**ALM Transcript**

Slide no – n Title –asdfasf

Thank you for an opportunity to present ALM i.e. Application Lifecycle Management presentation. Agenda for todays presentation as follow - We will cover what is ALM? What is Cybage ALM Vision? What are the benefits of ALM? What are the technologies and Tools? What are the services we offer in ALM and Devops, ALM tool Stack and integration between all tools. At a Process level we will discuss Continuous Integration & Delivery and AT lastly we will discuss Devops Ecosystem and How cybage implements ALM and Process.

Application Lifecycle Management (ALM) is a continuous process of managing the life of an application through governance, development, and maintenance.

ALM tools help managing life cycle from planning, development, build, test and release. Let me share Cybage ALM Vision with you. ALM has objectives to enhance efficiency, visibility, transparency, and collaboration. Different benefits of ALM is efficiency gain and visibility is very important the transparency in reports we get through tools. So we get transparency in reports of ticket items and build and testing reports. The collaboration is important where different collaboration tools are used where all the team members and stack holders are working on single platform sharing views and ideas and everyone also knows what is happening in the project. Where all the benefits of ALM Efficiency is very crucial which improves quality and quality gates are defined which works in automated way. So different Quality Gates listed here, One is Code review, Automated Code Review which includes compulsion of code review for developers, where developers can’t merge their code into upstream branch without code review performed by one or two team leaders. Static Code analysis is important here, without performing Static code Analysis code can’t be merged to upstream branch until Major and Critical Blocker issues being resolved. In automation we can achieve enhanced productivity and also you can remove manual steps here. There will be no manual intervention and manual resolution of issues. It is complete automation pipeline which leads the efficiency and accuracy in the processes. Reduction in release cycle time, time it takes code commit to code deployment is most important factor for moving towards agile and it cuts down all the waste and bottlenecks in the pipeline. So we have a faster release cycle time using ALM tools and Devops. In next section we have transparency, everyone have distinct and defined roles & responsibilities for Developers, QA and operations team. It leads effective collaboration between Operations and Developers. Team Maturity increase because of the agile processes and reporting. Audit ability is there, every action or steps taken by developer or QA or Operations Guy will stored in audit trail. So we can see who have committed the code, on every commit why the build failed who is responsible for build failure, why did the test fail. who wrote the unit test So everything can be auditable and hence everything becomes transparent here. Collaboration is also an important factor now Chatops is becoming famous where people are using single platform for communication between the teams and decision making. In this process everyone can see the documents created by Business Analyst at a one place with updated versions and one can also see code review comments, We can also get continuous feedback and voting for from team for better decision making on any issue. For CI and CD Process continuous feedback is very important. Real time dashboards and reports of CI & CD Process also help Top management people to take effective decisions. One can get burndown charts and sprint velocity charts, How many developers are lagging behind their assigned tickets. How many tickets are released for particular release version and also it reduces the dependency. Next we move to visibility, It makes an open environment for managing the project. Everyone can be on same platform because End to end traceability is there. From requirement gathering to production release, we can trace every phase of SDLC with two way traceability. It also gives the total time of ticket moving from to do state to deployed to production state so it gives us the predictability. We have automated scripts and process for the deployment so it will never fail or stop so it gives us the better predictability and more confidence. Analytics of various data points, it is very important because we get reports of how many times build failed, Which developer is performing better than other developers for last two months and also we can see automation test reports here why some tickets or features are always failing in the automation testing and you can do the root cause analysis so we can use analytics on various datapoints and also we can use BigData here for analysis. We can also do log and operational matrix monitoring and we can get very dynamic reports here.

Now we can move towards Cybage ALM technology steams, We can do ALM implementation in two technology streams first one is Microsoft Project, we have invested lot of time in creating TFS Microsoft project. We have also Opensource Technology, under opensource technology, we support JAVA, LAMP, Python, Perl, JS and SCALA technologies. We have also expertise in Atlassian tools. We have implemented Atlassian on premise tools in lots of Projects and Oncloud tools as well. Now most of the clients are moving towards the onCloud Services of Atlassian tools because it has less maintenance, operations and also they provide auto backup store and auto upgrading.

What all services Cybage offers in ALM? We will discuss about it. Process of GAP assessment and end to end implementation and training where we work on a project we do the the GAP assessment We do work on the solution document after finding the GAP. We do the complete end to end implementation. We work along with the team for solving pain area and eventually we have complete end to end detailed training on implemented tools so team members can also implement and move forward with the tools. All the ALM solution and stuff will be taught to the Team members and guidance is provide to team members. We can also do consultancy for Pain areas of project. We suggest list of tools and process to solve the problems of the project. We have good consultancy experience and record in cybage project. Once we implement ALM in Project then we do the audit where we do the ALM Maturity Level Check of ALM Implementation and regular audit trails are performed to check ALM Maturity Level of the Project Team. We also give suggestions to team to move high towards the ALM Maturity Level.

We also offer DEVOPS service, we have lots of experience and records of devops implementation in our projects. In devops, we offers services like Infrastructure as Code, provisioning of system and configuration management we do this in automated way. Because manual configuration of environment is very complex, there can be manual errors occurs. It takes lots of efforts and time and it can’t be stable. So what we do whenever you want to configure or provision your cloud, Bare Metal Machine , VMs, we write a code or scripts to automate the configuration and Provisioning of systems. We can do this using CLoudformation, chef, Puppet and Ansible tools. In cybage we write your code scripts, once script is ready, it is tested and code review and SCA is also performed. Once script is validated, we try to do repeated infra setup and infra provisioning using the code. So we have extensive knowledge on chef puppet ansible. Now We have configured lot of infra on AWS Cloud using Cloudformation in Cybage Projects using IAAC. Secondone is Continous Integration, it talks about once your code is committed and how you integrate your code on daily basis in Global Repo. So every developer have to check in and merge his code in global repo on daily basis which is a best practice so he can find errors at a correct time. He also get a good feedback on regular basis and early. Once you check in to the branch it will automatically trigger continuous integration process. We extensively work on Opensource Jenkins CI Server, We also have expertise in pais version of Jenkins provided by Cloudbees and we have a partnership with Cloudbees for better support and technology. We also provide service on Atlassian Bamboo, Clodbased CI Servers like Codeship, CircleCI etc. What is Continuous Integration Tool? In this process, we try to have a automated build tools like Maven, Grunt and Gulp. We can perform complete Compilation, run unit tests, packaging and its versioning and storing packages in Artifactory and Nexus like artifactory management tools and automated deployment to various environments. Now we also use Docker in Continuous integration pipeline where you can test your code and convert it to image and deploy or push this image to Dockerhub which again becomes your artifactory manager. So CI is the first step for moving towards ALM and Continuous Delivery. So CI is most integral part for starting CD Pipeline. Continuous Delivery and Deployment is next Phase. So we have 3 different Phases Continuous Integration, Continuous Delivery and Continuous Deployment. In CI we check the stability of the code, we don’t look at the functionality. In CD, We do pipelining where we have multiple environment, we try to do automated deployment in multiple environments. For example, we first deploy into QA Environment, where one can do automation UI Testing using Selenium and Nightwatch, then after successfully passing from QA Environment, Code will be deployed into Staging where Load and Performance Testing using Jmeter and Gatling will be performed. Also we can perform Security Testing here. Once we test our code on multiple environments, we come to know that our code is stable. This whole process we have implemented using Docker in some projects, same Docker image is tested and stable. It is run through the multiple environments, once image is completed tested and stable from CI Process, We can deploy it into multiple environments using CD Pipeline. Once CD Pipeline is complete, we take the artifactory, and version it and store it into artifactory manger and in last we deploy it into Production Environment. In CD we are not deploying to Production, we just versioning the artifactory, and ready to deploy. Next phase in maturity level is Continuous Deployment; It is the highest maturity level of ALM and Devops where all Quality Gate and Promotion Build is executed. All the deployment and configuration scripts are automated, once the developer commits the code and it passes all the Quality Gates implemented in CD Pipeline, it will directly deploy into Production Server. Once deployed into Production Environment, We need zero downtime. For that we implement different strategy like Canery Release , rolling Upgrade and red-blue deployment. Once this all strategy is implemented for zero downtime, latency will be decreased. Your client will not get ant 503 status code while accessing the application. Where your server will up 24\*7 and no need to down your server even for a seconds. It is most important part of the continuous deployment. If your deployed version is not as per your requirement, you can do auto rollback. So Auto rollback is also important factor for Zero Downtime and everything is automated here. After deployment, Continuous monitoring will be done i.e log tracking and application health checking. It is most important factor of Devops Culture. There are multiple tools available in market for continuous monitoring. If you are using cloud then cloudwatch and cloudtrail is good for continuous monitoring. If you are using Docker container then Sysdig, Ruxit and Appdynamic is there. So every tool provides automated continuous monitoring where you can put automated alarm and alert system and auto notification system, where one can get messages to emails, slack or hipchat server communication channel. Once you get notification one can easily and quickly solve the production issues by doing continuous monitoring. We can have self healing system on production where application can automatically restart or scale when any issues occur on production environment in cloud. So developer doesn’t need to worry about solving the issue. Next section is virtualization and containerization. Virtualization leads to effectively increase the hardware usage and capacity at a low cost. We can run different loads on single machine. One can easily scale application using virtualization. For virtualization, there are different tools like Virtualbox or VMWare Player. Now Enterprises are moving towards Containerization, Docker and Rocket is providing containerization solution. So using container one can run different application on a single machine with different workloads. They are very effective and lightweight than virtual environment, virtual environments are very heavy and takes more time to load and boot. Where containers take only milliseconds to boot and load. You can run 100 containers in a single machine at a time where you can only run 4-5 VM on a single machine. Hence you get benefit of increased productivity at reduced cost. So containerization has important place in CI, Continuous delivery and deployment. It gives portability, the same container will run on AWS Cloud machine, also run on Azure, on Google Appengine and your bare metal machine.

The Devops Ecosystem, it furnishing nowadays because of the cloud service provider. AWS, Microsoft Azure and Google AppEngine is using Devops culture and tools. They are also providing services for opensource tools and they are also contributing into the opensource devops tools. Cloud based service provider are promoting Devops, so it is easy in cloud environment to setup and provisioning environment using devops tools. You can have docker container on Cloud like Elasticsearch container service on AWS. Whereas Microsoft Azure also gives the Team service where one can implemented entire CD Pipeline, It provides code commit in version control system and multiple environment configuration where we can perform all type of testing on Cloud environment. Same with AWS, It gives three important services like Code Commit, using git repo, Code deploy, it provides services of auto deploy on different stack like opsworks and Elasticbeanstack. Also we can use Code Deploy for Docker now. Code Deploy is fully automated, it provides auto rollback with Zero Downtime with Canary Release setup and Also every code deploy is versioned and everything can be auditable. In Last Code Pipeline, one can implement complete CD Pipeline using code Pipeline where one can perform all type testing just using tools provided by AWS. One just need to call REST APIs for performing all type of testing. So one can now implement whole CI-CD Cycle using Cloud Machine rather than using bare metal machine and servers. Everything is automated here where Pipeline gives you the detailed status about any issue occurred within the pipeline. One can reverse the code commit and we can also know which developer have done changes in the code. It is completely auditable with full transparency.

In ALM tools Integration we will discuss different ALM tools required for different ALM Phases. First for Requirements Gathering, Business Analyst gathers and understands requirements from clients and makes a document. We recommend Atlassian Confluence for storing all the requirements Documents at one place where all team members can access and review these documents. We can also version and manage all the documents using Confluence. We can also store How to Articles, FAQs, MOMs and Technical Documents using confluence. If BA wants to change the requirements, he can just change the change page in requirements and update it. So there will be no any other multiple documents for requirements it solves the problem of multiple requirement understanding by Team members. It will also reduces dependency because all the documents are at one place so no need to worry for asking documents from team members. It also provides search option so one can easily search document, he/she wants. Every Team Members like developer, QA, PM and Client will be on same level of understanding of requirements using confluence. So Confluence is a collaboration tool where we can manage all the documents, one can give comment on any requirements and gives complete solution for requirement gathering at one place. Next phase is Project Management, It includes How we manage Projects and Task Assignment within team according to requirement given by Client. In Project Management, nowadays all are using Agile Methodology. To implement Agile Methodology, there are two frameworks available first is Scrum and second one is Kanban. Scrum Framework is a timebound process which consists of Sprints of 2 weeks and Small Team Size. Kanban Framework mainly depends on the Work in Progress Limit defined. By Controlling WIP Limits, Project tasks are performed. To implement these frameworks we recommend Atlassian JIRA which is a Issue and Project Management tool. One can easily manage and allocate issues or task within the team. One can also create Sprintwise Scrum Dashboards for Task status and progress tracking. It gives options like estimation where developers can give estimates for assigned task and log work against issue assigned. It also gives BurnDown Chart and Velocity Chart for proper estimation and task assignment and planning. It will also give you detailed reports about task assignment. We can also do release planning using JIRA. So it gives better governance of Project. We can also do portfolio management using JIRA.

Then we move to the development phase, where developers are using multiple tools of ALM. Every developer uses GIT or SVN Version Control System for Managing Repo. Every Developer commits code in the repo and gives Jira ticket in commit message which gives proper tracking with JIRA and all tools. There are also different repo version control systems like mercurial and perforce etc. Repository Manager is web based repo manager which sits on GIT, which gives multiple features like user management, authentication, authorization, branching, branch management and pull request for code review. Number one repo Manager tool in market is Github. It has millions of repo hosted with millions of users contributing their code into it. We have Atlassian Bitbucket and FishEye-Crucible for repo management and code review. Nowadays Bitbucket is very famous for repo management. It has two versions like bitbucket on server premise and Bitbucket oncloud. In next phase Unit Testing, Developers work on different unit testing frameworks because we work on TDD Methodology where developer writes unit test for proper understanding of requirements and he has to pass the unit test. In JAVA we use Junit and TestNG for Unit testing. We use Junit because it can easily run through Maven, Gradle and ANT. For PHP we use PHPUnit. Every programming Language has different unit test tools. We provide services like how to write Unit tests? How to automatically run Unit tests and generate reports using Jenkins and Atlassian Bamboo?

In next phase we talk about, Test Case Management, It is also most important tools. In this tool we can manage manual test case written by QA and Developers. It’s a management of test cases version and which test case is linked to which Jira issueid. So we will get traceability between Requirements, Issues or Tickets and Test Cases. Before releasing or deployment of any versions, one should know about how many test cases are failed or passed. We can also get graphs of failed vs. passed test cases. We provide services on Tools like Zephyr, TestRail and Testlink for Test case management. We have our separate QA team for ALM. We have implemented test case management in various projects. We also have Selenium for Automated UI Testing, and for Load testing Gatling and Jmeter.

Now moving towards CI servers, Which runs CI pipeline, for running a build we have Maven, ANT, Gradle and Gulp for JAVAScript. Moving towards Static Code Analysis, We use Sonarqube which has its own database and dashboards for generating real time and historical Reports. We have different rules for different programming languages in Sonarqube. We get different Static Code Analysis reports according to different rules defined in Sonarqube. We can see Major, Critical and Minor issues reports in Sonarqube. For unit testing reports, we have also different Code Coverage tools like Cobertura for JAVA which shows how much percentage of your code is covered with unit test. Best practice is 80 % code coverage. Anything higher than 85% is very good. In CI Server we extensively use Jenkins and Bamboo for triggering build and packaging. We store and version packages in Artifactory or Nexus when code is stable for future deployment.

Moving towards Devops, In devops we have main 3 configuration management tools Chef, Puppet and Ansible and for provisioning in Cloud we have Cloudformation. We can also provision environment on AWS cloud using Chef and Cloudformation. We can also provision multiple Virtual machines and bare metal machine at a time using these tools. Under Chef Master we can write chef scripts or cookbooks to provision and configure multiple environments on different servers. We can also perform same task using Ansible scripts and Puppet Scripts. So All three application is most useful in automation of Configuration and Provision of Servers. In Devops, managing a application is very easy. We have also expertise in Cloud Service Provider services like Amazon AWS and Microsoft Azure. We have also expertise in Containerization using docker. How one can manage multiple Docker containers, How one can orchestrated Docker containers. How one can schedule Docker Containers etc.

In next slide we will discuss about common tools used in ALM Process, First is Project management where issue assignment, Release planning is done. In next phase during CI Process, when developers checked in the code in global or share repo as fast as possible after testing the code in the local repo. Once developer commits the code automatically CI Server will start CI Process or build by checking out the latest code from repo manager. So you can say that we have webhook integration between Repo Manager and CI Server. Once build starts CI server checks out latest code, complie it and run unit test like Junit or PHPUnit using automated build tool like maven or gradle or ant. We also perform Static Code Analysis during Build Process. Once build is stable i.e Code is stable we package it and stored into the artifactory or nexus with proper version of builds. We can also get different reports for different build run on CI Server. It gives result like pass or fail. If Build is failed, then it revert back the changes in the git branch. If build is success, we move towards CD Pipeline. Build Status like pass or fail can be notified to Technical Lead or Developers via Email or Communication tools like Slack or Hipchat. If Build succeeds, code is automatically merged into the upstream branch. Once build succeeds, we try to deploy it to target server where we can test it further or we can directly deployed it to Production Environment.

**Continuous Delivery Definition Add.** We should ready with stable build for delivery anytime what the client needs meeting all the functional and nonfunctional requirements. So there is no wastage. It helps us to reduce release cycle time and cost by delivering incremental changes to users following set of practices. So basically CD Process starts from code commit by developer, it will start CI Process( Compiling, Running Unit Test and integration Testing, and Packaging) as a final product we get stable build package, it will be deployed into QA environment for Automated UI Testing. After passing all the tests in QA, build will be deployed into Staging Environment, where performance and load and security testing is performed. This whole process is automated without any manual intervention. So this is an entire CI-CD Pipeline, we can also increase more environments for more regression and quality gates for improves quality.

Now we will talk about the Devops Ecosystem, What is Devops? It is a culture of collaboration between Developers and Operation within Project Environment. They both should be working together. They should be working on same set of tools. They both should be on same pipeline. They both should take any work like developer should use operations tools and Operation guys should take Developers Tools for Provisioning and configuring Development and Operations Environments. As we talked about provisioning, we have three major tools like Chef, Puppet and Ansible. In Cloud, we have Terraform by Hashicorp and Packers also from Hashicorp which is also used for writing amy scripts. You can also use multiple Docker images for cloud providers, we have Cloudformation by AWS. In virtualization, we have different virtual machines. We have hypervisor like Vagrant to create your virtual environment. We also have docker Containers and Rocket Containers. Docker is being used 75% in virtualization market. In Load Balancer we have nginx and HAProxy which is again very critical tool to distributing of your load for high availability and scalability. We also have ELB for AWS. In Cloud Providers we have AWS, DigitalOcean and Azure. We have also DockerOS in which all the services and Application runs in a form of Container which system or init system is Docker. We have RanchorOS , CoreOS and Snapy as a Docker OS. For Data persisting in Docker, How we can manage Data in Container? How data moves once the Container moves which is critical for mysql, mongodb or cassandradb. So we can use NFS, GlusterFS and Flocker. For continous monitoring, How much CPU and RAM is utilized by Containers are observed. Which container is down and why it is down can be monitored. If you want to see complete UI or dashboards of application containers at one place, we can use ELK. For log monitoring we can use Sysdig, they have also partnership with mesosphere. Ruxit is also important tool. Nagios is used for managing operations and resources hardware, managing your RAM and CPU of a machine. Service discovery is again important in microservices, where every service want to discover where is the service and How to access the service? What is the metadata of the service? So we have counsel by hashicorp. We have etcd by coreos and Eureka by Netflix. In Alert System we have Pagerduty and CloudWatch. Where you can configure alarm for any issue, when this issue occurs automating alert is generated. Alert can be sent to SMS or Email or Slack Channel. Pagerduty is one of the most used tool in market and Cloudwatch is provided by AWS. For orchestration, once you created your container and placed in ecosystem, if you have 100 or 200 or bigger node cluster, if you have multiple service application and for every application you have different containers, How would you scale your containers or run your containers. How to deploy the container and how to do see your network working on Container level? Which container talks to which container? Which container is allowed to talk to which container? How do you manage these containers. Everything is managed by orchestration tools today. Kubernetes from google is mostly used tool for orchestration today. It is being used by most of the big companies like amazon, google, Netflix, Rackspace,etc. It covers the 70% of the market today. Ranchor is Provided for Container management. We also have google Appengine providing Kebernetes. Mesos from Mesosphere with different frameworks works well for Container management and scaling. Microsoft Azure has also support for Mesos. AWS has also support for Mesos. Ranchor has also a container Manager service. It makes easy to manage and run containers using UI for Developers and non Operation guy. It provides you everything, which a container manager requires. Docker Swarm is also native container Manager and orchestration tool which provides scheduler and most deployment features.

Moving to Docker Ecosystem, Docker plays important role in Devops today. It simplifies the life of operations and developers. It effectively manages multiple and distributed application. Once you create an Docker image, it runs on any Cloud Machine or bare metal machine. If developer tests application in Docker container image. This same image will run on all QA, Staging, etc environments. So Docker gives same control and visibility in all environments. It also gives version controlled environment to Operations, QA and Developers. If QA wants to test application, he can take this image from Docker Registry or Dockerhub and run it on any OS. We just build it once and ship it anywhere which gives portability. Docker machine gives docker client we can build, write and run containers. Docker engine is also a important part of Docker Ecosystem. Dcoker Compose is used when we want to run group of containers and don’t want to write multiple run commands for running multiple containers. We write simple docker-compose.yml. So we can start multiple containers running single file. So it makes simple to link and group containers into a single unit. If you have 100 or more node cluster, we can use Docker Swarm for orchestration, we can setup one Docker Master and using it we can deploy and manage container on multiple machines from a single machine. For orchestration we also have kebernetes, Mesos and Marathon. Marathon gives us scheduler used by twitter on Production. Chronos is also tool for short running jobs whereas marathon is for very long running jobs. One can also run Jenkins, Hadoop, Spark and microservices using marathon. Jenkins provides a plugin for running Jenkins slave as a docker container using Mesos slaves. We can also schedule Jenkins slave using kubernetes slave. Lets support we have 50-100 jobs in queue, so we can schedule these jobs using Mesos Slaves nodes to speed up CI Process for faster feedback. Even if you have 1000 jobs it can be concurrently run on cluster of nodes of Mesos. For CAAS, we have Ranchor and IBM Bluemix also provides docker containerization and scheduler. It is paid whereas ranchor is free. Docker and Cloud, ECS is a new service provided by AWS where one can schedule containers, can do fault tolerance, high availability and Load balancing of the containers. We can also have rolling upgrade with zero downtime. Microsoft Azure also supports Docker in their Linux environments. For Docker Monitoring, we have CAdvisor which is free opensource tool from google for managing containers and sending operational matrix of the container to single repository where we can do analytics. For Docker state full we have flocker and convoy. Then we have sysdig and Ruxit. For docker networking we have weave. Regarding Docker Registry we have DockerHub and AWS ECR. AWS ECR is new service for AWS which is elastic Docker registry where we can have complete authentication, authorization secured environment with encryption keys. Which will store your docker image in AWS which is more secured than storing on Dockerhub. So it is all about Docker Ecosystem.