IBM Software Group

ALM NonUCM and Buildforge demo
April 18, 2008

Stuart Poulin stuart.poulin@us.ibm.com **206-686-9824**

Rational software

Disclaimer and Non-Disclosure

- IBM Rational Plans are subject to change

- Nothing in this document is intended to create any representations or warranties

- IBM Rational warranties are contained in the applicable license agreements

Demo to Present

▶ NonUCM – any system that is not UCM – i.e. Base ClearCase, Subversion, CMVC,…

▶ Methods to utilize ALM in such build environments.

▶ ALM Baseline and Build tools

▶ Describe ALMBaseline and BTBuild (Build) records

▶ Describe tie into ALMActivities

▶ Describe reference to ALMTasks and Requests

▶ Demo an example of how BuildForge can be used to drive and populate ALM baseline and build data
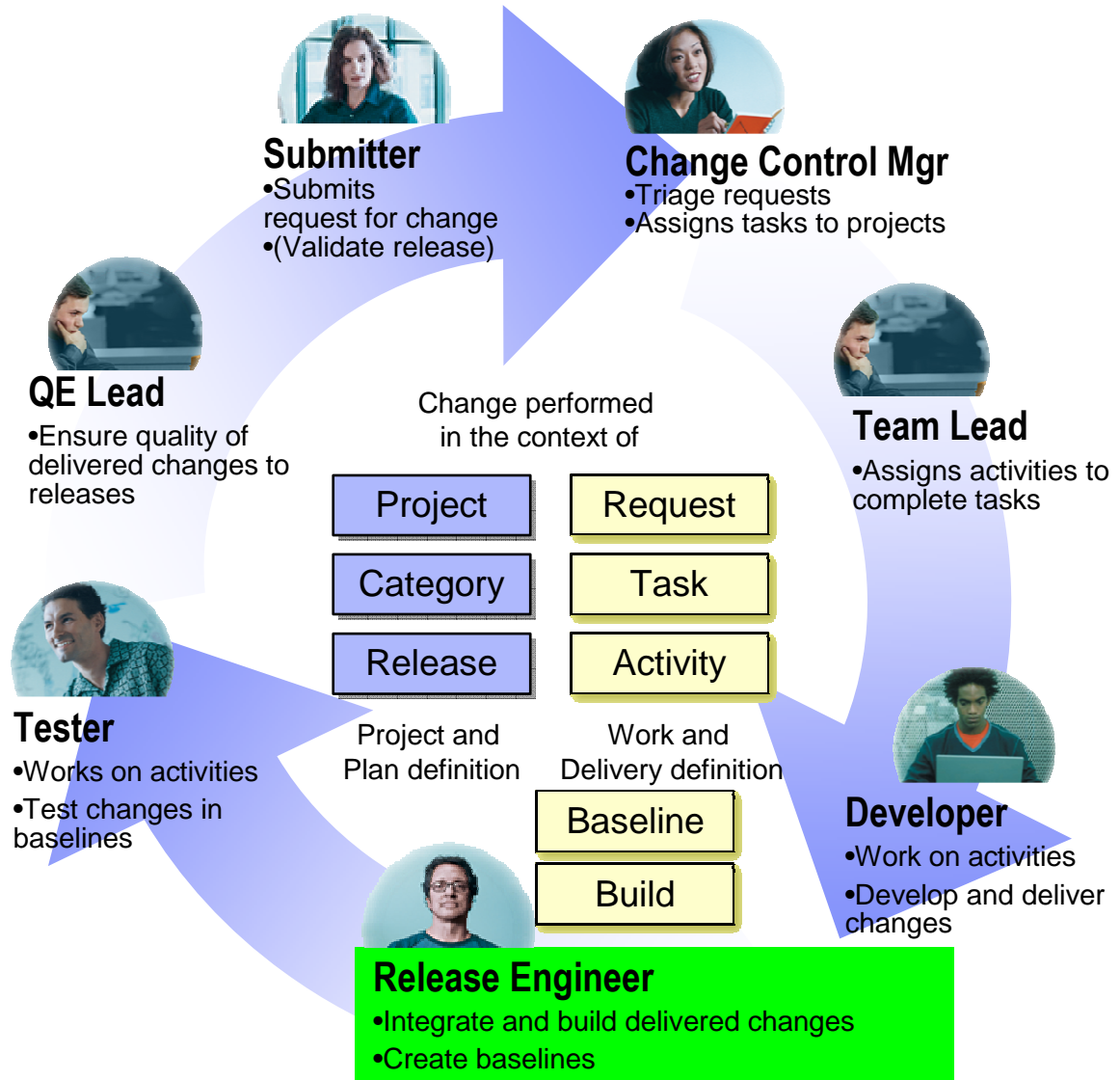
Prerequisites

- View a CQ ALM demo explaining the overall high level ALM work flow and features.

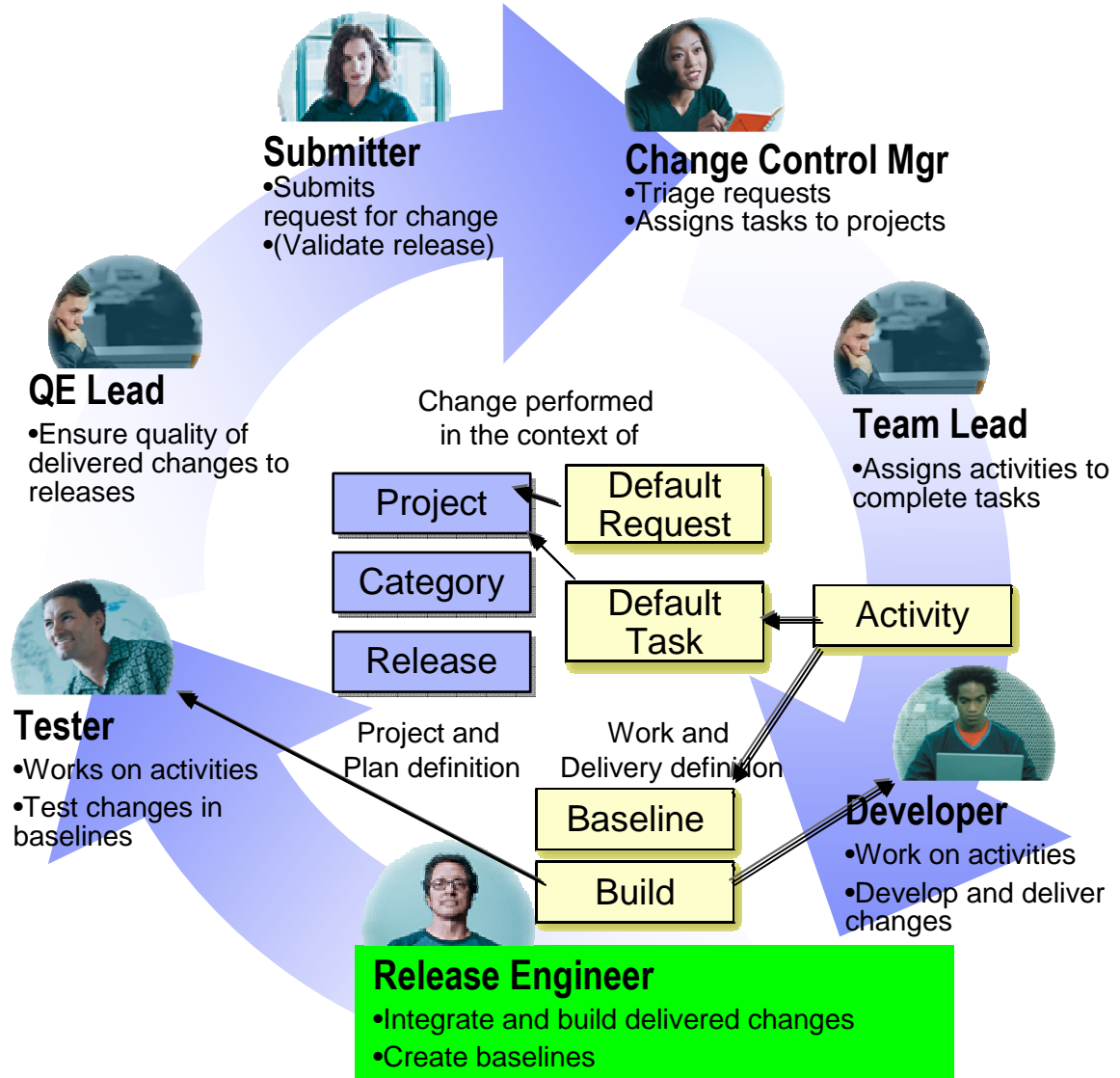- ALM UCM demo - to compare against NonUCM usage

What is CQ-ALM?
Brief introduction

- Out-of-box ALM Solution for ClearQuest (schema and packages)

- Process model focused on Requests being submitted for a Release of a Project

- Requests are planned as Tasks and performed as Activities

- Workload can be balanced across Phases and Iterations of a Project

- Each project team can customize the configuration

- Baseline and Build records track progress of activities

Submitter
- Submits request for change
- (Validate release)

Change Control Mgr
- Triage requests
- Assigns tasks to projects

QE Lead
- Ensure quality of delivered changes to releases

Team Lead
- Assigns activities to complete tasks

Change performed in the context of

Project	Request
Category	Task
Release	Activity

Project and Plan definition

Work and Delivery definition

Tester
- Works on activities
- Test changes in baselines

Baseline
Build

Developer
- Work on activities
- Develop and deliver changes

Release Engineer
- Integrate and build delivered changes
- Create baselines

Agile Project

- ALMProject assigns a Default ALMRequest and Default ALMTask

- Developers create ALMActivities against the default ALMTask

- Baseline and Build records track progress of activities

- Developers and Testers can Verify ALMActivities in Builds

Submitter
•Submits request for change
•(Validate release)

Change Control Mgr
•Triage requests
•Assigns tasks to projects

QE Lead
•Ensure quality of delivered changes to releases

Team Lead
•Assigns activities to complete tasks

Change performed in the context of

Project	Default Request
Category	Default Task
Release	

Tester
•Works on activities
•Test changes in baselines

Project and Plan definition

Work and Delivery definition

Activity

Developer
•Work on activities
•Develop and deliver changes

Baseline

Build

Release Engineer
•Integrate and build delivered changes
•Create baselines

Baseline

1. **Choose Project**
2. **Identify UCM stream**
3. **Identify PVOB or Location**

4. **List completed activities (Diff Baselines)**

Automate the creation of the Baseline and Build records as part of your build process

ALM Baselines provide instant knowledge of Status, Disposition, and location of Activity artifacts in UCM and non-UCM baselines and builds.

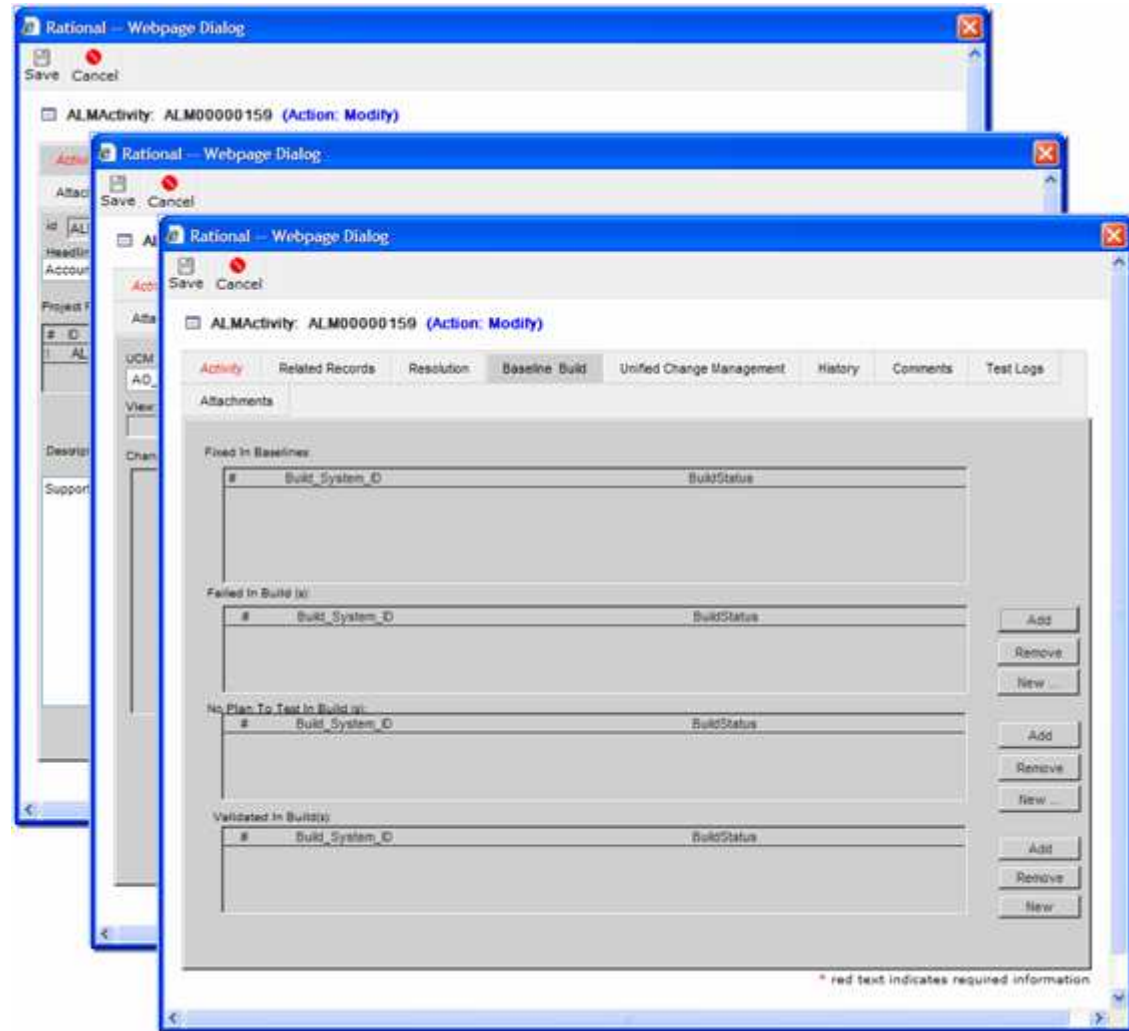# Activities can be tracked in Baselines

Contributors

Scenarios

- Assign, perform and deliver work activities

- Monitor builds and quality

- Collaborate

UI / Forms

- Task types
- Activities
- Task actions

Baseline/Build scripts provided with ALM

- Designed to interface UCM and NonUCM Change Management (CM) systems to ALM

- Written in RATLPerl. Released as source to allow you to extend for your own process if needed

- All explained in ALM documentation and have a commandline –help for addition information.

Baseline/Build scripts provided with ALM

- ## Scripts supplied and presented in this demo

- ## Create_baseline_record.pl

  ▸ Creates an ALMBaseline record from the commandline

- ## Create_build_record.pl

  ▸ Given an ALMBaseline record creates a BTBuild record off it

- ## Update_baseline_record.pl

  ▸ Updates an existing ALMBaseline record from UCM or supplied commandline

- ## Update_build_record.pl

  ▸ Update an existing BTBuild record from the commandline

- ## Runbasicquery.pl

  ▸ Runs a CQ stored query given commandline params for filters and saves the output to a file.

BTBuilds are created off Baselines

(1) **Choose Project**

(2) **Set Build Type and Build Status**

(3) **Choose Baseline**

Create (BTBuild) ALM00000503

Main | Build Details | ALM | History

☐ Obsolete?

ALMProject :   [Add] [Remove]

id	Category	Release	Is Obsolete?
ALM00000407	Category_RWM	ReleaseLabel01...	No

ALMBuildType: [Platform ▼]   ALMSecurityPolicy: [SecurityPolicy_RWM ▼]

ALMBuildStatus: [Passed ▼]   ALMOwner: [ReleaseEngineer_RWM ▼]

ALMRoles:

Project	RoleLabel	Primary	
ALM00000...	ReleaseEngi...	Releas...	

[Add] [Remove]

ALMBaseline: [Add] [Remove]

PVOB_OrLocation	Name	Obsolete	Owner	Project's Category	Project's Relea
609PVOBorLOC	609Baseline	No	ReleaseEngi...	Category_RWM	ReleaseLabel0

Automate the creation of the Baseline and Build records as part of your build process

Animation

04_Project_ToDoNonUCM_Component

▸ Create Complete Activities

▸ Start Build, Review steps

▸ UCM is not used in this example

▸ Create_baseline_record options change:

- "Location" needs to be passed since there is no UCM PVOB,
  – Example: This could be repository for your CM system, or a directory
- Option to pass Location is –pvob <location>
  – Field on ALMBaseline record is "PVOB or Location"

▸ Adding Activities to ALMBaseline record

- Since there is no "UCM diffbl" a query is run to find ALMActivities that are completed since the last ALMBaseline was created
- Create_baseline_record has added option "-from" to allow activities to be read from a file and placed on the baseline record.

▸ Create_build_record remains the same

When to create Baseline and Build records

- Your choice on when to do a source build, create an ALMBaseline, and ALMBuild

- Not every CM Label needs an ALMBaseline

  ‣ Only "significant" labels need be tracked

- For example: You may want to not create baseline or build records unless source builds successfully off a label