What are your ALM solution's strongest capabilities?
4.4.8	Product Differentiation	How does your ALM solution differentiate from the other products in the market?
4.4.9	Project Efficiency	Describe how your ALM solution provides WMATA IT Department with ways to improve organizational efficiency and reduce the total cost of ownership (TCO).
4.4.10	Installation	Describe the installation, setup, and configuration requirements and the usual installation time from start to active use.
4.4.11	Post-installation	Describe any post-installation tasks and other activities to ensure that the installed ALM solution is optimized.
4.4.12	Product Support	Does the proposed ALM solution currently support the following? If not, what plans does the Offeror have to provide such support in future: • 4.4.11.1: Service-oriented architecture (SOA); • 4.4.11.2: Cloud-based services, such as testing; • 4.4.11.3: Mobile capabilities; • 4.4.11.4: Continuous integration; • 4.4.11.5: Network visualization application testing (to simulate the end-to-end effects on latency and response times of accessing data over WMATA's wireless and WAN networks); • 4.4.11.6: Data analytics (application development data available vertically for management hierarchy and horizontally across project teams for better-informed decisions and transparency).
4.4.13	Product Adoption Success Criteria	What are the Offeror's recommendations for defining and measuring WMATA's progress for adopting the ALM solution and for tracking success in terms of process, financial, and other perspectives?

4.3 Product Demonstration

WMATA reserves the right to require product demonstrations and/or presentations, which may take place at the Jackson Graham Building, Washington DC, at the Carmen Turner Facility in Landover MD, or at both locations. WMATA may specify the content and format of such demonstrations or presentations as it sees fit.

5. Evaluation Criteria

Technical sufficiency will be evaluated according to the following factors:

Section Number	Title	Relative Order of Importance	Details
5.1	Technical Requirements	1	The capability of the ALM product proposed to satisfy the technical requirements specified in Appendix I, as indicated by the Offeror on the Technical Requirements Template provided in Appendix I.
5.2	Customer Service and Knowledge Transfer	2	The capability of the Offeror to provide consulting services, help desk support and training for all proposed tools associated with the ALM solution. Offerors will use the Customer Service and Knowledge Transfer Template provided in Appendix II to provide the information requested in Submittal Requirements Section 4.2 above.
5.3	Strategic Direction	3	The future vision and implementation approach for the proposed ALM product(s) as described in the vendor's response to Section 4.4 above

Appendix I:

Technical Requirements Response Template

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
1. GENERAL REQUIREMENTS				
A. Deployment Options: Support ALM implementation for				
AIX (web interface and non-web interface)				
Windows (web interface and non-web interface/command line)				
Virtual and physical server environments				
Initial deployment consists of two environments: development and production, each environment consisting of ALM application server (Windows), ALM database server, and ALM build and deploy server. Future plans include deploying backup server(s)				
50 total users				
Up to 45 concurrent users				
Database platform support for SQL Server, Oracle, MS Access				
B. Productivity/Integration Features				
Support of mainstream browsers: Internet Explorer, Chrome, Firefox, Opera				
Support deployment and execution in a virtualized environment				
Provide for centralized and/or distributed repository for storing ALM artifacts, such as requirements, defects, source code/development artifact, etc.				
Support comprehensive search capability to find any artifact related to application lifecycle, such as software components, objects, requirements, user cases, user stories, defect tickets, and all the other ALM artifacts				
Support point/click/drag and drop navigation for creating, modifying, deleting for all ALM artifacts				
Support point/click/drag and drop navigating for organizing, traceability and linking ALM artifacts (e.g., for mapping requirements into use cases to create a link between the two; for mapping software products to a release baseline, etc.)				
Support point/click/drag and drop to navigate artifact hierarchy, such as requirements, software builds and releases, test cases, defects, etc. by scrolling vertically and horizontally across branches				
Represent ALM artifact graphically				
Provide capability to add documentation notes to the graphical depiction of ALM artifact				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Provide a graphical scheduler to define schedules and tasking/work dependencies across all life-cycle phases of the application development project				
Provide workflow management for the project from concept → requirements → implementation → verification → release → support				
Support collaboration tools, such as email notification, instant messaging, and other technologies., for project participants to communicate handoff of work from one work flow process to another				
Set up workflow for a project using pre-defined template(s) that support the various development methodologies: agile, feature driven development (FDD), rapid prototyping, waterfall, spiral, or some hybrid as defined by the user				
Perform configuration management by automatically detecting version changes in ALM artifact; for example, automatically versioning and tracking changes in requirements, software code and executables/libraries, and release versions				
Provide visual cues to display status of tasks				
Provide the ability to copy, cut and paste objects and sessions into ALM artifact				
Provide for automated notification of project participants based on standard or user-specified events (e.g., when a requirement has changed; when software build is completed; when source code is checked in; etc.)				
Provide for automated notification of project participants when they are required to perform a task for project to progress (e.g., approvers to sign off requirements; configuration management to kick off building a baseline; testers to execute tests against a new software build; etc.)				
Support metrics reporting (e.g., progress/status, requirements, development, testing velocity, source code, test changes, tracing metrics, etc.) with charting and export capabilities				
Provide for rollback functionality to revert an ALM artifact (requirement, software build, test, etc.) to its previous saved version				
Support baselining of ALM artifacts (e.g., baselining a set of requirements for initial release; baselining test cases; baselining software builds for release; etc.)				
Support rollback/restoration of a previous baseline				
Provide for capability to compare/differentiate between multiple versions of a baseline or branch				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support capability to produce dashboard reports using existing templates provided by ALM tool or user-defined templates				
Support capturing mathematical equations of varying complexity and linking them to one or more ALM artifacts				
Support capturing pseudo code of varying complexity and linking them to one or more ALM artifacts				
Support integration with SharePoint so that users can create a link between ALM artifact and a document located in a SharePoint repository				
Support integration with other continuous integration (CI) tools such as Jenkins/Hudson				
Provide for capturing resources that can consist not only of computers/servers (e.g., CM server, development servers, build servers, servers used for testing a release build, etc.), but also users and user groups as well as other discovered resources associated with a project				
Provide the means to organize, plan, schedule, and allocate resources (user, user groups, and servers) into easily manageable units, enabling users to create an organized structure that logically represents the kinds of tasks performed for a project.				
Provide the ability to automatically discover relationships between ALM artifacts (e.g., parent –child relationship for requirements, tree-branch relationship for software baselines, etc.)				
Support scripting or programming for complex conditional internal logic to automate ALM tasks, such as software builds, deploying software, etc.				
Provide advanced users the capability to execute SQL queries to generate reports related to any aspect of the ALM database				
Provide the features and tools for collaboration within a project team in which team members are not co-located				
Provide the features and tools for collaboration between multiple project teams that are not co-located, perhaps using a different development methodology and hardware platform				
Provide for capability to view the revision/update history (date, comments, etc.) for any ALM artifact (e.g., view the revision history of a requirement or the change history of a source code file)				
C. Ownership & Security				
Provide a GUI for security administration				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support individual, group, and role-based security				
Support customization of roles				
Support grouping of users and their access by projects or other user-specified criteria				
Provide storage and transfer of passwords in encrypted format, with passwords decryptable only by administrator or owner.				
Provide the ability to audit security settings, events and changes with configurable reports, as well as from the GUI interface				
Provide the ability to limit connectivity to authorized servers only				
Support user authentication using an authorized Identify and Access Management (IAM) system (specify if using LDAP or something else for this requirement)				
Provide the ability to segregate access privileges for System Administration from Security Administration				
Provide system-level password controls				
Provide the ability to send and receive encrypted messages				
Support failover and disaster recovery architecture				
Provide the ability to group mappings into process flows				
Provide the ability to model job schedules and attach schedules to executable objects				
Permit the designer of a schema or mapping to specify the optimal parallel degree for querying and for loading				
Automate the processes for loading a table/querying a table from the schema/mapping				
D. Ease of Use				
Easy-to-use and intuitive ALM tool to use; that it does not require extensive customization				
Provide capability to automatically create an ALM artifact from template or by cloning an existing artifact				
Provide workflow support for project tasks: for example, ability to assign tasks and deadlines, track tasks, send notifications, escalate assignments; etc.				
Provide end-user dashboard features for tracking projects, assignments and status				
Provide the ability to integrate business process and other related documents into the requirements, design and testing portals				
Provide the ability to associate and link documents with specific ALM tasks and objects				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Provide built-in design and monitoring GUI interfaces for all phases of the application development lifecycle				
Support customizing on-line help for ALM tasks and tools				
Support cut and paste from interface to Windows clipboard				
Provide standard and custom reporting capability related to project standards, tasks, and performance				
Provide ability to work off-line (i.e., no connection to ALM repository for requirement artifacts, source code, or tests) and merge artifacts updates to appropriate repository upon reconnecting				
Provide for context-sensitive help within the ALM system				
Provide for learning aids (e.g., sample requirements, workflow tutorials, etc.)				
Support linking external artifacts, documents, wiki with ALM artifacts (e.g., linking notes from gathering sessions to individual requirements or linking products from another tool to a build record in ALM)				
Provide for capability to make ALM artifact modifications in real-time (e.g. on a projector in a meeting make live edits of requirements)				
Support viewing by searching/filtering/sorting of ALM artifacts (e.g., requirements, defects, etc.) by user-selected criteria that can include multiple fields/parameters				
Support the capability to save user-specified ad-hoc searching/filtering/sorting criteria (analyst creates custom filtered view that will be used in the future)				
Support searching ALM artifacts (e.g., requirements, defects, etc.) by keywords				
Support grouping of ALM artifacts (requirements, defects, etc.) by project with the option of setting up a system that includes multiple projects				
Support use of pre-defined templates (including workflow templates) for managing ALM artifacts, such as requirements, defects, source code, test cases, etc. for new projects to minimize setup				
Support correction of ALM artifacts, such as requirements, defects, test scripts, etc. for spelling and grammatical errors				
Support customized views of ALM artifact traceability data in a table where user can select the objects that show up in the columns (e.g., user wants to identify all the software builds and test scripts associated with a set of requirements)				
Support the automated export of ALM artifacts, such as requirements, defects, test scripts, etc., in variety of formats: Word, Excel. RTF, HTML, XML, etc.				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Provide features that can be executed via mobile applications. Improve the way software professionals plan, code, test, deliver and maintain applications using smartphones and tablet computers				
Support documenting a requirement, defect, test case, and other ALM artifacts with rich text formatting				
Support describing an ALM artifact, such as a requirement, defect, test case with an image directly in the database (e.g., a bitmap context diagram or Visio file image or screen capture)				
2. REQUIREMENTS MANAGEMENT REQUIREMENTS				
Support capability to add new requirements				
Support the definition and capture of different types of requirements using different types of methodologies (e.g., traditional requirements [functional, system, performance, etc.], use cases, user stories)				
Support capability for user to define metadata to capture each type of requirement (i.e., create custom data fields to specify, for example, source of requirement, status, etc.)				
Support automatic creation of unique ID for each requirement with the option of using a user-defined template				
Support capability to edit requirements with an undo functionality				
Support documenting a defect with rich text formatting				
Support capability to delete requirements and related artifacts—with an undo functionality				
Support capability to link requirements to business process(es)				
Support to link requirements to test case/scripts and other ALM artifacts that trace how a requirement is captured, implemented, verified, and deployed				
Support organizing requirements by user-specified hierarchy/groupings with sub-grouping capability				
Support describing a requirement with an image directly in the database (e.g., a bitmap context diagram or Visio file image)				
Support describing a requirement with an embedded document in the database (e.g., a Visio diagram)				
Support linking requirements to actual documents in a SharePoint location (i.e., clicking on link opens the document)				
Support linking requirements of the same type				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support automatically identifying requirements from external text documents by key words, structure, etc. (e.g. specify what keywords to search for in the doc and import based on those)				
Support importing requirements from external documents such as Excel, Word, text delimited file, etc. or other requirements management tools				
Support use case modeling for enabling a user to visually model requirements that automatically translates them into text requirements				
Support defining hierarchical relationships between requirements				
Support defining dependencies between requirements separate from hierarchy of requirements				
Support automatic generation of diagrams (such as context diagrams, uses case diagrams, etc.) from written requirements				
Support linking requirements to sub-objects in models (e.g., linking requirements to a step of a process flow, to a system in a context diagram, to a state object in a state diagram)				
Support traceability analysis to identify missing links within the requirements (e.g., functional requirement orphans not linked to a user case)				
Define relationship between requirements (e.g., “traces from”, “dependency”, etc.)				
Define cardinality (bi-directional linking) between requirements (e.g., data modeling with respect to one-to-many, or many-to-one)				
Support automated error checking for problems with requirements links (e.g., parent requirement is invalid so children are marked as invalid; orphaned requirements; duplicate requirements; etc.)				
Support visual representation of a requirement’s traceability results (e.g., displaying a matrix table that maps functional requirements to use cases)				
Support the preview and automated generation of requirements documentation in various pre-defined government and commercial standard formats				
Support requirement workflow from initial capture to review to approval/signoff				
Support automated notification (email or some other means) as requirements are handed off from one team to another—e.g. automatically notify appropriate personnel when a requirement is ready for approval				
Support baselining of requirements				
Support comparison of baselines of requirements				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support setting access permissions for individual users to different requirement baselines or other types of requirement groupings				
Support automatically maintaining audit trail for requirement changes (user, time/date, annotation of change, and change detail, etc.)				
Support automatic notification of affected project participants by e-mail about requirement changes				
Provide for automatic generation of reports of requirements capture progress/status/velocity using tool provided and user-defined metrics				
Provide for reporting on the maturity of requirements using requirement status and number of reviews per requirement				
Provide ability to work disconnected (i.e., no connection to ALM requirements repository) and merge changes upon reconnecting				
Support assignment of different users to work on different groups of requirements on the same project				
Support merging of changes made by different users on a set of requirements				
Support definition of glossary for requirements that allow user to hover or click on a keyword in a requirement to get the definition				
Provide for capability to compare/differentiate between different versions of the same requirement				
Support defining priority for a requirement				
Support assignment of requirements to owners who are responsible for reviewing and approving them				
Support capability to define/assign verifications methods to requirements				
Provide for the capability to re-use the same requirements on different projects				
Provide for the capability to mock up screens based on defined user interface requirements				
3. DEFECT MANAGEMENT				
Provides a centralized and/or distributed repository for tracking and managing defects across projects				
Support tracking of software defects against any ALM artifact, such as requirements, software build versions, designs, test scripts, etc.				
Support capability to add a defect against any ALM artifact, such as requirement, software build version, design, test script, etc. using a pre-defined or user-defined template				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support capability for user to define metadata to capture each type of defect (i.e., create custom data fields to specify, for example, who identified defect, where was it found, status, etc.)				
Support automatic creation of unique ID for each defect with the option of using a user-defined template				
Support documenting a defect with rich text formatting				
Support capability to edit defect—with an undo functionality				
Support capability to delete defect and related artifacts—with an undo functionality				
Support capability to link defects to other ALM artifacts, such as software builds, requirements, test scripts, etc. so to trace the source/cause of the defect				
Support organizing defects by user-specified hierarchy/groupings with sub-grouping capability				
Support describing a defect with an image directly in the database (e.g., a screenshot of error)				
Support linking defects to actual documents in a SharePoint location (i.e., clicking on link opens the document)				
Support linking defect of the same type (e.g. multiple defects may have a similar cause)				
Support to import defects from other defect management tools				
Define relationship between requirements (e.g., “traces from”, “dependency”, etc.)				
Support visual representation of requirements’ traceability results (e.g., displaying a matrix table that maps defects to software builds to requirements)				
Support defect workflow from initial capture to review/categorization to prioritization to assignment to verification to closure				
Provides automated notification (email or some other means) of resource assignments to defects				
Support automated notification (email or some other means) as defects are handed off from one team to another—e.g. automatically notify appropriate personnel when a fix for defect is ready for verification				
Support automatically maintaining audit trail for defect status updates (user, time/date, annotation of update, and update detail, etc.)				
Support automatic notification of affected project participants by e-mail about changes in defect status				
Provide for automatic generation of reports of defect resolution progress/status/velocity using tool provided and user-defined metrics				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Provide ability to work off-line (i.e., no connection to ALM defect repository) and merge changes upon reconnecting				
Support assignment of different users to work on different defects for the same project				
Support merging changes made by different users on a defect ticket				
Support definition of a glossary for defects that allow users to hover or click on a keyword in a defect ticket to get the definition				
Provide for capability to compare/differentiate between different versions of the same defect ticket				
Support defining priority for a defect				
Support assigning severity level for a defect				
Support user-defined hierarchy of priorities that can be applied to defects				
Support user-defined level of severity that can be applied to defects				
Support user-defined states that can be applied to the status of a defect ticket				
Support capability to define/assign verifications methods to defects				
Support the linking of a defect to software builds or test cases/scripts or other ALM artifacts to trace how the defect is resolved and verified				
Support user-defined criteria for defect ticket closure (e.g., user defines what processes must be completed before a ticket can be closed)				
Provide management reporting, for example, the number of open defects grouped by various criteria such as open defects by project, severity, and priority				
Provide for the capability to capture other items in addition to defects, such as customer suggestions, customer complaints or enhancement requests.				
4. CONFIGURATION MANAGEMENT (CM) REQUIREMENTS (incl. Source Code Management and Build & Deploy Management)				
Provide a centralized and/or distributed repository for tracking and managing source code and other development related changes, including binaries, across projects				
Support version control of CM build artifacts				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Supports check out/check in of source code by enabling developers to check out a file by downloading a copy from the central repository to their local computer before modifying it				
Support the capability for multiple developers to modify the same source code file at the same time—the ALM tool shall be able to merge the changes from the various developers into the configured item				
Support branching and merging source code in various software baselines				
Provide for user to view, print, save the differences between two or more baselines/branches				
Support the visualization of a software baselines and associated branches in the form of a repository tree				
Provide capability for developers to enter to annotate their code changes to baseline (e.g., who made change, when, why)				
Provide for capability to view the check-in history (date, comments, etc.) for source code file				
Provide the check-in history for each file, including the comments made by each developer when checking in the file				
Support the tagging or labelling of a release				
Provide for systematic control of changes to the source code and other related artifacts for maintaining the integrity and traceability throughout the system's life cycle				
Support the capturing, controlling, and correlating of configuration items (Cis), configuration units (Cus), and configuration components (CCs) consisting of source code and configuration files (e.g., .c, .xml), scripts, documents, directories and other development artifacts, within a number of individual baselines across the life cycle				
Support renaming or moving a file or directory into the repository				
Support the identification, control, restoration of software baselines				
Support automated integration with revision control system for any linked documents – (e.g., SharePoint, CVS, SourceSafe, etc.)				
Support linking of a CM artifact, CI, CU, or CC, with other ALM artifacts, such as requirements, defects, tests, etc.				
Support automation of CM tasks such as source-code check-in (commits)/out, baseline branching, etc. by creating scripts or other means				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support development environments on multiple and different operating systems, such as AIX, Windows, Linux, Unix, etc.				
Integrate with other integrated developer environments (IDE), such as Eclipse, Jbuilder, NetBeans, Jcreator (for Java), Visual C++, C++ Builder (for C++), Visual Studio (for Visual Basic .NET).				
Provide capability to implement continuous integration (CI), including integration with other CI tools such as Jenkins/Hudson				
Support Java and .NET development build				
Support traditional (e.g., waterfall, spiral) and agile development (feature driven) methodologies				
Support multiple users executing CM capabilities of the ALM tool at the same time on a given project				
Provide for graphical user interface as well as command line interface				
Provide capability to automatically compare different versions of a CM artifact (e.g., doing a “diff” command of different versions of a .c file)				
Support update features to manage complex tasks of tracking and applying software updates to client computers in the enterprise				
Support automated dashboard report generation of user-selected metadata relating to users, user groups, tasks, schedules, hardware and software inventory, software updates, site status, and other CM operations				
Support importing source code from other repositories				
Support manual (user-initiated) software builds				
Provide for a means to automate software builds—i.e., retrieving the right release files, building it, and packaging the build outputs for installation				
Support automated software builds triggered by a user-defined event, such as a code commit or schedule or some other dependency				
Support the capability to execute multiple builds simultaneously				
Provide the capability to queue multiple software builds based on user-specified criteria, such as schedule, dependency, etc.				
Provide the capability to build on as well as to deploy software products to a virtualized environment				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Provide auditing capability to verify build against correct software releases (i.e., ensure that a build consists of binaries generated from the approved/correct versions of the various software modules)				
Support integration with external build tools like Ant, Maven, Make, Command Line Tools, MSBuild				
Provide simple software (including OS, security, configuration) deployment platform that automates deployments, updates, repairs, and removals to all computers/servers across WMATA's network				
Support the automated updates of OS, software, file, etc. to all computers/servers in the enterprise				
Support the full range of deployment models from full bare-metal deployments to incremental patch updates				
Provide for automated rollbacks of the entire application deployment				
Provide for automated rollback of patch (application, OS, security, configuration, etc.) update to all client computers/servers in the enterprise				
Provide for an automated deployment capability that supports rapid feedback and continuous delivery in agile development with audit trail, versioning and approval workflow				
Provide for the capability to push any appropriate ALM artifacts (e.g., software build/release package or update, shell scripts for automating tasks or tests, etc.) to all client computers/servers in the enterprise				
Support the automated capture of configuration and security differences across servers and environments				
Provide for automated dashboard and reports identifying what is deployed where and who changed what and when				
Integrate deployment management with configuration management, build and test management functionalities of the ALM tool suite to automatically deploy, test, and promote new builds				
Support different rights to various teams for "push-button" deployment of their respective applications and changes to their respective environments				
Support release management and planning capability for different development approaches. For example, in Agile, provide capability to capture, track, and status product backlog items that through release planning are assigned into release backlogs that are assigned for implementation during an iteration planning session				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Provide for automated generation of dashboards and reports with release status, velocity, and other related information necessary for effective planning and execution				
5. QUALITY MANAGEMENT REQUIREMENTS (incl. Test Case and Execution Management, Load/performance and Functional/Unit Testing)				
Provide an automated functional testing and regression testing tool that supports functional, regression, GUI, and data-driven testing				
Provide an automated testing tool that supports a wide range of web-based applications, such as .Net, Java, Siebel, SAP, terminal emulator-based applications, PowerBuilder, Ajax, Apache/Adobe Flex, Dojo Toolkit, Eclipse Graphical Editing Framework (GEF)				
Support the design and creation of tests for verifying that a proposed release meets the requirements associated with a project				
Support test design and creation by storyboarding—i.e., using natural language, process flows, graphics, etc.				
Support test design and creation based on different programming paradigms (component based, model-driven, etc.), languages, operating systems (OS)/platforms, network configuration, etc.				
Support test design and creation based on application specific characteristics that require specific testing techniques, e.g., web-based, GUI testing, real-time testing, database testing, protocol processing testing, etc.				
Support capability for user to define metadata to capture for each type of test (i.e., create custom data fields to specify, for example, test creator, test development status, type of test: functional, stress, exception, performance, status, etc.)				
Support automatic creation of unique ID for each test with the option of using a user-defined template				
Support documenting a test with rich text formatting				
Support capability to edit test—with an undo functionality				
Support capability to delete test and related artifacts—with an undo functionality				
Support the linking of a test artifact to requirements and software products to defects/software bugs				
Support linking requirements to testing status (i.e., from a requirement, user can see test-related data, such as test plan and test coverage, whether test was executed for a given software release, and if so whether it was successful)				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support the creation, editing and organization of different test artifacts like test cases, test scripts, test plans and test suites				
Provide capability to record user actions with multiple customization options and intelligent script maintenance capabilities				
Support the capability to automate the execution of test scripts				
Support organizing tests by user-specified hierarchy/groupings with sub-grouping capability, for example, grouping tests by data sets, by functionality, features, or software module, by test environments, etc.				
Support test planning, such as the capability to define what tests to run against which baseline, when to run them, etc.				
Support the automatic generation of an overall draft User Acceptance Test (UAT) scripts (based on defined test cases/scripts) for a baselined set of requirements against a targeted software release				
Support creation of a regression test suite that can be executed against every release				
Support the scheduling of tests to be executed (manually or by a test execution tool)				
Support test workflow from requirements capture analysis to test planning to test development: design and creation to test execution to test reporting to test results analysis to defect retesting to regression testing to test closure				
Integrate test management tool with requirements management, defect management, configuration management, build and deployment management so to provide mapping of a test object with other aspects of the application life-cycle				
Support simultaneous project participations at different locations, including creating, modifying, and executing tests				
Support integration with unit testing tools such as Junit, SpecFlow/Selenium, ReSharper, DevPartner, etc.				
Provide for automated checking of source code for quality using unit tests, code coverage, testing for memory leak, security scans, performance analyses, etc.				
Support automated generation of dash boards and reports that provide metrics and information about test progress, such as number of tests executed, number passed or failed, number of incidents raised, fixed, or outstanding, etc.				
Provide test monitoring capabilities, e.g., tracking of the estimated and actual time with metrics to measure the progress of testing activities as well as test coverage from different sources (e.g. requirements, test cases), etc.				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support designing and creating test cases for the required test level (unit, integration, system)				
Support designing and creating test cases based on different testing methodologies: static vs dynamic, box testing (white, black, etc.), specification-based testing, visual testing, etc.				
Support designing and creating different types of tests: e.g., installation testing, compatibility testing, sanity/smoke testing, regression testing, acceptance testing, alpha testing, beta testing, destructive testing, security testing, etc.				
Support designing and creating data-driven tests that enable user to execute the same series of tests with varying sets of test data				
Support designing and creating tests, including templates, based on various testing processes: traditional waterfall development model. Agile or extreme development model, top-down/bottom up, etc.				
Support capturing/definition on test data schema/layout for generating data used for executing tests				
Support prioritization of test cases, test suites, etc.				
Support valid/invalid data generation or provide for integration with test utility that generates/simulates test data or plays back recorded data for test execution, such as Mocaroo, DataGenerator, etc.				
Support the management and maintenance of test data				
Provide or support integration with test utility that enables step-by-step execution and conditional breakpoint at source level or in machine code				
Provide for or support integration with performance testing tool, such as Apache JMeter, Rational Performance Tester, LoadRunner, etc.				
Provide for or support integration with benchmarking utility, such as ApacheBench, Curl-Loader, Httperf, etc.				
Import test artifacts from other testing tools				
Support automated test execution triggered by a user-defined event, such as software build/release, schedule or some other dependency				
Support setting-up and tearing-down of the test environment/pre condition and respectively the post conditions for a set of test cases				
Support roll-back to initial state in case of unexpected errors during test execution				
Support stopping or continuation of the execution of a suspended test case				
Support logging information on executed test cases				

TECHNICAL REQUIREMENTS	Response - please check one for each requirement			
	Yes	Qualified No - partial support	No support	Explanation /Comments
Support the definition or capture of expected outcome: outputs, state, or criteria for a successful execution of a test case				
Automated comparison of test outcomes with expected results to automatically determine if test passed or failed				
Identify how/why test failed—for example, expected output not produced, test did not complete because application crashed, test outputs different from expected, etc.				
Creation of automated tests to be (re)used for regression testing in other projects				

Appendix II:

Customer Service Evaluation Template

Customer Service Evaluation Criteria		
		Response
A. Technical Support	1. Coverage (hours/days of the week available)	
	2. Number of qualified technical support resources available for remote help	
	3. Number of qualified technical support resources available for on-site help	
	4. Methods of delivery (telephone, online, email, etc.)	
	5. Dedicated POC assigned?	
B. Documentation	6. Content – topics covered	
	7. Modes of delivery (e.g., paper, online interactive, pdf, etc.)	
	8. Frequency updates	
C. Knowledge Transfer (Training)	9. Availability (course offerings and locations)	
	10. Modes of delivery (e.g., learning center, customer site, interactive web, self-guided online, etc.)	
	11. Comprehensiveness (e.g., type of users, skills and skill levels)	
D. Consulting Qualification	12. Technical expertise and experience of consulting resources in implementing proposed ALM solution	
	13. Strength of qualified consulting resources	
	14. Geographic availability of qualified consulting resources	
	15. On-site consulting availability	