OneShield - CI-CD Proposal
Hanover Insurance

Dec 7th 2017

VALUEMOMENTUM

Deliver Value. Drive Momentum℠

Cover Letter

December 7th, 2017

To Hanover Insurance Evaluation Committee,

We are excited to present our proposal for the 'OneShield - CI-CD project' to Hanover Insurance.

It is ValueMomentum's (VMI) goal to continue to build upon an enduring and trusting relationship with Hanover. I assure you of our commitment to work in collaboration with your team and be a value-added partner.

We believe that this candid response demonstrates why we are a strong cultural and technical fit for your initiatives.

Thank you in advance for your consideration.

Sincerely,

Chaitanya Kolli
Account Manager.
ValueMomentum Inc.

VALUE MOMENTUM

Deliver Value. Drive Momentum.

Agenda

- Background & Executive Summary
- Current Process
- Challenges and Pain Points
- Scope & Out of Scope
- Proposed Solution
- Activities Planned
- Deliverables
- Timeline and Resource

Background & Executive 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

This document contains:

- Understanding of current OneShield build process.
- Provides best practices and tools to improve the build process.
- Provides automation options to be considered using the existing CI/CD tools.
- Provide indicative timeline, schedule with resource and cost.

Executive Summary

ValueMomentum thanks Hanover leadership for inclusion in your evaluation process!

Our understanding of Hanover's business processes and systems gathered from our years of servicing you, coupled with our expertise in executing OneShield implementations and CI/CD tools, enabled our integration SMEs to propose implementation practices that best fit Hanover requirements.

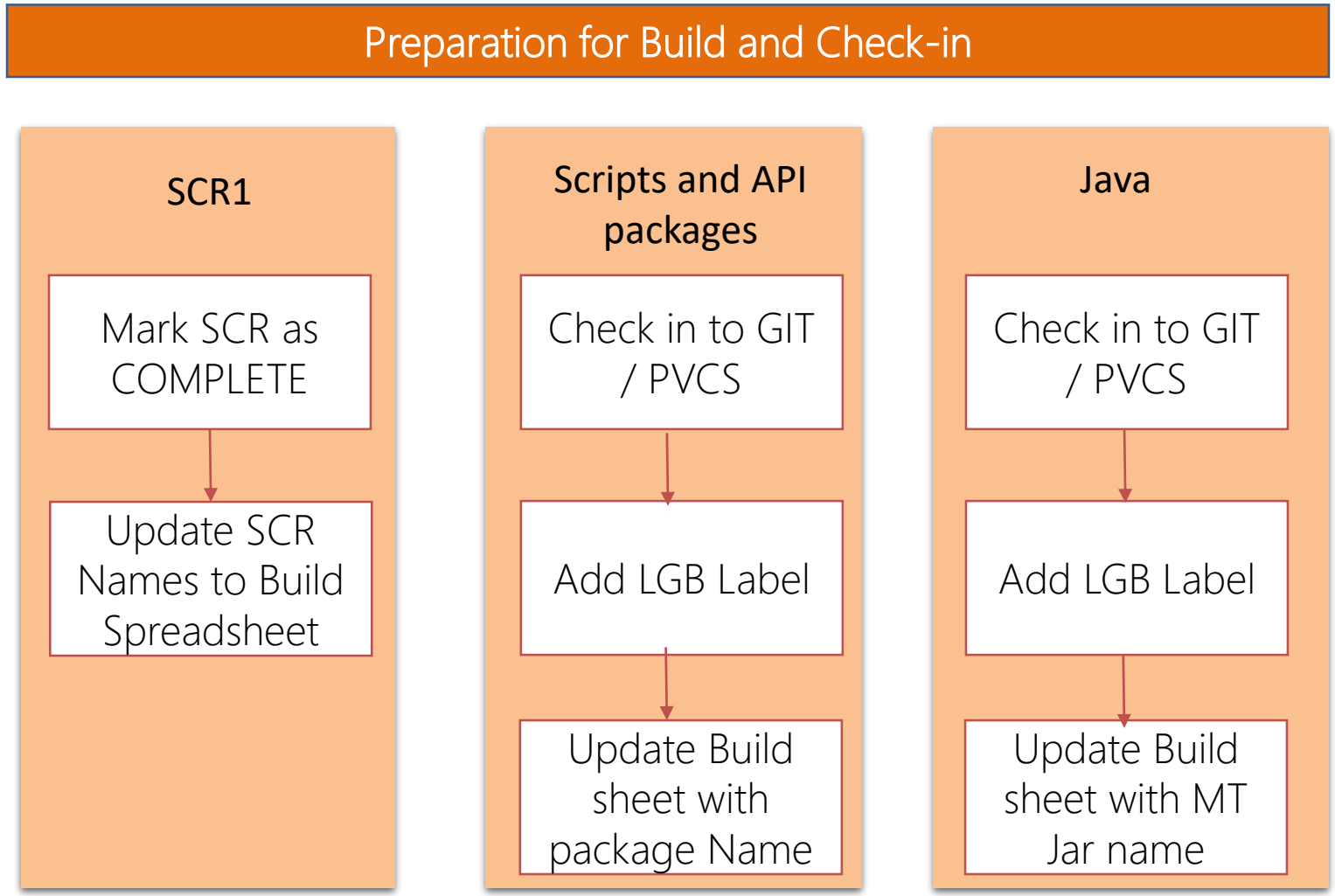
We look forward to our continued discussions as you work towards your final sourcing decision.

Proposal Highlights

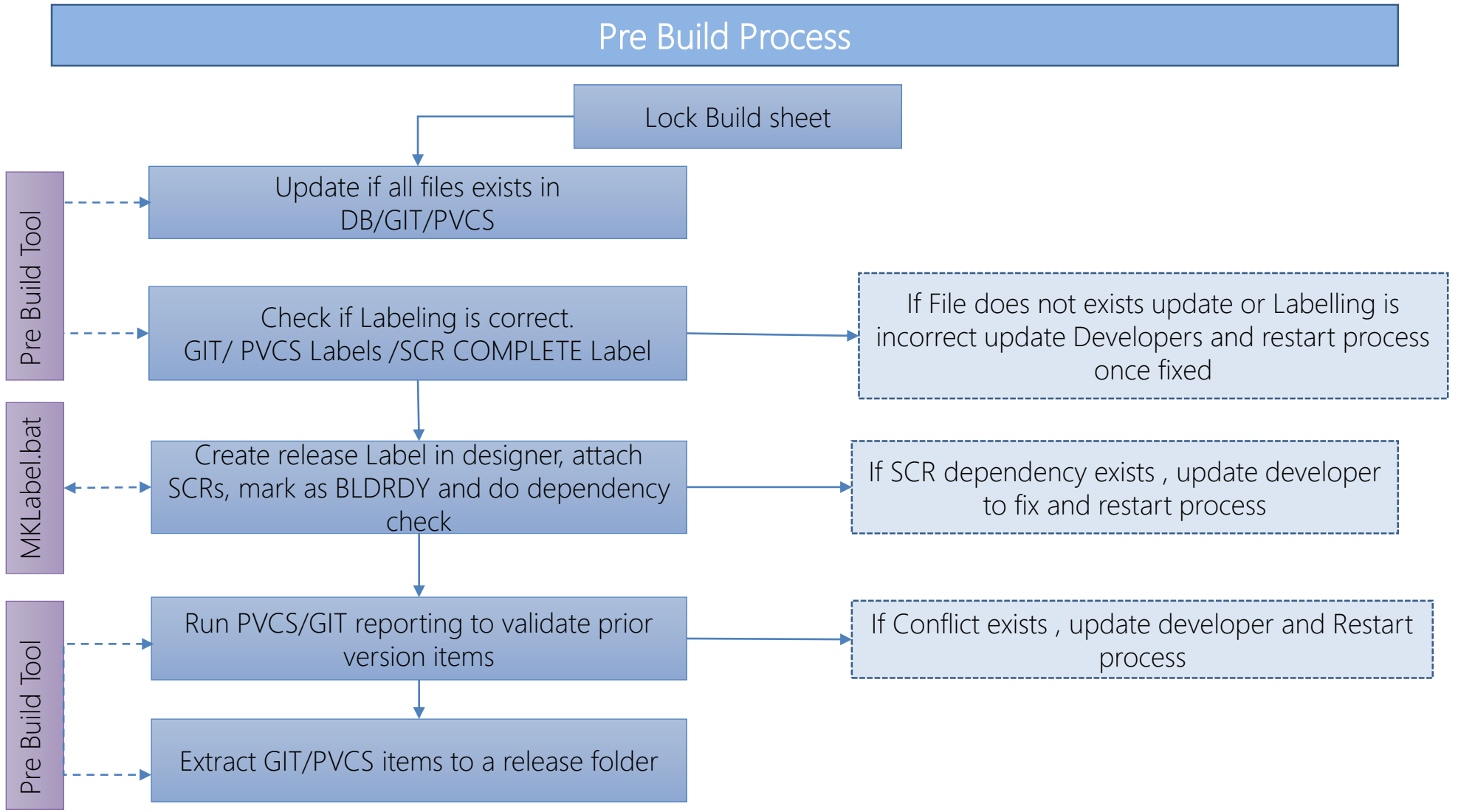
- Reusable & Repeatable Process leveraging existing Hanover toolset; clear indication of process/implementation timeline improvements

Our Understanding : Step 1 : Current Process…

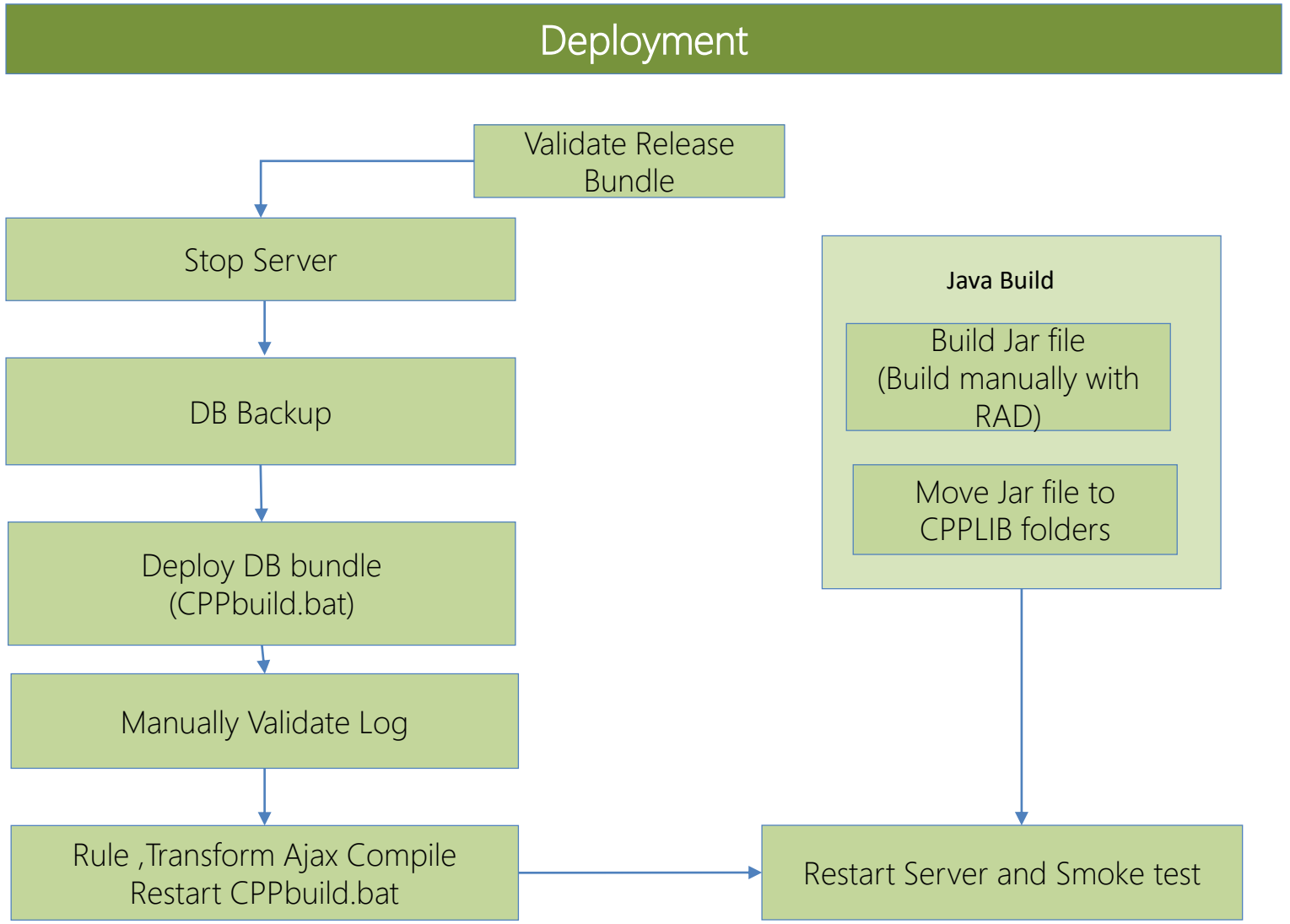
High level Hanover OneShield build and deployment process



Our Understanding : Step 2 : Current Process…



Our Understanding : Step 3 : Current Process…



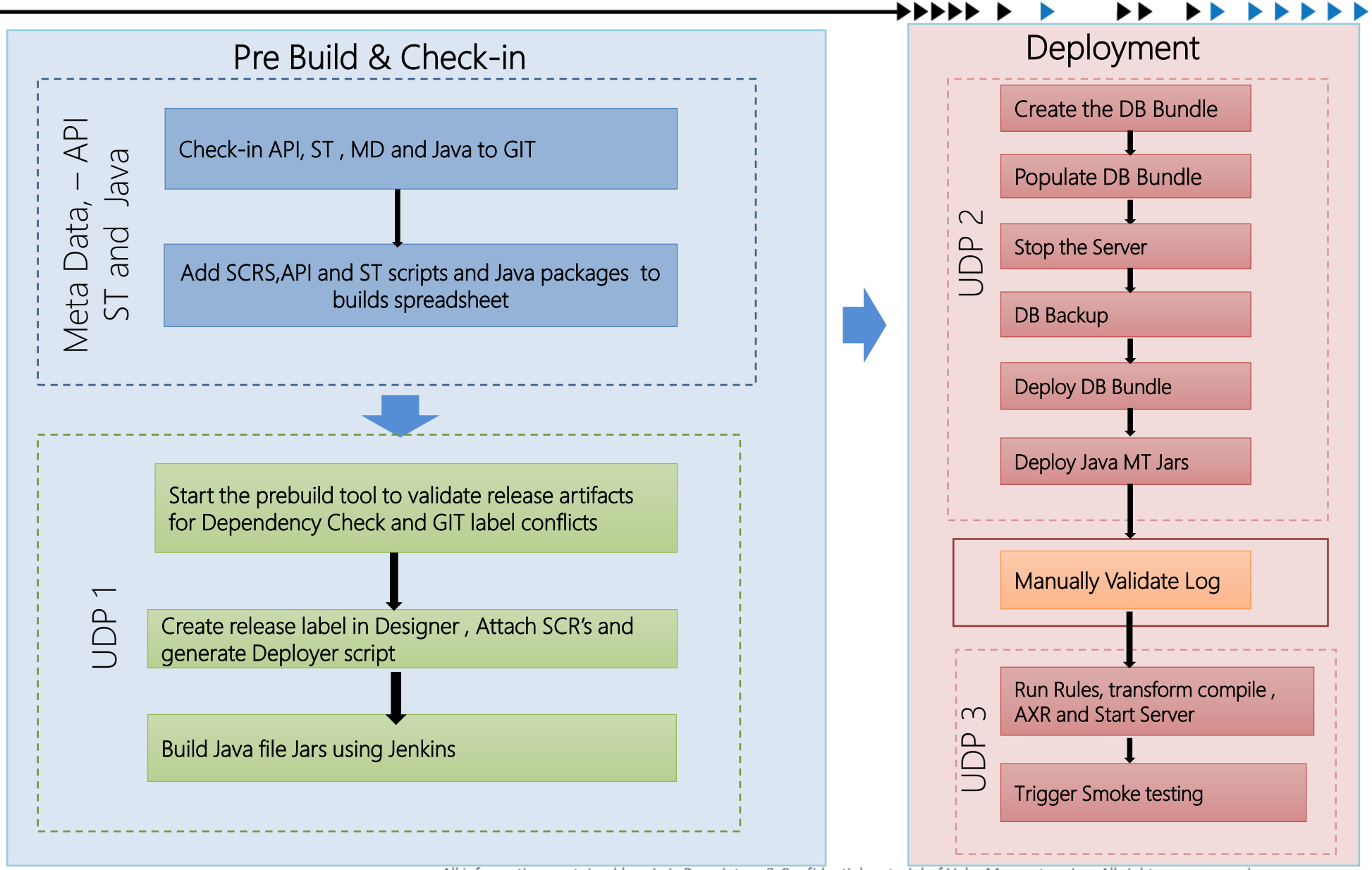
Challenges and Pain Points

	Preparation	Pre Build Process	Deployment
Time	30 mins-1 hr	1-2 hrs	1-2 hrs
Roles	Developers	Pre Build Team	ECM Build Team/DBA
Tools and Technology	Dragon Designer PVCS/GIT Excel	Dragon Designer Pre Build Tool	Batch script WAS console DBA
Challenges	• Developers have to ensure that all SCRs/artifacts are updated in the build sheet. Any missed items will require a full build restart if there is dependency.	• Pre Build tool has multiple steps automated , but multiple points of manual intervention for validation. • Manual Extraction of deployment scripts from the Designer adds more delay to this step.	• Batch script used by ECM team for deployment is complicated with multiple steps and was developed as part of initial implementation. • Some of the process in the script might not be relevant anymore. • DBA involvement for DB backup and restarts.

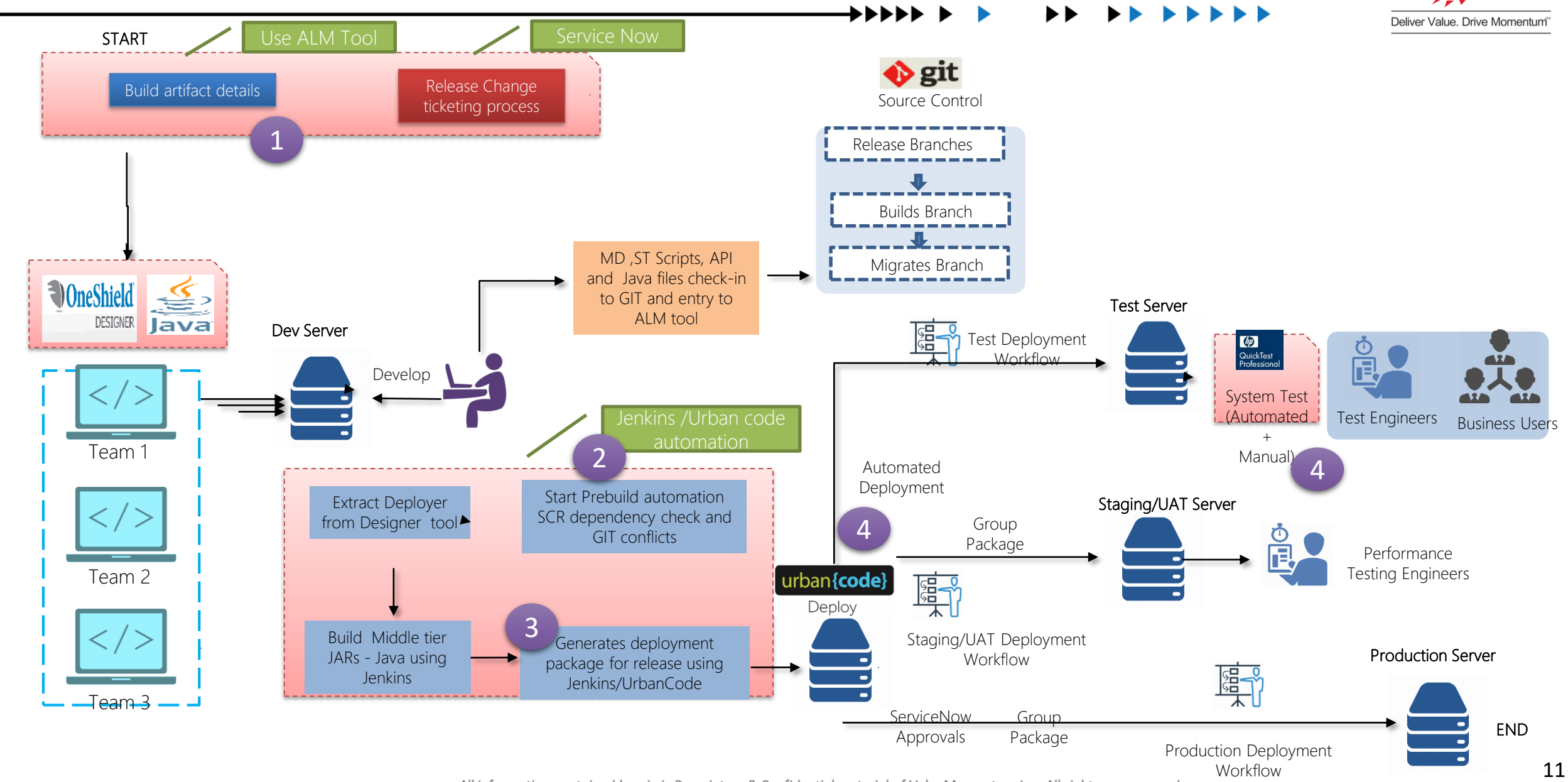
Scope

- Migrate the Build spread sheet to Jira to capture build artifacts.
- Prebuild tool updates to execute the build based on data captured using Jira .
- Jenkins/UrbanCode scripting automation for prebuild process, triggering of existing pre build tool, generation of build artifacts into a release folder and Java build
- Jenkins/urban code scripting for automation of deployment steps that includes DB back up, application server restarts, Jar file deployment and execution of cppbuild.bat script for deployment.
- Clean up of cppbuild.bat script to remove unwanted logic
- Invoke automated smoke test cases

Proposed Solution - CI/CD Pipeline with Jenkins



Proposed Solution



Benefits

Section	Task	Benefits	Apprx Improvement time
1	Migrate to ALM tool from Build spread sheet	Better way of managing the release artifact information capture . Reporting of release artifacts will be easier	5 -10 mins
2	Prebuild steps Automation with Jenkins or Urban code	A single point triggering process with very less manual intervention with automation of deployed script generation and Jenkins java build	10- 20 mins
3	Process change to generate artifacts once and use it for deploy of further builds to environments	This will provide significant improvement area as the major time in pre buid build is taken for generation and creation of bundle today. 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	30 mins – 1 hr
4	Automation of the release using urban code deploy and Urban code release	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	30 mins – 1 hr
5	Automated smoke test	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	5-10 mins. (This will help more in identfying regression issues from build)

Additional improvements to consider

- DB Tuning and frequent data base re-orgs[re-build the indexes] to improves overall database performance.
- Clean up unwanted rules from the dragon to improve the Rules compilation execution time.
- Automation for unit testing [PLSQL and JAVA changes].
- Code promotion to production environment using Urban code implementation.
- Execute automated smoke test and regression test suite after each build
- Communicate each team changes implemented in that week and discuss the dependencies
- Identify the code components from each team and resolve the dependencies one day before assembly build
- Always verify whether you are making changes on the latest version of meta data or not.
    - By verifying the table history
    - By Leveraging the code comparison tool

Activities Planned

Category	Task Description	Primary	Assist	Validate
Discover and Plan	Analysis of Build Spreadsheet and feasibility for migration	VMI	HAN	HAN
	Analysis of PreBuild Tool	VMI	HAN	HAN
	Analysis of CPPbuild.bat	VMI	HAN	HAN
	Analysis the current Handover Urban Code and Jenkins Release Process	VMI	HAN	HAN
Design	Migration of Build Spreadsheet to new tool	VMI	HAN	HAN
	Design Automation of PreBuild Tool process with Jenkins	VMI	HAN	HAN
	Design Automation of DB administration and WebSphere administration tasks	VMI	HAN	HAN
	Design Regression test suite for Oneshield Application	HAN	VMI	HAN
	Design End to End flow Automation with Urban Code/Jenkins	VMI	HAN	HAN
	Design Urban code release with Service Now Integrations	VMI	HAN	HAN
Implement	Create Template in Jira, design the task and action	VMI	HAN	HAN
	Create Workflow to call PreBuild Tool with Jenkins	VMI	HAN	HAN
	Create Workflow to create release label in Designer and attach SCR	VMI	HAN	HAN
	Create workflow to generate deployer script from Designer	VMI	HAN	HAN
	Create workflow for extract release artifacts from GIT	VMI	HAN	HAN
	Create Jenkins build script for java projects	VMI	HAN	HAN
	Create workflow for packaging the release artifacts	VMI	HAN	HAN
	Create workflow CI management task for Deployment and server readiness	VMI	HAN	HAN
	Create workflow for Rules, transform and ajax compile	VMI	HAN	HAN
	Create workflow for Regression test invocation	VMI	HAN	HAN
Monitor	Monitoring all of Deployment builds of Oneshield Application in DEV	VMI	HAN	HAN
	Monitoring all of Deployment builds of Oneshield Application in UAT	VMI	HAN	HAN
	Monitoring all of Deployment builds of Oneshield Application in PROD	VMI	HAN	HAN

Deliverables

- Modify Jeri to accommodate SCR number & artifacts
- Update existing scripts to invoke Prebuild tool
- Script to generate deployer script from Dragon designer
- Jenkins scripts for middle Java jar build
- Orchestration scripts to invoke DB backup, app service restart, execute cppbuild.bat & jar file copy
- Script to transform compile, rules compile & ajax compile
- Scripts to invoke automated smoke test

Timeline, Resource and Dependencies

Indicative Timeline

Tasks / Weeks	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12
Discover and Plan												
Design												
Implement												
Monitor												

Resource details

1. One DevOps expert resource will be deployed at Hanover office.
2. This will be T&M engagement
3. The hourly rate would be $125 + expenses

Assumptions/ Dependencies

1. Hanover will provide necessary infrastructure, machine(Laptops) and network access
2. Hanover will provide an access to the required stakeholders and documentations for assessment of various applications

THANK YOU!