OneShield CICD

Technical Design Document

Prepared by: ValueMomentum Inc

Status: Baseline

Version: 0.1

Last Updated: 03/01/2018

The purpose of the Technical Design document is to fully document for engineers the technical specifications for developing the new system or making changes in existing systems.

The goal of this phase is to produce a reviewable specification, which can be the focus of a technical walkthrough and then approved.

**Note:** All required areas must have "N/A" if not applicable with a short explanation describing the teams reasoning rather than just leaving it blank or removing it.

**Audience:**

Hanover Enterprise Architecture

Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Editor** | **Change Description** |
| 03/05/2018 | 0.1 | ValueMomentum team | Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

Contents

[1 Project 5](#_Toc508278885)

[**1.1** **Project Summary** 5](#_Toc508278886)

[**1.2** **Business Driver** 5](#_Toc508278887)

[**1.3** **Business Objective** 5](#_Toc508278888)

[**1.4** **In- Scope** 5](#_Toc508278889)

[**1.5** **SLA Specifications** 6](#_Toc508278890)

[**1.6** **Out of Scope** 6](#_Toc508278891)

[**1.7** **Assumptions** 6](#_Toc508278892)

[2 Solution Architecture 7](#_Toc508278893)

[3 CICD Pipeline with Jenkins 8](#_Toc508278894)

[**3.1** **Prebuild Integration** 8](#_Toc508278895)

[3.1.1 Overview 8](#_Toc508278896)

[3.1.2 Dependencies 8](#_Toc508278897)

[**3.2** **Application server start** 8](#_Toc508278898)

[3.2.1 Overview 8](#_Toc508278899)

[3.2.2 Dependencies 8](#_Toc508278900)

[**3.3** **Recycle the database** 8](#_Toc508278901)

[3.3.1 Overview 8](#_Toc508278902)

[3.3.2 Dependencies 9](#_Toc508278903)

[**3.4** **Ajax Refresh** 9](#_Toc508278904)

[3.4.1 Overview 9](#_Toc508278905)

[3.4.2 Dependencies 9](#_Toc508278906)

[**3.5** **Cppnew.bat script** 9](#_Toc508278907)

[3.5.1 Overview 9](#_Toc508278908)

[3.5.2 Dependencies 10](#_Toc508278909)

[**3.6** **Auto-cpp-exe.sh script** 10](#_Toc508278910)

[3.6.1 Overview 10](#_Toc508278911)

[3.6.2 Dependencies 11](#_Toc508278912)

[**3.7** **Application server stop** 11](#_Toc508278913)

[3.7.1 Overview 11](#_Toc508278914)

[3.7.2 Dependencies 11](#_Toc508278915)

[**3.8** **Recycle the ACT Server** 11](#_Toc508278916)

[3.8.1 Overview 11](#_Toc508278917)

[3.8.2 Dependencies 11](#_Toc508278918)

[**3.9** **Deployment process** 11](#_Toc508278919)

[3.9.1 Overview 11](#_Toc508278920)

[3.9.2 Dependencies 12](#_Toc508278921)

[**3.10** **Test Automation scripts** 12](#_Toc508278922)

[3.10.1 Overview 12](#_Toc508278923)

[3.10.2 Dependencies 12](#_Toc508278924)

[4 Production deployment process 13](#_Toc508278925)

[4.1.1 Overview 13](#_Toc508278926)

[4.1.2 Dependencies 13](#_Toc508278927)

[5 Reference Documents 14](#_Toc508278928)

[6 Deliverables 15](#_Toc508278929)

1. **Project**
   1. **Project Summary**

As we understand from our interactions with Hanover team, OneShield deployment needs to be streamlined with regards to the deployment time, effort and to manage consistency. Hanover requested ValueMomentum (VM) to understand their current process, recommend improvements, integrate with current tool chain and change management process. VM team on the ground gathered information required to recommend OneShield deployment process

**Project Share point 🡪 <<TBD>>**

* 1. **Business Driver**

The business drivers for this Integration are to:

* Better way of managing the release artifact information capture. Reporting of release artifacts will be easier
* A single point triggering process with very less manual intervention with automation of deployed script generation and Jenkins java build
* A versioned approach of generating artifacts once per build and subquesnet builds only perform the delta changes generation will make the build process much faster
* The complete build /deploy process automation using tools will reduce the coordination done between DBA, ECM and Development team done today for different process of DB back up, Server restart and validation of the log file
* Creating an automated smoke testing with multiple products and scenarios will help better validate the build as opposed to a manual smoke test of one product done today
  1. **Business Objective**

As part Continuous Integration and Continuous Delivery, One shield builds needs to be automated.

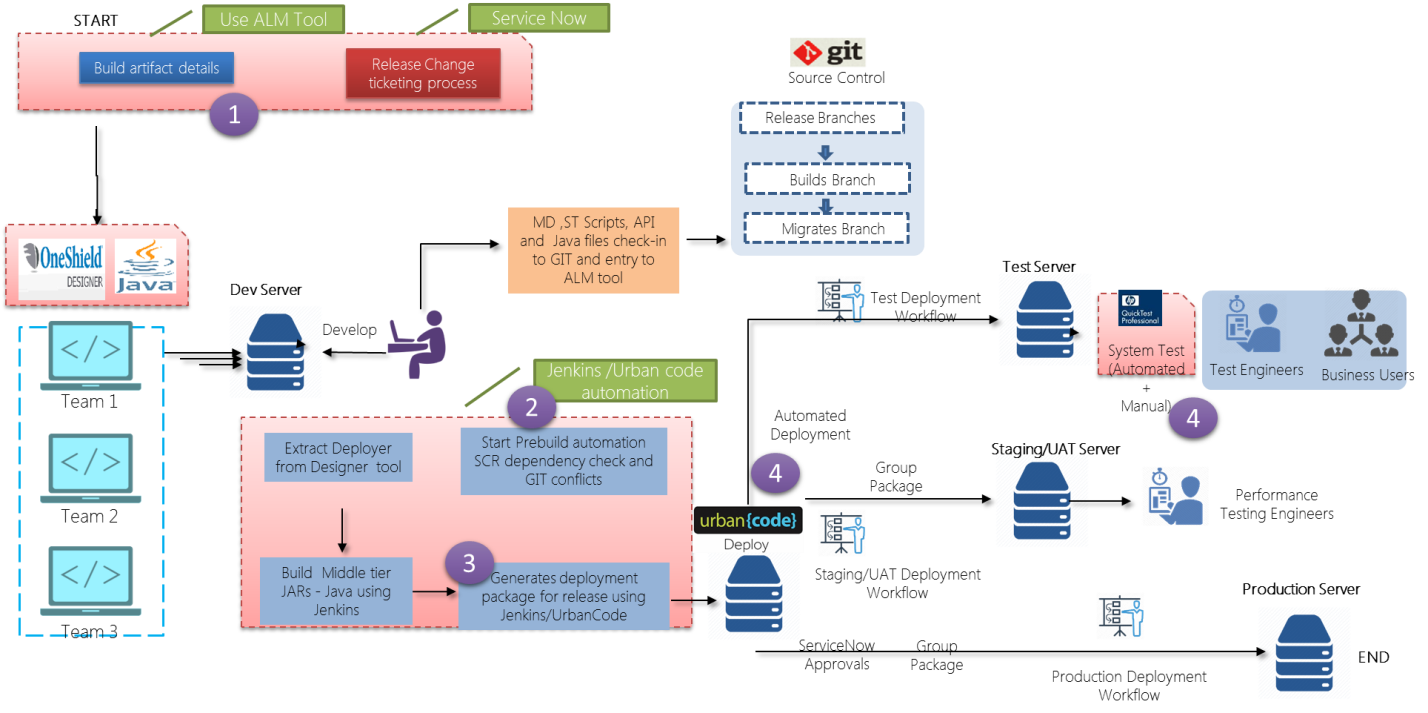
* 1. **In- Scope**

VM team will perform the following services:

* Finalize the build spread sheet for the purpose of capturing build artifacts.
* Execute Prebuild tool updates to execute the build based on data captured in the spread Sheet
* Script to invoke existing pre-built tool
* Move generated OneShield artifacts,  database scripts and compile Java artifacts to a predefined location with agreed upon file naming conversion
* Jenkins scripting for automation of ECM deployment steps
* Invoke pre-existing DB backup & reboot  scripts
* Invoke WAS application server reboots
* Deploy using Urban Deploy
* Invoke Execution of cppbuild.bat script
* Urban code  scripting for  invoking  automated smoke test cases
* Jenkins Script to generate deployer script from Dragon designer.
* Jenkins scripts for middle Java jar build
* Jenkins Orchestration scripts to invoke DB backup, app service restart, execute cppbuild.bat & jar file copy
* Jenkins  Script to transform compile, rules compile & ajax compile
* Urban code deploys tool scripting to execute Oneshield deployment.
  1. **Out of Scope**
* Creation or any fixes needed to current DB backup scripts or WAS application restart scripts.
* Creation of automation test suites.
* Updates or fixes to Oneshield core code for automation generation of  deployer script
* Modifying existing Scripts, we will want Hanover to own the scripts and also make necessary modifications.
* Clean up of cppbuild.bat script to remove references to PVCS
* Modifying Prebuild tool will be made by Hanover

* 1. **Assumptions**
* This is a running document and we will be revisiting whenever there is a change in the requirement
* ACT DB back up Script is available to be called directly.
* ACT application script is available to be called directly.
* Automatic Deployer Script generation script from Dragon Designer is possible. After ACT upgrade there was issues in automatic script generation.   Analysis will be done to fix this and if its determined Oneshield support is needed,   the new build automation will have the  manual step of script generation
* Creation of automated test suite will be done by QA team
* New build process using GIT and new prebuild tool is being done for ACT

1. **Solution Architecture**



The above diagram represents a high-level integration architecture about the Oneshield build and deployment process.

We will be creating a Jenkins CICD pipeline for the OneShield Build and Deployment process.

1. **CICD Pipeline with Jenkins** 
   1. **Prebuild Integration**
      1. Overview

We will be integrating the Prebuild Tool with Jenkins with the help of Prebuild Tool API or with tool URL or through Ant Script. This can be achieved through Jenkins script. Need further analysis on this to determine the best solution to use. Once this step is done we will send email communication to the concerned people about the status of the job.

* + 1. Dependencies

Prebuild Tool Jar file and Ant file

Email address to send the notification

Access to Jenkins

CPP\_Build\_Content spreadsheet

* 1. **Application server start**
     1. Overview

Currently Application Servers (Totally 5 WAS 8.5 servers) are stopping manually before starting the build by logging in to the admin console. We will be integrating these scripts into Jenkins. This step will be executed only above step is completed successfully.

* + 1. Dependencies

Application server start scripts and access

* 1. **Recycle the database**
     1. Overview

Currently database is recycling manually in the respective region by logging into LXPR01ACTBLD and executing the region specific scripts. We will be integrating these scripts into Jenkins. This step will be executed only above step is complete successfully.

* + 1. Dependencies

Recycling scripts cd /db\_builds/oracle\_scripts and ./recycle\_database.sh OSXXX where XXX- region name

Access to LXPR01ACTBLD

* 1. **Ajax Refresh**
     1. Overview

Currently below 2 sql files are coping manually from [\\lxpr01actbld\db\_blds](file:///\\lxpr01actbld\db_blds) to [\\lxpr01actbld\db\_blds\release\_<Release](file:///\\lxpr01actbld\db_blds\release_%3cRelease) branch>.

AJAX\_API.sql

AJAX\_MD.sql

We will be integrating this step with Jenkins.

This step will be executed only above step is complete successfully.

* + 1. Dependencies

Ajax refresh Sql files AJAX\_API.sql and AJAX\_MD.sql

* 1. **Cppnew.bat script**
     1. Overview

This script is executing it manually in windows by passing some arguments which are dynamic . i.e for each release/region, we need to modify the arguments accordingly. While executing this script it is currently prompting for the following questions.

Want to extract PVCS components (Y/N)

Want to extract Metadata (Y/N)

Please run deployer from DD6 UI and then press Y

After completing this batch script Linux console secureCRT will be opend

We will be integrating this step with Jenkins.

* + 1. Dependencies

Cppnew.bat file

If the answers to the above questions remain same, Can the script be updated by removing those questions?

* 1. **Auto-cpp-exe.sh script**
     1. Overview

This script is executing it manually in Linux . This Script expects the following as input

Please enter the region for which you want to run CPP [e.g. - INT, INE, INT, INT, INT, INT, INE]

Please enter the version [e.g.- 200912.12, 201005.33 etc.]

i.e for each release/region, we need to modify the arguments accordingly. While executing this script it is currently prompting for the following questions.

Want to run pre-build check first [Y/N]? Y

After this, System displays the number of errors. If there are any errors then currently build team will inform to the concerned people.

Want to run pre-deployer scripts now [Y/N] ? Y

Want to execute pre-build check again [Y/N]? Y

Want to run meta-deployer scripts [Y/N]? Y

Want to run history script [Y/N]?Y and select the database

Give Y in command prompt

After this It will generate the Rules& Transforms, Generates rules and Transforms, Compilation of rules and finally it displays the number of invalid objects.

Mail will be send to the concerned people if the there are any invalid objects else it will be proceed.

There are some invalid objects that currently ignored by the build team and proceeding further

During Ajax refresh process system is prompting for

Run forms Renumbering Step : Y/N

We will be integrating this step with Jenkins.

* + 1. Dependencies

Auto-cpp-exe.sh script

If the answers to the above questions remain same, Can the script be updated by removing those questions?

Can existing script be updated for Ajax Refesh step to avoid manual intervention?

* 1. **Application server stop**
     1. Overview

Currently Application Servers (Totally 5 WAS 8.5 servers) are starting manually after build and deployment by logging in to the admin console. We will be integrating these scripts into Jenkins. This step will be executed only above step is complete successfully.

* + 1. Dependencies

Application server start scripts and access

* 1. **Recycle the ACT Server**
     1. Overview

Once build and deployment is completed then currently Web Admin team restarts the server once they get email communication from build team .If the above Build step is not completed successfully then this step will not be executed.

* + 1. Dependencies

Restart ACT server scripts

* 1. **Deployment process** 
     1. Overview

We will be promoting the executables to various environments based on the status of the previous steps using Jenkins. Need to discuss further on this.

* + 1. Dependencies

Environment details.

* 1. **Test Automation scripts**
     1. Overview

We will be integrating the automated test scripts with Jenkins. This step will be executed automatically once executables are deployed to the test environment.

* + 1. Dependencies

Automated scripts

Test environment

1. **Production deployment process**
   * 1. Overview

We will be promoting the code to production using Udeploy once the code is tested and completely approved. This process needs more analysis/Discussion to understand the existing process and to come up with a detailed approach.

* + 1. Dependencies

Udeploy access

Production environment details

1. **Reference Documents**
   * 1. Existing System Documentation

This document is prepared based on the existing build and deploy document and after discussing with Subject Matter Experts. This document will be updated further for any change in the requirement.

1. **Deliverables**

|  |  |
| --- | --- |
| **Task** | **Due date** |
| Detailed design document | 16-Mar |
| Workflow Orchestration Creation | 30-Mar |
| Pre build integration with release folder creation | 6-Apr |
| Application Server start and stop scripts integraton | 13-Apr |
| Database recycle scripts integration | 20-Apr |
| Integrating Ajax refresh scripts | 27-Apr |
| Windows- Cppnew.bat script integration | 4-May |
| Linux - Auto-cpp-exe.sh script Integration | 11-May |
| Integrating the ACT server Recycle scripts | 4-May |
| Test automation Integration | 11-May |
| End to End workflow execution / testing | 18-May |
| Urban Code work flow set up With ECM for release env | 18-May |
| Transition to ECM team | 18-May |