Enabling reuse of assets across the globe using RSA and RAM

Bhawana Gupta
bhawana.gupta@in.ibm.com

Innovate2010
The Rational Software Conference

Let's build a smarter planet

The premiere software and product delivery event.
June 6–10 Orlando, Florida

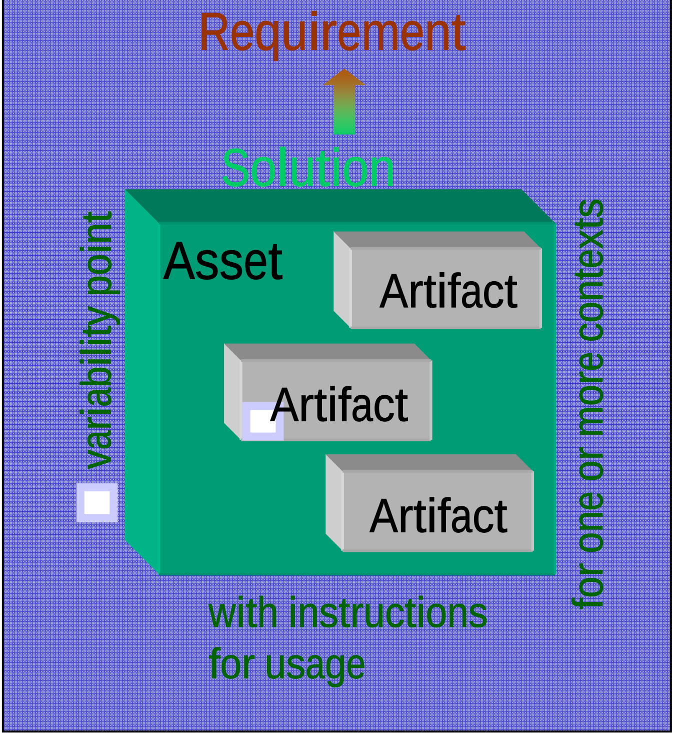
Table of Contents

- Introduction to Asset Management
- Challenges faced during Asset Management
- What does one need for Asset Management
- What do we recommend?

What is an Asset?

- An Asset is
  - a collection of Artifacts
  - which provides a solution to a requirement or business need
    - for one or more contexts
    - with instructions for usage
    - and variability points
- What are Artifacts?
  - Workproducts from the software process
    - Requirements, Designs, Models, XML, XSD, Source code, Data, Tests, User Interfaces, Documentation, Service Interfaces, Estimates and so on…
- Kinds of assets
  - Applications, components, business process, information model, patterns, services, frameworks, templates, …

Rational Asset Manager Asset Definition



Asset management helps address delivery challenges

Asset warehousing is critical to success

- Increase quality from reuse of well-tested assets
- Increase the ROI on assets
- Lower cost from asset reuse
- Lower risk from reuse of proven assets

> ***Software asset management is all of the infrastructure and processes necessary for the effective management, control and protection of the software assets within an organization, throughout all stages of their lifecycle***

Benefits of managing assets of an enterprise

Enterprise Architects

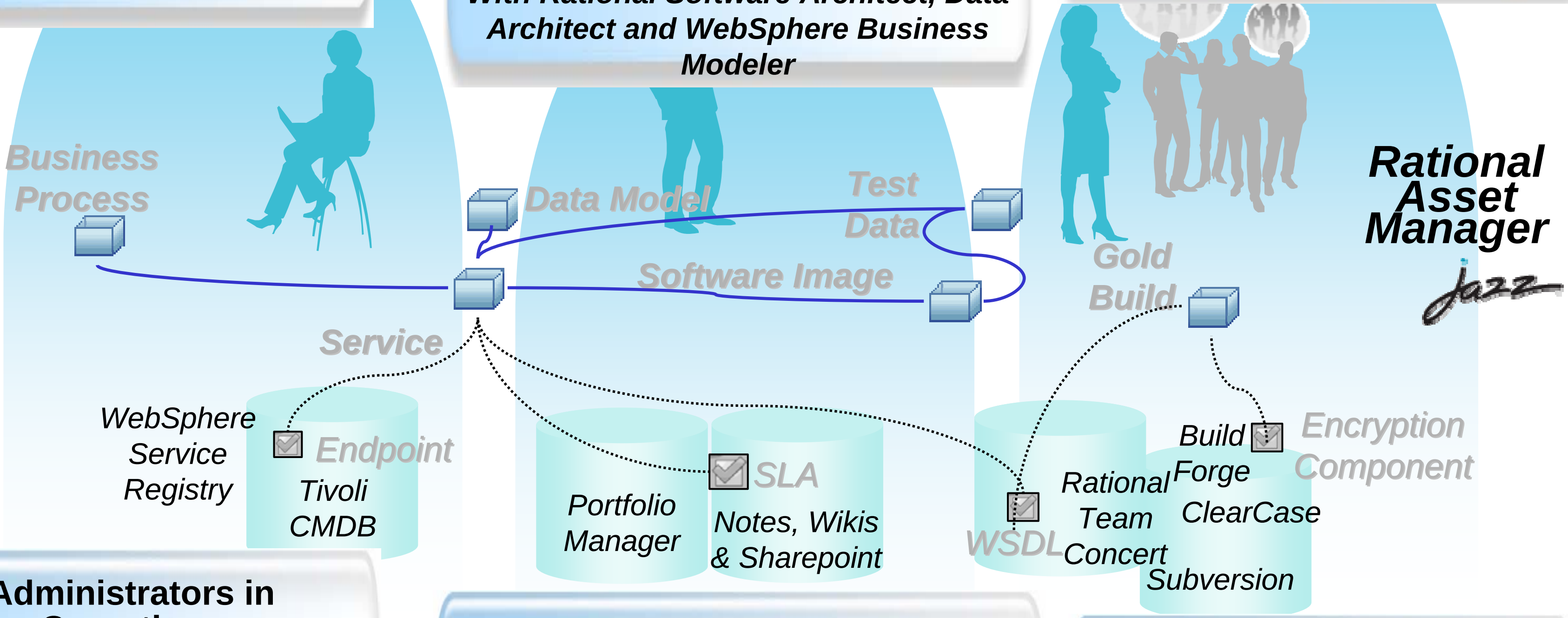
To reduce risks, automate reviews and EA adoption ***With System Architect***

Architects

To reduce costs by sharing data models, business processes & proven reference architectures ***With Rational Software Architect, Data Architect and WebSphere Business Modeler***

Testers

To reduce defects by sharing the correct test data & virtual environments ***With Rational Testing Solutions***



Administrators in Operations

To reduce outages by understanding the impact of operational change on business & technical assets ***with Tivoli CCMDB & WebSphere***

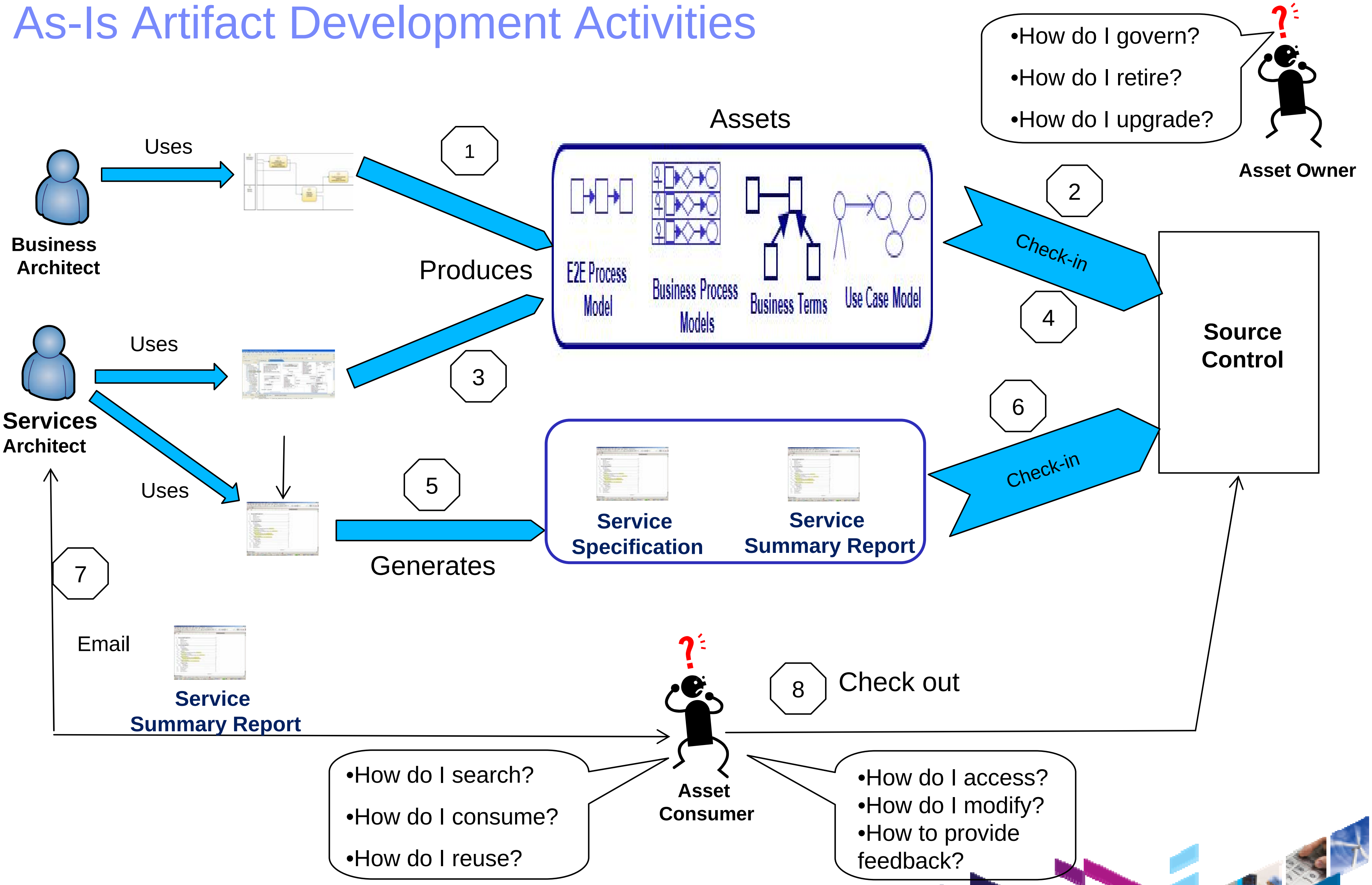
Business People

To save time finding solutions in docs like Viso, Word, PDF ***in Sharepoint and Wikis or*** *assets that meet* ***requirements in Req Pro & Doors***

Developers

To reduce costs and legal risks by re-using only approved components ***With Eclipse, Build Forge & Rational Team Concert***

As-Is Artifact Development Activities

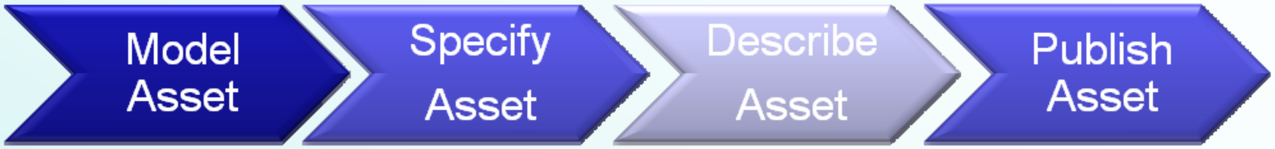


Challenges

- The **ability to easily describe and discover assets** (service model, business process model, use case model, service spec) using the metadata is severely restricted.
- In order for consumers to use assets or make modification requests, there is a sequence of email interactions between asset consumers and asset producers; because email is used in this way, the audit trail and later **discovery and closure of these requests is very difficult**;
- In addition, **understanding the impact of change to dependent assets** is almost impossible. That is determining the impact to changing Account Manager service is not feasible as there is no metadata describing the dependency relationships is captured in Word.
- This kind of asset discovery is **time-consuming and elongates development-turn-around time** for building applications from these assets.
- The **Owners of these Assets have no visibility into the usage of these assets** so that they can be retired or enhanced as well as for making investment decisions

What do we want to do:

- Asset Creation



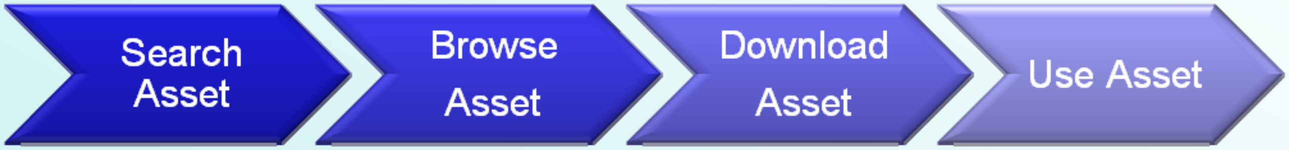
Model Asset

Specify Asset

Describe Asset

Publish Asset

- Asset Consumption



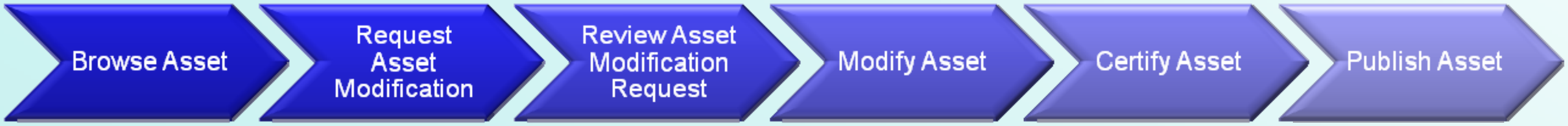
Search Asset

Browse Asset

Download Asset

Use Asset

- Asset Governance



Browse Asset

Request Asset Modification

Review Asset Modification Request

Modify Asset

Certify Asset

Publish Asset

Challenge: Unknown location, usage and duplication of software components, frameworks and collaborative assets

Scenario: Financial Institution

- A financial institution with multiple lines of businesses, is challenged with managing distributed software components across their distributed environments
- Though each line of businesses (LOBs) are managed in silos, many software components that implement common business processes are related
- Assets are not only inaccessible across the LOBs but are so within LOBs themselves

Governed

“How do I prevent duplicate efforts?”

“How do my multiple lines of businesses know of the standards existence?”

“How do I ensure compliance with industry and engineering standards across all LOBs?”

“How can asset collaboration be achieved to break through the LOB barriers?”

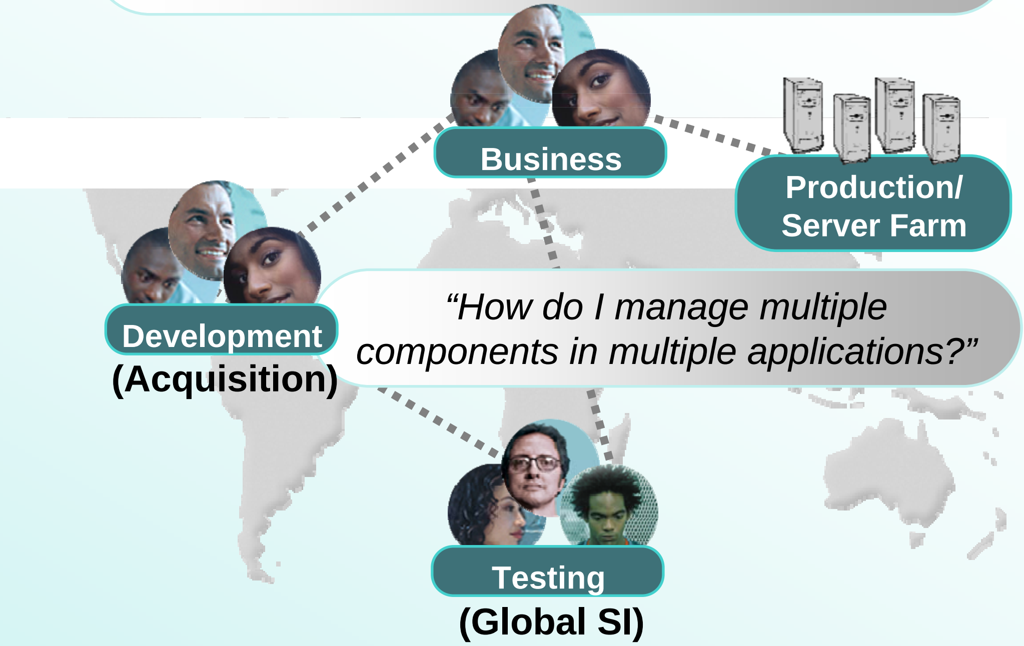
Challenge: Govern software assets

Scenario: Financial Institution

- For a multinational bank managing hundreds of branches worldwide, and managing a global ATM network, a security audit determined that select Web applications posed a "backdoor" risk
  - Old versions of code and pieces of applications made it to production and applications exposed because of changes in common components put live applications at risk
- In addition, the audit took over 500 person hours, found no centralized audit records, making it difficult to do forensic analysis – what, who, when, where, and also found inconsistent approvals/checkpoints

Governed

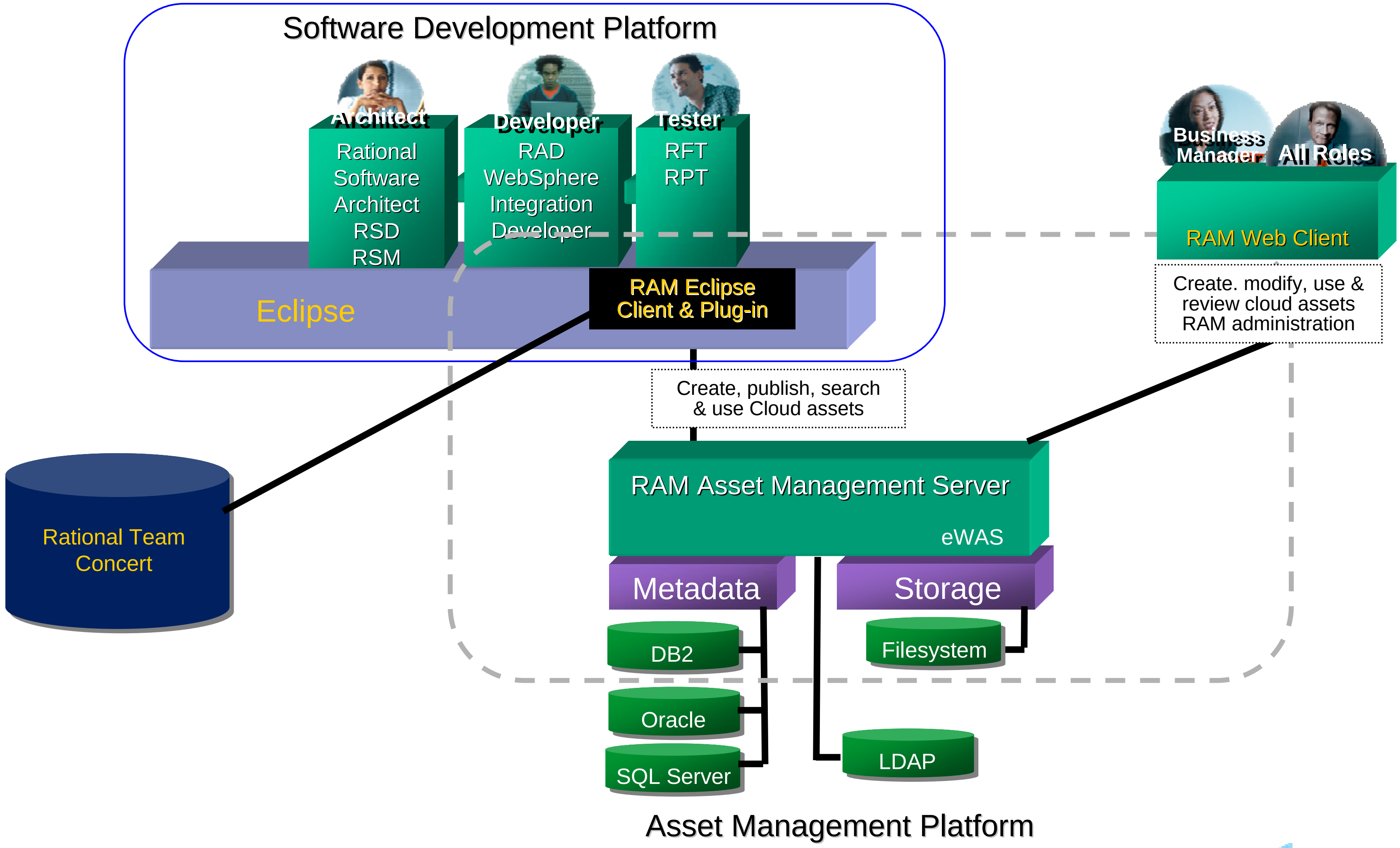
"How do I prevent the team from deploying multiple versions of the same application?"



"How do I manage multiple components in multiple applications?"

"How do I ensure the security of my assets, and enforce approvals at each checkpoint?"

What do we recommend?



Software Development Platform

Architect
Rational Software Architect
RSD
RSM

Developer
RAD
WebSphere Integration Developer

Tester
RFT
RPT

Eclipse

RAM Eclipse Client & Plug-in

Business Manager

All Roles

RAM Web Client

Create. modify, use & review cloud assets
RAM administration

Create, publish, search & use Cloud assets

Rational Team Concert

RAM Asset Management Server
eWAS

Metadata

Storage

DB2

Filesystem

Oracle

SQL Server

LDAP

Asset Management Platform

Rational Asset Manager

Scalable, Flexible, Enterprise-Level Solution

Store and Organize

Repository to store and quickly find assets & services

- Reduce maintenance costs
- Improve consistency

Asset Lifecycles

Automate reviews and business and technical policy enforcement

- Eliminate costs from difficult to enforce policies

Impact Assessment

Capture asset relationships, plan change and understand impact

- Prevent changes that result in unexpected costs

Audit and Report

Identify value add or non- performing assets

- Allocate resources to optimize ROI
- Reduce compliance & duplication costs

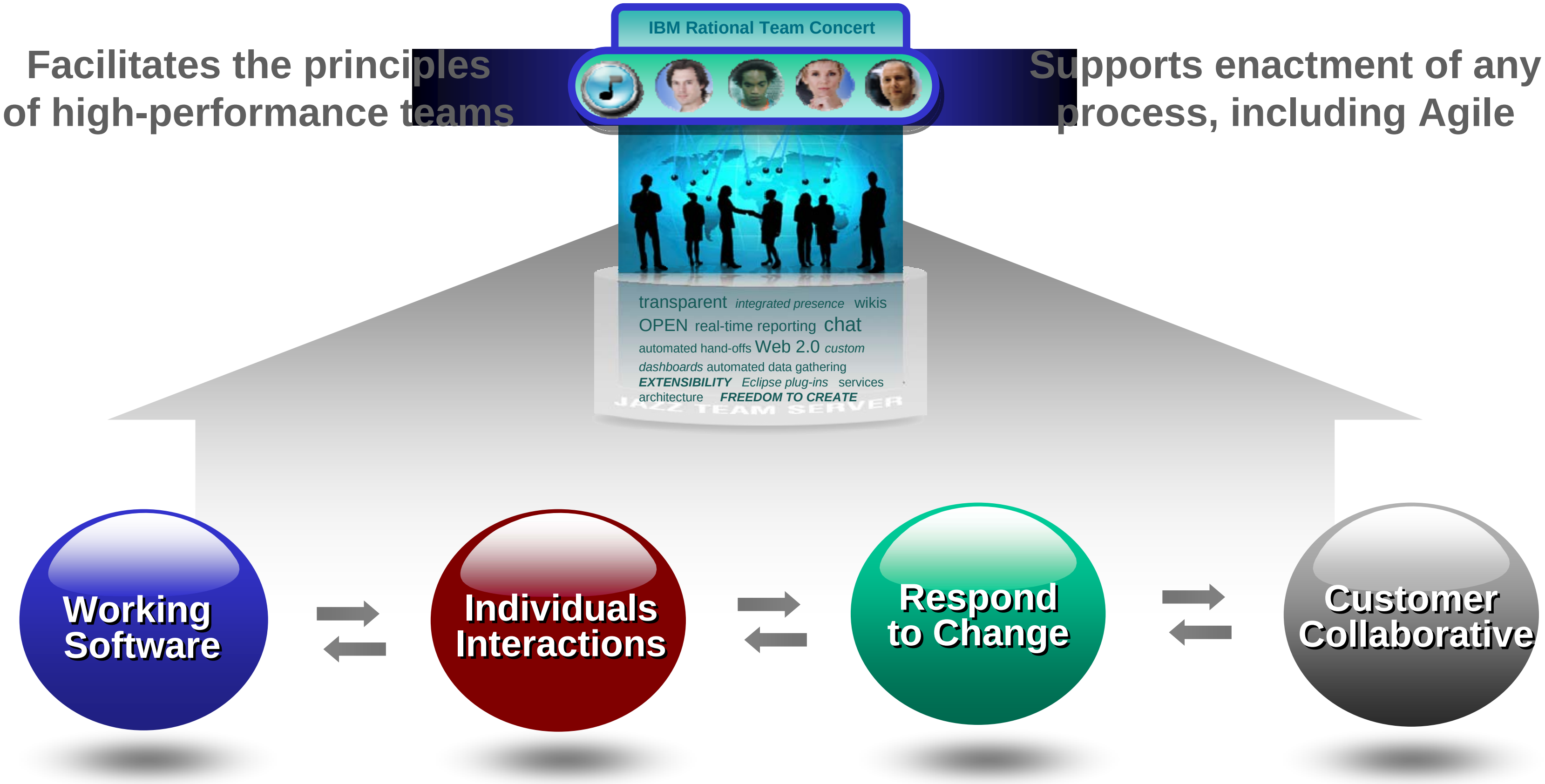
Communicate

Communicate asset information

- Subscription, Notification, Feeds, Forums
- Reduce cost with efficient communication

Integrate with your design and operational environment including other existing repositories

Rational Team Concert



Facilitates the principles of high-performance teams

Supports enactment of any process, including Agile

IBM Rational Team Concert
transparent *integrated presence* wikis
OPEN real-time reporting chat
automated hand-offs Web 2.0 *custom dashboards* automated data gathering
EXTENSIBILITY *Eclipse plug-ins* services architecture ***FREEDOM TO CREATE***
JAZZ TEAM SERVER

Working Software ⇄ Individuals Interactions ⇄ Respond to Change ⇄ Customer Collaborative

- Continuous integration
- Manage team assets
- Change driven
- Integrated / traceable

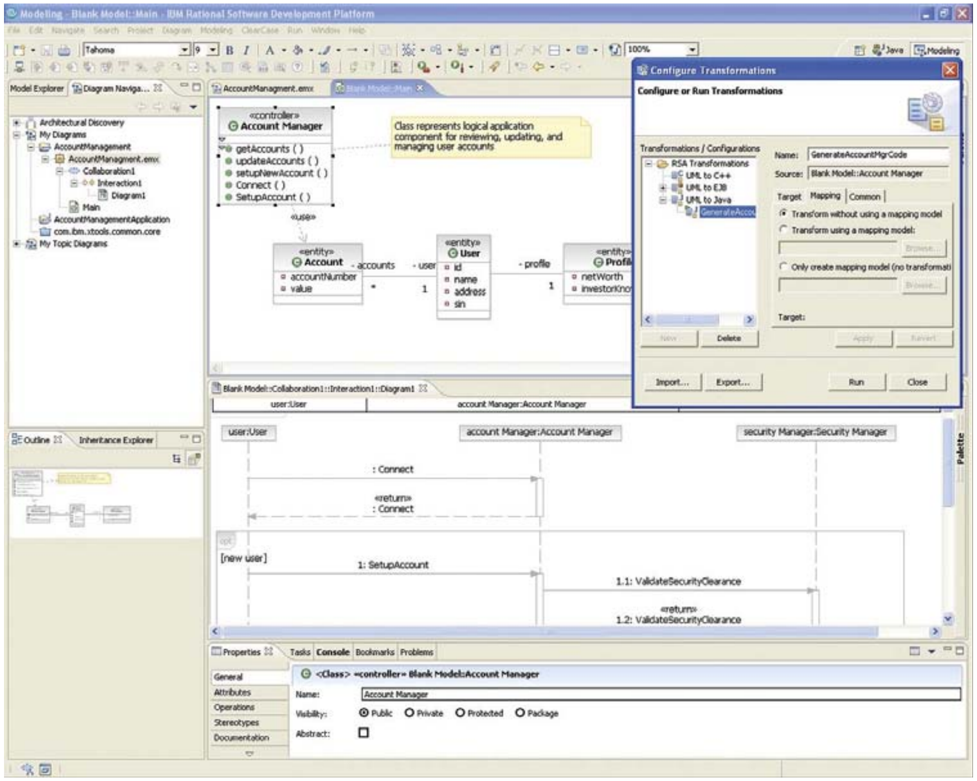
- Starting ad-hoc teams
- Team awareness
- Process awareness
- Ad-hoc sharing

- Process flexibility
- Iterative plan-execution
- Multiple releases
- JIT code reviews

- Transparency
- Objective commonality
- Project health checks
- Context driven

IBM Rational Software Architect

- **IBM Rational Software Architect is an integrated design and development tool.**
- Leverages model-driven development with the UML for creating well-architected applications and services.
- Develop applications and Web services more productively than ever
- Exploit the latest in modeling language technology and leverage an open and extensible modeling platform
- Review and control the structure of Java and service-oriented applications
- Simplifies the design and development tool solution by integrating with other facets of the lifecycle



Class represents logical application component for reviewing, updating, and managing user accounts

Name: GenerateAccountMgrCode

Source: Blank Model::Account Manager

● Transform without using a mapping model
○ Transform using a mapping model:
○ Only create mapping model (no transformati

Name: Account Manager

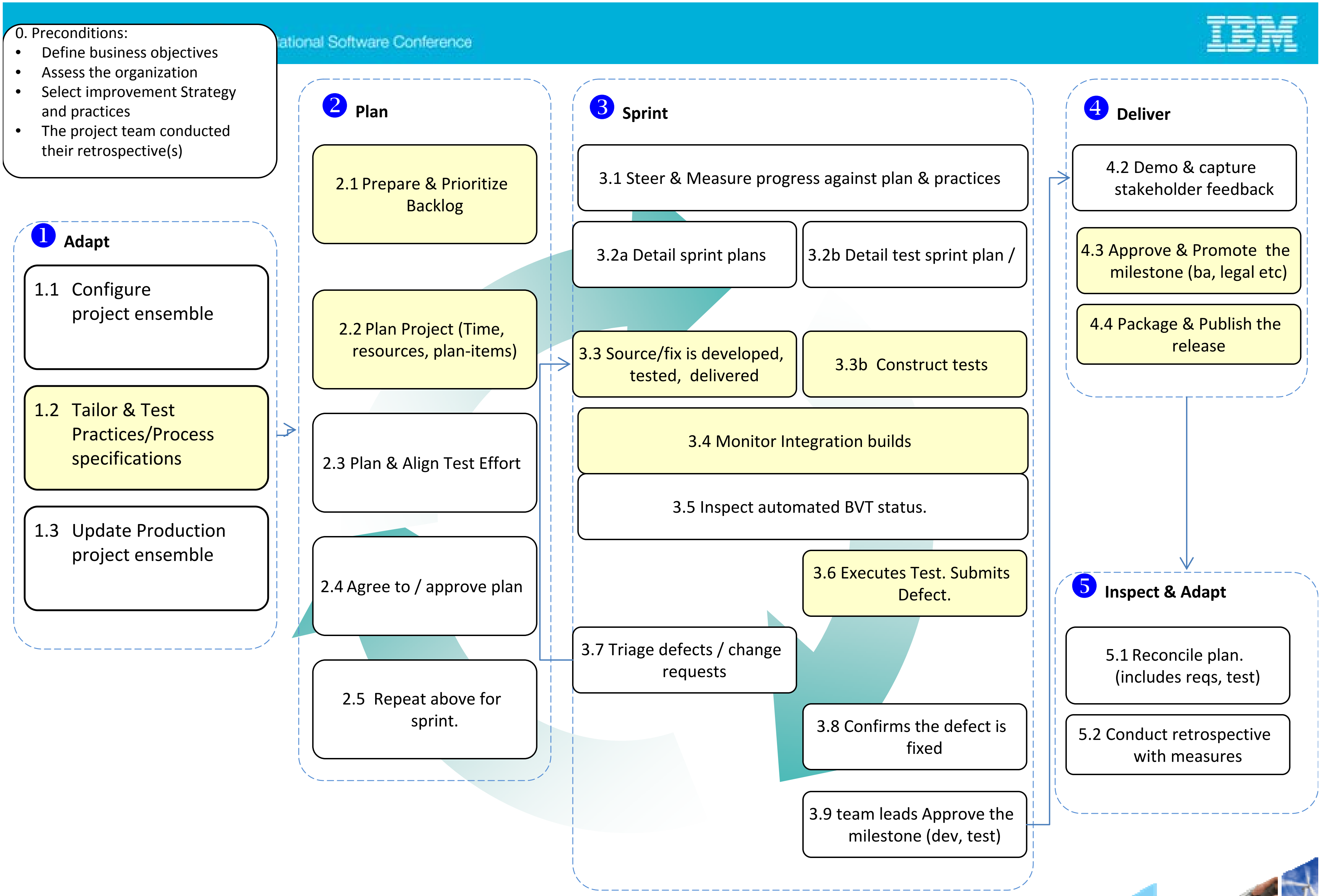
Visibility: ● Public ○ Private ○ Protected ○ Package

Abstract: □

Roles of RAM, RTC and RSA

Use IBM® Rational® Asset Manager with RSA and RTC to:

- Browse and search for assets stored in from RSA / RTC
- Rank, comment, or discuss on assets
- Govern asset management and define submission and approval lifecycles
- Create and submit assets from RSA / RTC into RAM



0. Preconditions:

- Define business objectives
- Assess the organization
- Select improvement Strategy and practices
- The project team conducted their retrospective(s)

1 Adapt

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Production project ensemble

2 Plan

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items)

2.3 Plan & Align Test Effort

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

3 Sprint

3.1 Steer & Measure progress against plan & practices

3.2a Detail sprint plans

3.2b Detail test sprint plan /

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect automated BVT status.

3.6 Executes Test. Submits Defect.

3.7 Triage defects / change requests

3.8 Confirms the defect is fixed

3.9 team leads Approve the milestone (dev, test)

4 Deliver

4.2 Demo & capture stakeholder feedback

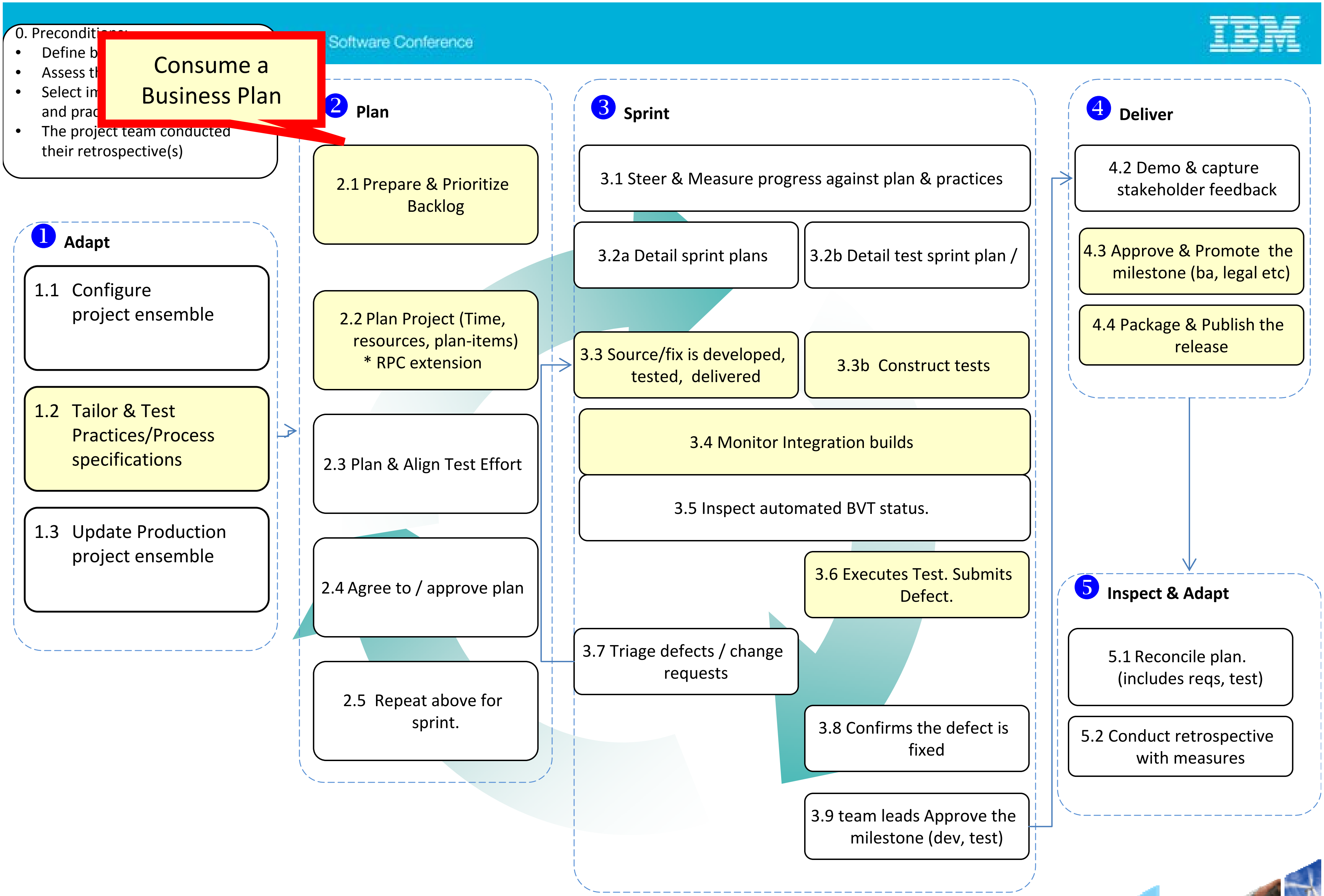
4.3 Approve & Promote the milestone (ba, legal etc)

4.4 Package & Publish the release

5 Inspect & Adapt

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures



0. Preconditi
- Define b
- Assess t
- Select in and prac
- The project team conducted their retrospective(s)

Consume a Business Plan

1 **Adapt**

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Production project ensemble

2 **Plan**

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items) * RPC extension

2.3 Plan & Align Test Effort

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

3 **Sprint**

3.1 Steer & Measure progress against plan & practices

3.2a Detail sprint plans

3.2b Detail test sprint plan /

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect automated BVT status.

3.6 Executes Test. Submits Defect.

3.7 Triage defects / change requests

3.8 Confirms the defect is fixed

3.9 team leads Approve the milestone (dev, test)

4 **Deliver**

4.2 Demo & capture stakeholder feedback

4.3 Approve & Promote the milestone (ba, legal etc)

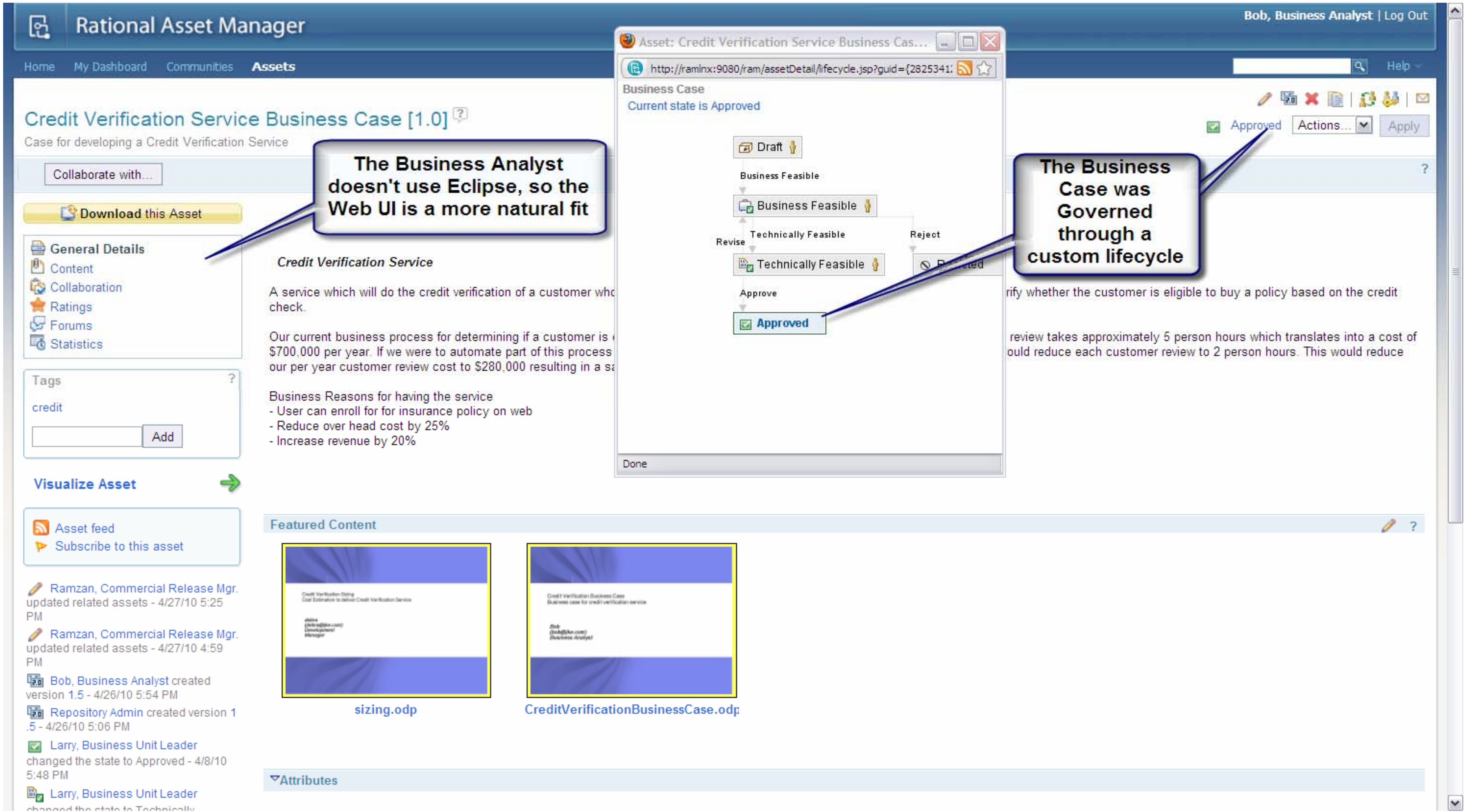
4.4 Package & Publish the release

5 **Inspect & Adapt**

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures

Business User publishes the Business Case



Credit Verification Service Business Case [1.0]

Case for developing a Credit Verification Service

Tags: credit

Credit Verification Service

A service which will do the credit verification of a customer whc … rify whether the customer is eligible to buy a policy based on the credit check.

Our current business process for determining if a customer is … review takes approximately 5 person hours which translates into a cost of $700,000 per year. If we were to automate part of this process … ould reduce each customer review to 2 person hours. This would reduce our per year customer review cost to $280,000 resulting in a sa

Business Reasons for having the service
- User can enroll for for insurance policy on web
- Reduce over head cost by 25%
- Increase revenue by 20%

Ramzan, Commercial Release Mgr. updated related assets - 4/27/10 5:25 PM

Ramzan, Commercial Release Mgr. updated related assets - 4/27/10 4:59 PM

Bob, Business Analyst created version 1.5 - 4/26/10 5:54 PM

Repository Admin created version 1.5 - 4/26/10 5:06 PM

Larry, Business Unit Leader changed the state to Approved - 4/8/10 5:48 PM

Larry, Business Unit Leader changed the state to Technically

Asset: Credit Verification Service Business Cas...

http://ramlnx:9080/ram/assetDetail/lifecycle.jsp?guid={2825341:

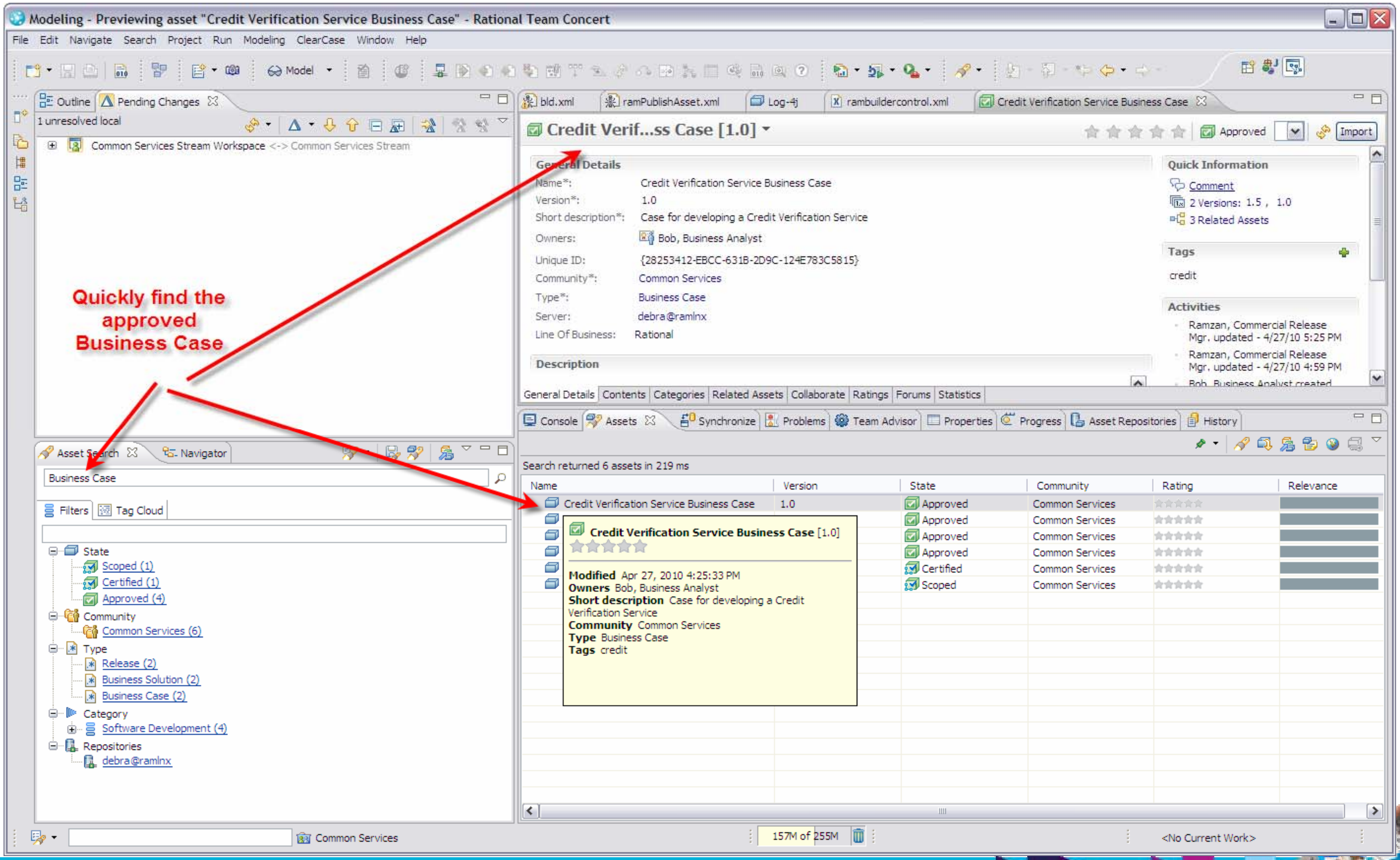
Business Case

Current state is Approved

The Business Analyst doesn't use Eclipse, so the Web UI is a more natural fit

The Business Case was Governed through a custom lifecycle

Scrum Master prepares backlog, using the approved Business Case



Quickly find the approved Business Case

General Details

Name*:	Credit Verification Service Business Case
Version*:	1.0
Short description*:	Case for developing a Credit Verification Service
Owners:	Bob, Business Analyst
Unique ID:	{28253412-EBCC-631B-2D9C-124E783C5815}
Community*:	Common Services
Type*:	Business Case
Server:	debra@ramlnx
Line Of Business:	Rational

Description

Quick Information

Comment
2 Versions: 1.5 , 1.0
3 Related Assets

Tags

credit

Activities

- Ramzan, Commercial Release Mgr. updated - 4/27/10 5:25 PM
- Ramzan, Commercial Release Mgr. updated - 4/27/10 4:59 PM
- Bob, Business Analyst created

Business Case

- State
  - Scoped (1)
  - Certified (1)
  - Approved (4)
- Community
  - Common Services (6)
- Type
  - Release (2)
  - Business Solution (2)
  - Business Case (2)
- Category
  - Software Development (4)
- Repositories
  - debra@ramlnx

Search returned 6 assets in 219 ms

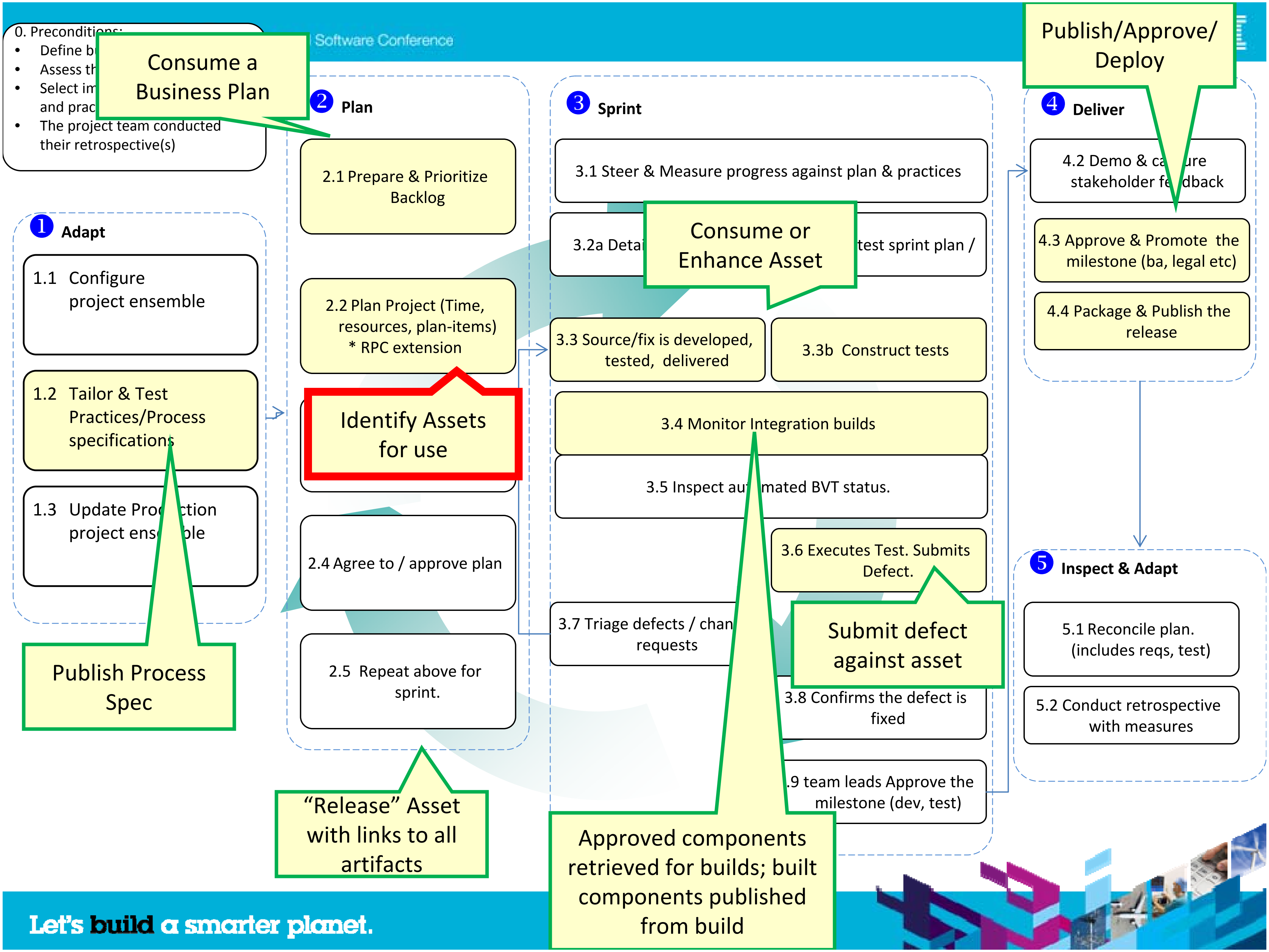
Name	Version	State	Community	Rating	Relevance
Credit Verification Service Business Case	1.0	Approved	Common Services	☆☆☆☆☆	
		Approved	Common Services	☆☆☆☆☆	
		Approved	Common Services	☆☆☆☆☆	
		Approved	Common Services	☆☆☆☆☆	
		Certified	Common Services	☆☆☆☆☆	
		Scoped	Common Services	☆☆☆☆☆	

Credit Verification Service Business Case [1.0]

☆☆☆☆☆

Modified Apr 27, 2010 4:25:33 PM
Owners Bob, Business Analyst
Short description Case for developing a Credit Verification Service
Community Common Services
Type Business Case
Tags credit

Common Services



0. Preconditions:

- Define bu
- Assess th
- Select im and prac
- The project team conducted their retrospective(s)

Consume a Business Plan

2 Plan

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items) * RPC extension

Identify Assets for use

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

"Release" Asset with links to all artifacts

1 Adapt

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Prod ction project ens ble

Publish Process Spec

3 Sprint

3.1 Steer & Measure progress against plan & practices

3.2a Detai

test sprint plan /

Consume or Enhance Asset

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect au mated BVT status.

3.6 Executes Test. Submits Defect.

3.7 Triage defects / chan requests

Submit defect against asset

3.8 Confirms the defect is fixed

.9 team leads Approve the milestone (dev, test)

Approved components retrieved for builds; built components published from build

Publish/Approve/ Deploy

4 Deliver

4.2 Demo & ca ure stakeholder fe dback

4.3 Approve & Promote the milestone (ba, legal etc)

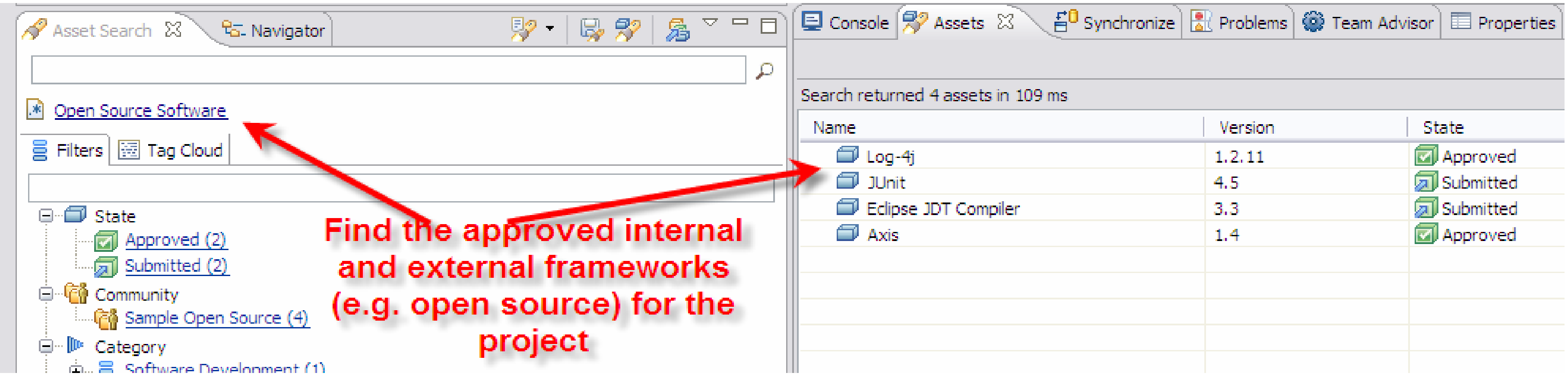
4.4 Package & Publish the release

5 Inspect & Adapt

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures

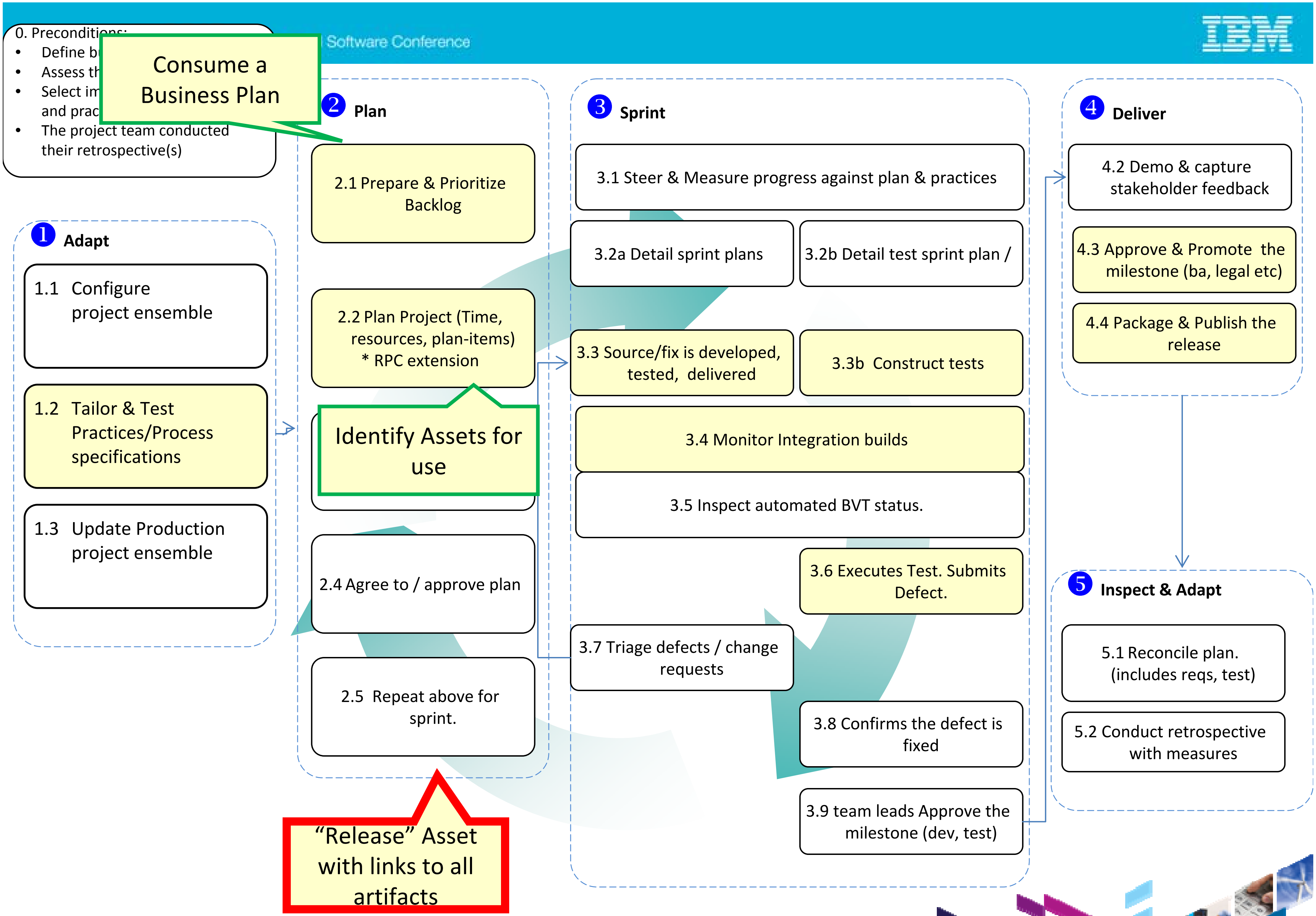
Identify Assets for use



Search returned 4 assets in 109 ms

Name	Version	State
Log-4j	1.2.11	Approved
JUnit	4.5	Submitted
Eclipse JDT Compiler	3.3	Submitted
Axis	1.4	Approved

Find the approved internal and external frameworks (e.g. open source) for the project



0. Preconditions:
- Define b
- Assess th
- Select im and prac
- The project team conducted their retrospective(s)

Consume a Business Plan

1 Adapt

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Production project ensemble

2 Plan

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items) * RPC extension

Identify Assets for use

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

"Release" Asset with links to all artifacts

3 Sprint

3.1 Steer & Measure progress against plan & practices

3.2a Detail sprint plans

3.2b Detail test sprint plan /

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect automated BVT status.

3.6 Executes Test. Submits Defect.

3.7 Triage defects / change requests

3.8 Confirms the defect is fixed

3.9 team leads Approve the milestone (dev, test)

4 Deliver

4.2 Demo & capture stakeholder feedback

4.3 Approve & Promote the milestone (ba, legal etc)

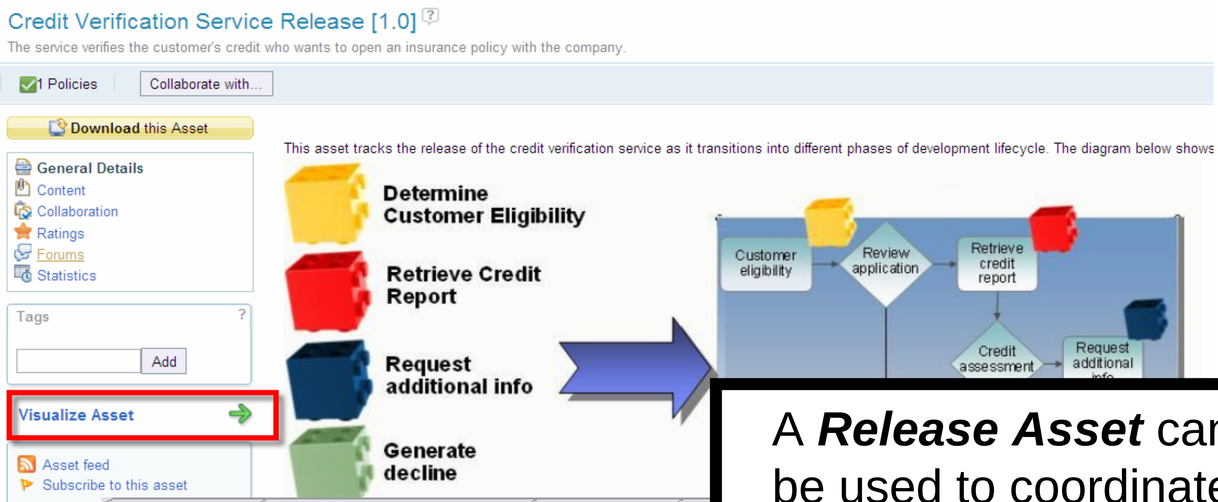
4.4 Package & Publish the release

5 Inspect & Adapt

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures

The Release Asset in RAM



Credit Verification Service Release [1.0]

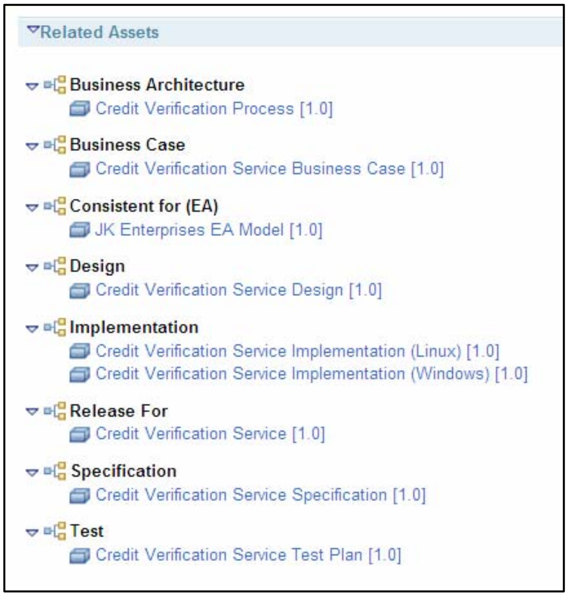
The service verifies the customer's credit who wants to open an insurance policy with the company.

This asset tracks the release of the credit verification service as it transitions into different phases of development lifecycle. The diagram below shows

Repository information and PM

Repository assets - 6/8/10

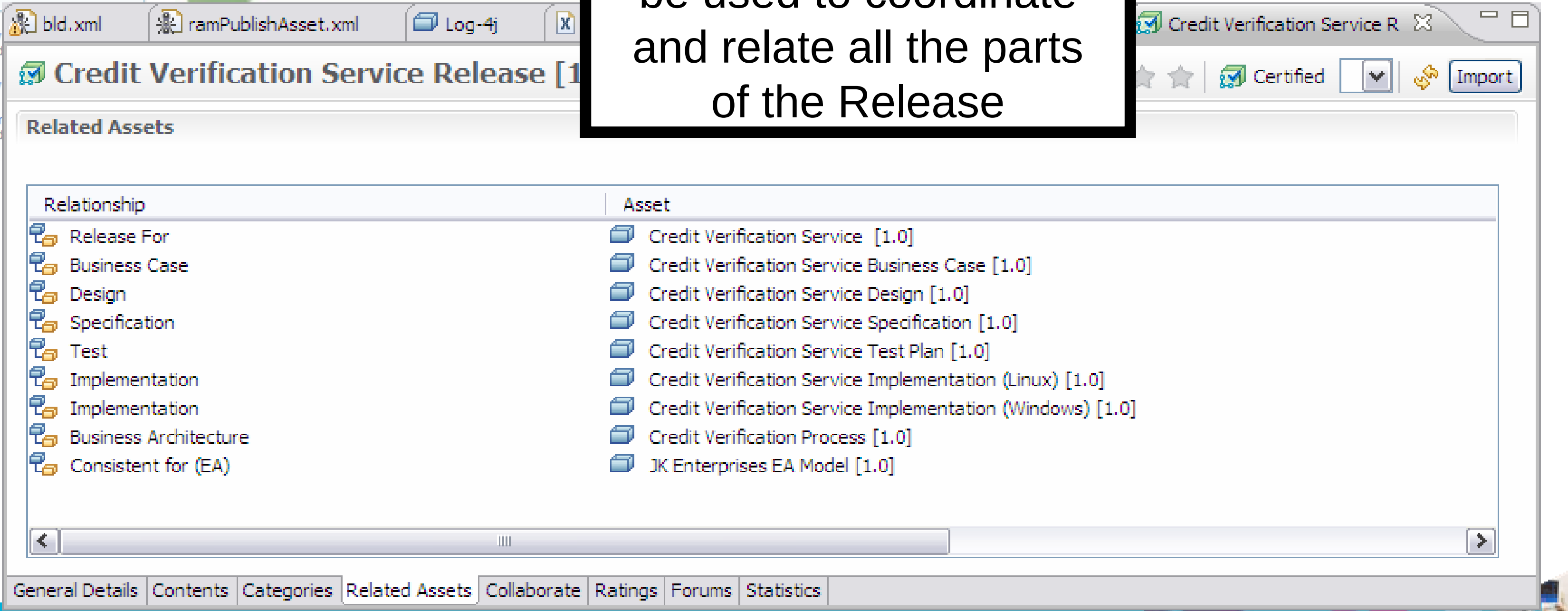
Debra, Co updated related PM



Related Assets

- Business Architecture
  - Credit Verification Process [1.0]
- Business Case
  - Credit Verification Service Business Case [1.0]
- Consistent for (EA)
  - JK Enterprises EA Model [1.0]
- Design
  - Credit Verification Service Design [1.0]
- Implementation
  - Credit Verification Service Implementation (Linux) [1.0]
  - Credit Verification Service Implementation (Windows) [1.0]
- Release For
  - Credit Verification Service [1.0]
- Specification
  - Credit Verification Service Specification [1.0]
- Test
  - Credit Verification Service Test Plan [1.0]

A ***Release Asset*** can be used to coordinate and relate all the parts of the Release

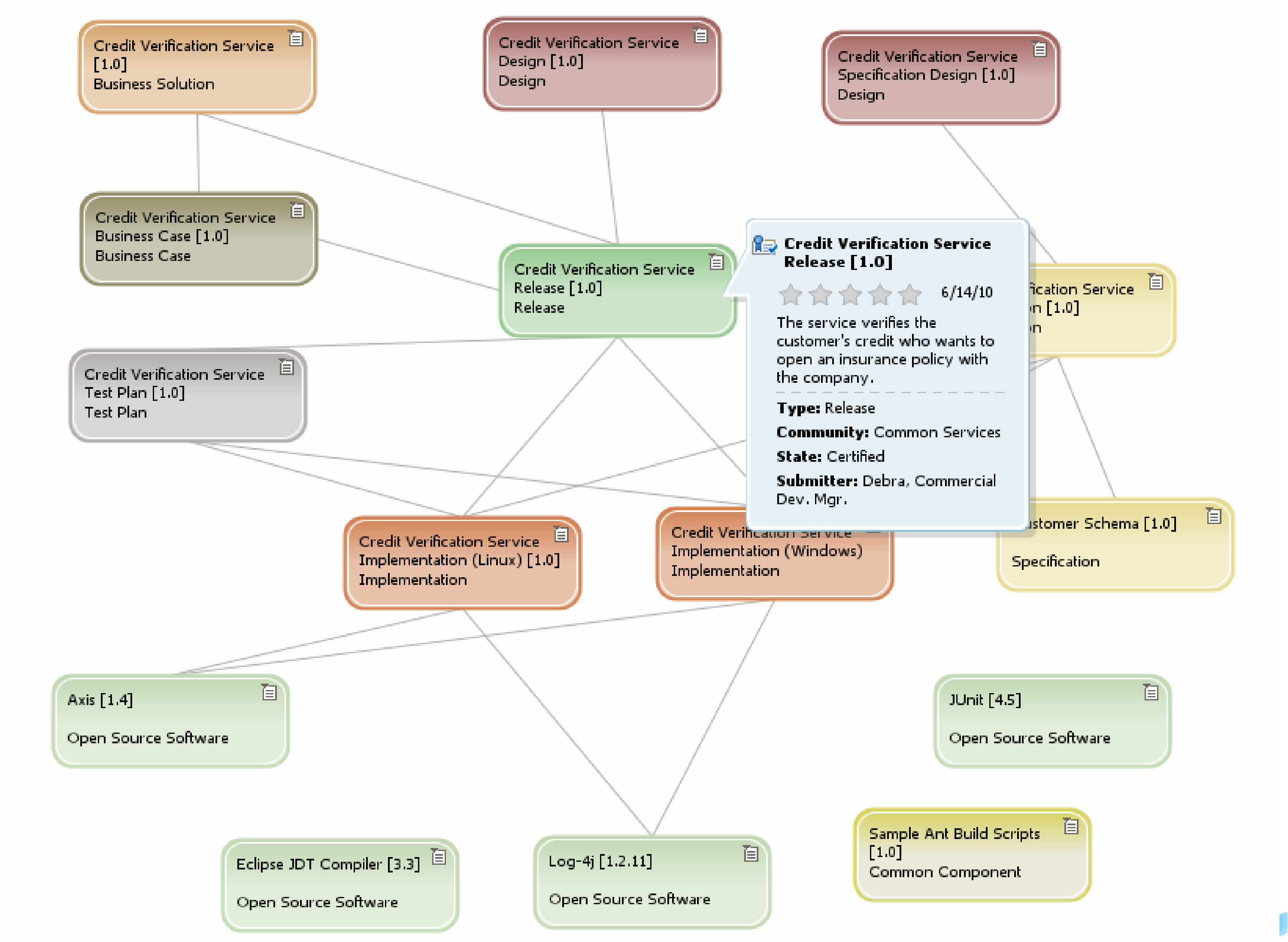


Credit Verification Service Release [1

Related Assets

Relationship	Asset
Release For	Credit Verification Service [1.0]
Business Case	Credit Verification Service Business Case [1.0]
Design	Credit Verification Service Design [1.0]
Specification	Credit Verification Service Specification [1.0]
Test	Credit Verification Service Test Plan [1.0]
Implementation	Credit Verification Service Implementation (Linux) [1.0]
Implementation	Credit Verification Service Implementation (Windows) [1.0]
Business Architecture	Credit Verification Process [1.0]
Consistent for (EA)	JK Enterprises EA Model [1.0]

Visual Browse the Release Asset



Credit Verification Service [1.0]
Business Solution

Credit Verification Service Design [1.0]
Design

Credit Verification Service Specification Design [1.0]
Design

Credit Verification Service Business Case [1.0]
Business Case

Credit Verification Service Release [1.0]
Release

Credit Verification Service Release [1.0]

☆☆☆☆☆ 6/14/10

The service verifies the customer's credit who wants to open an insurance policy with the company.

Type: Release

Community: Common Services

State: Certified

Submitter: Debra, Commercial Dev. Mgr.

ification Service
n [1.0]
n

Credit Verification Service Test Plan [1.0]
Test Plan

Credit Verification Service Implementation (Linux) [1.0]
Implementation

Credit Verification Service Implementation (Windows)
Implementation

stomer Schema [1.0]
Specification

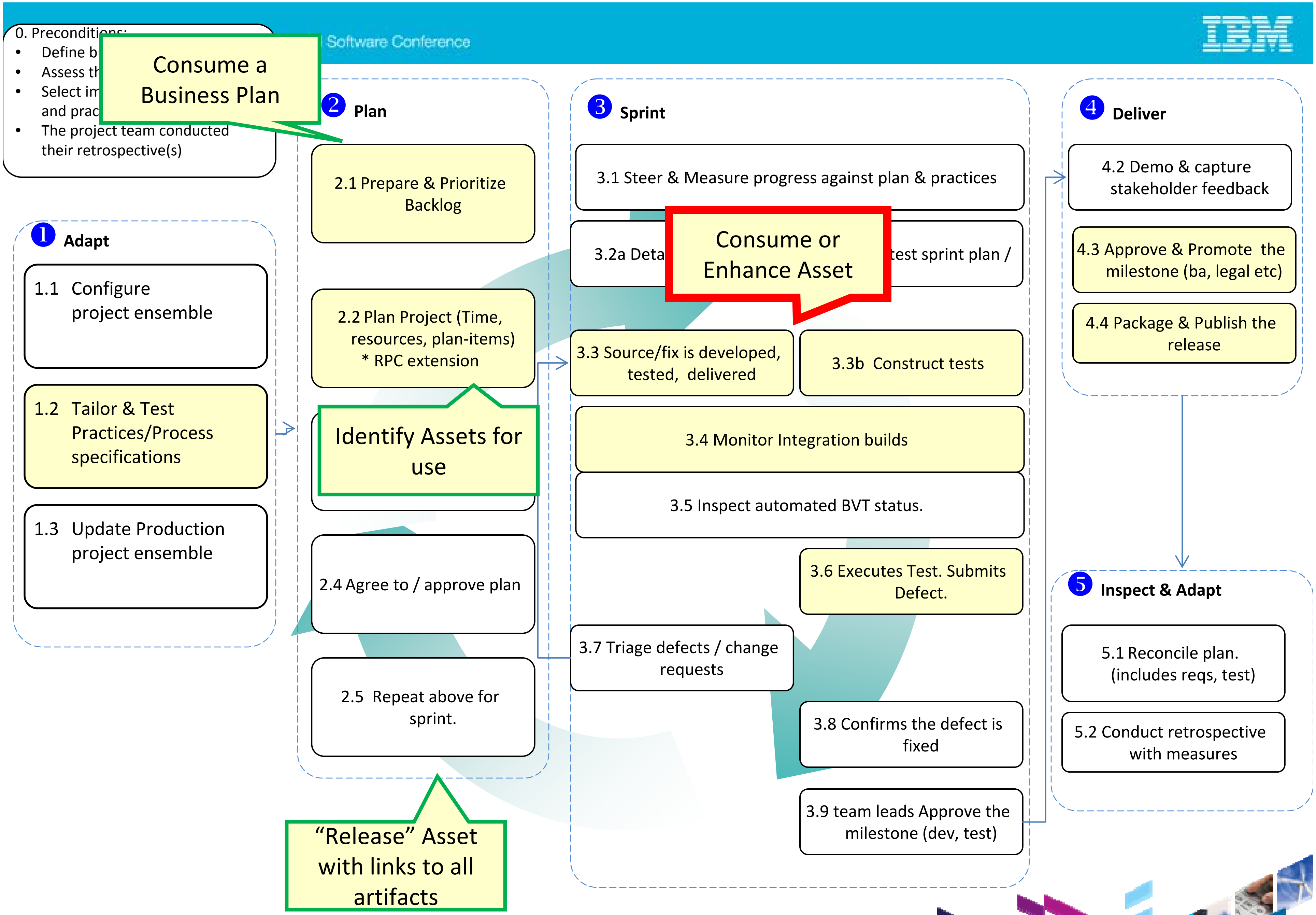
Axis [1.4]
Open Source Software

JUnit [4.5]
Open Source Software

Sample Ant Build Scripts [1.0]
Common Component

Eclipse JDT Compiler [3.3]
Open Source Software

Log-4j [1.2.11]
Open Source Software



0. Preconditions:

- Define b
- Assess th
- Select im and prac
- The project team conducted their retrospective(s)

Consume a Business Plan

1 Adapt

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Production project ensemble

2 Plan

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items) * RPC extension

Identify Assets for use

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

"Release" Asset with links to all artifacts

3 Sprint

3.1 Steer & Measure progress against plan & practices

3.2a Deta

Consume or Enhance Asset

test sprint plan /

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect automated BVT status.

3.6 Executes Test. Submits Defect.

3.7 Triage defects / change requests

3.8 Confirms the defect is fixed

3.9 team leads Approve the milestone (dev, test)

4 Deliver

4.2 Demo & capture stakeholder feedback

4.3 Approve & Promote the milestone (ba, legal etc)

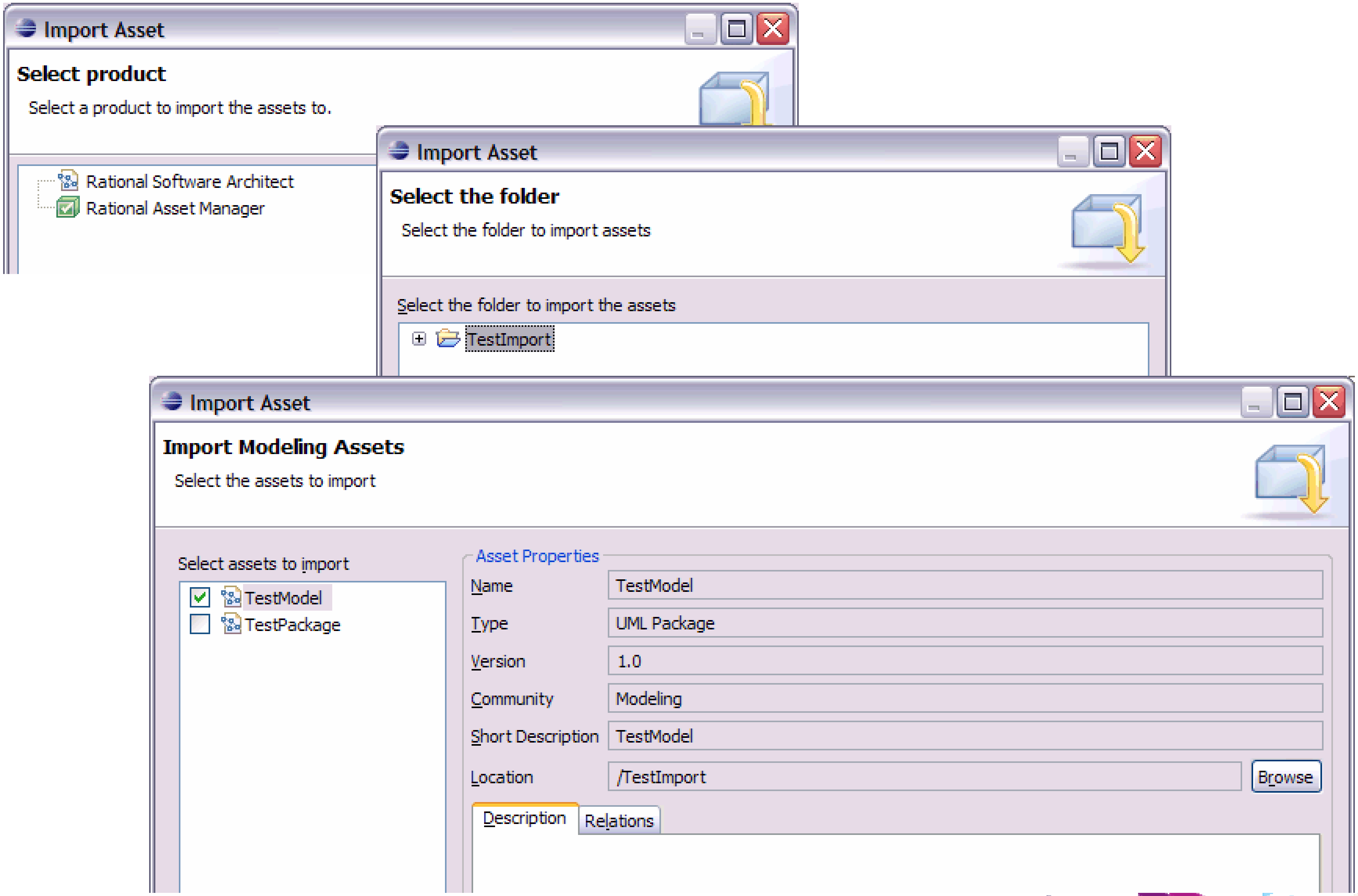
4.4 Package & Publish the release

5 Inspect & Adapt

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures

Consume or enhance assets



Import Asset

Select product

Select a product to import the assets to.

- Rational Software Architect
- Rational Asset Manager

Import Asset

Select the folder

Select the folder to import assets

Select the folder to import the assets

⊞ TestImport

Import Asset

Import Modeling Assets

Select the assets to import

Select assets to import

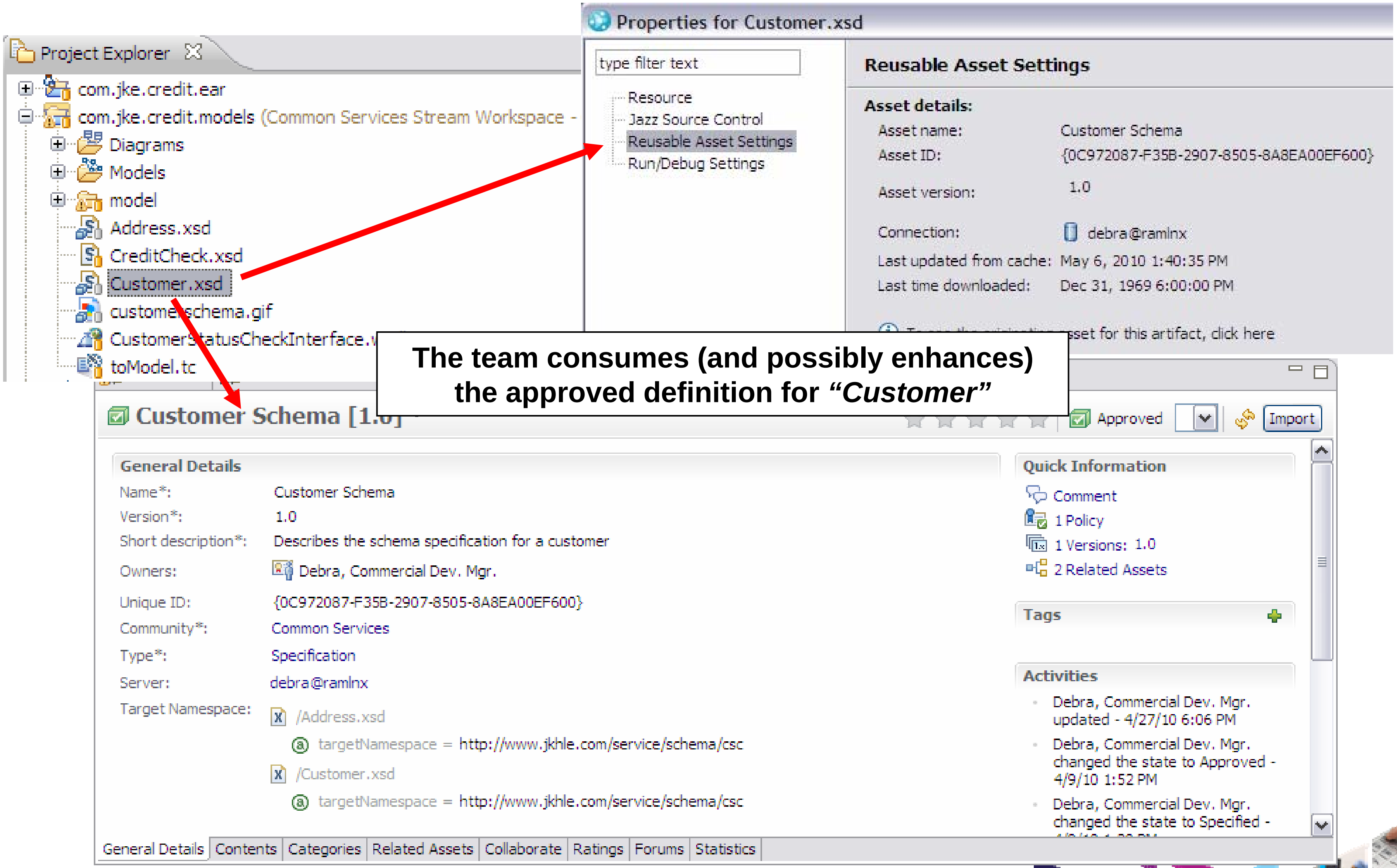
☑ TestModel

☐ TestPackage

Asset Properties

Name	TestModel
Type	UML Package
Version	1.0
Community	Modeling
Short Description	TestModel
Location	/TestImport

Consume or enhance assets



The team consumes (and possibly enhances) the approved definition for *"Customer"*

Properties for Customer.xsd

Reusable Asset Settings

Asset details:

Asset name:	Customer Schema
Asset ID:	{0C972087-F35B-2907-8505-8A8EA00EF600}
Asset version:	1.0
Connection:	debra@ramlnx
Last updated from cache:	May 6, 2010 1:40:35 PM
Last time downloaded:	Dec 31, 1969 6:00:00 PM

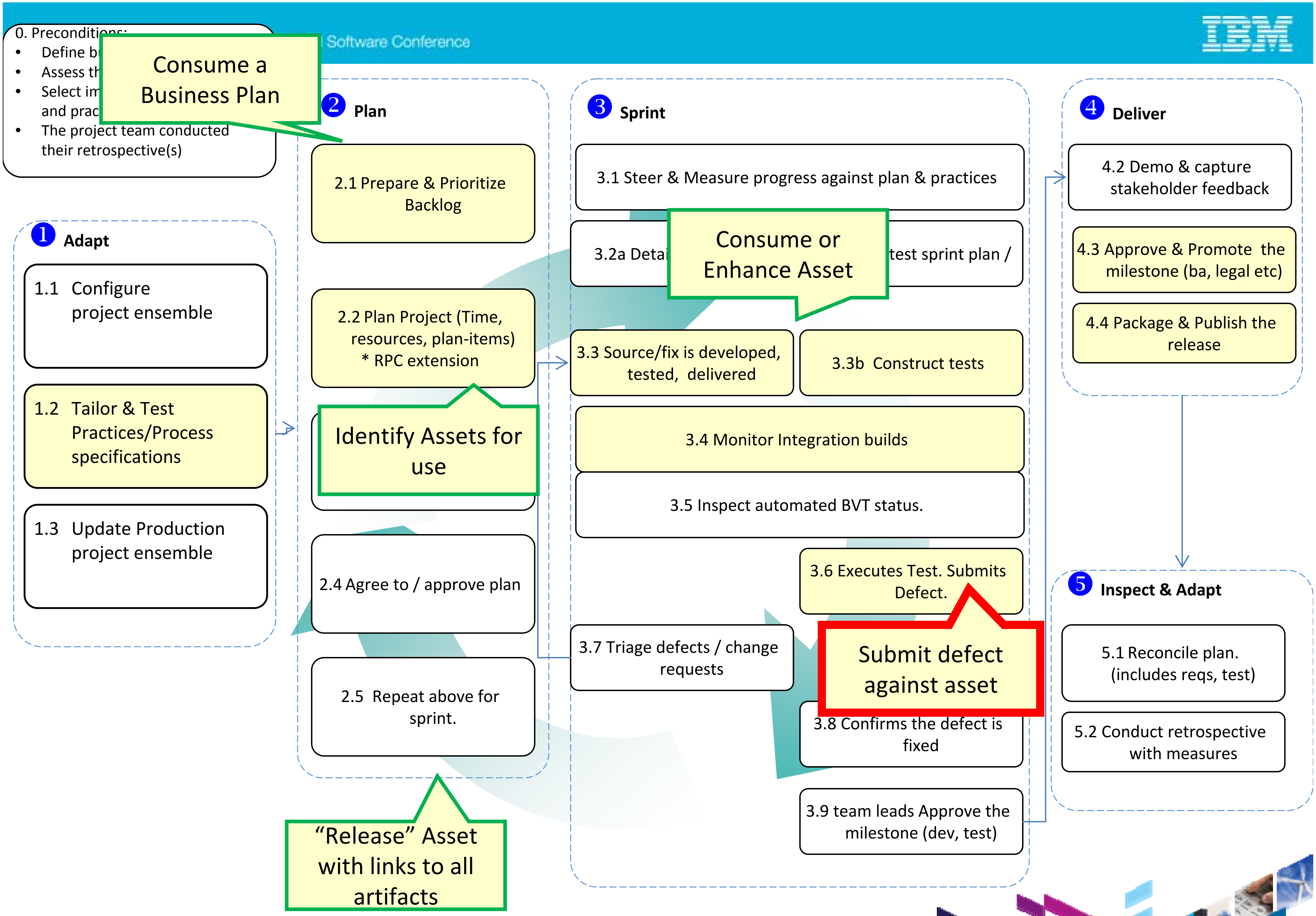
Customer Schema [1.0]

General Details

Name*:	Customer Schema
Version*:	1.0
Short description*:	Describes the schema specification for a customer
Owners:	Debra, Commercial Dev. Mgr.
Unique ID:	{0C972087-F35B-2907-8505-8A8EA00EF600}
Community*:	Common Services
Type*:	Specification
Server:	debra@ramlnx
Target Namespace:	/Address.xsd targetNamespace = http://www.jkhle.com/service/schema/csc /Customer.xsd targetNamespace = http://www.jkhle.com/service/schema/csc

Activities

- Debra, Commercial Dev. Mgr. updated - 4/27/10 6:06 PM
- Debra, Commercial Dev. Mgr. changed the state to Approved - 4/9/10 1:52 PM
- Debra, Commercial Dev. Mgr. changed the state to Specified -



0. Preconditions:
- Define b
- Assess th
- Select im and prac
- The project team conducted their retrospective(s)

Consume a Business Plan

1 Adapt

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Production project ensemble

2 Plan

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items) * RPC extension

Identify Assets for use

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

“Release” Asset with links to all artifacts

3 Sprint

3.1 Steer & Measure progress against plan & practices

3.2a Detai

Consume or Enhance Asset

test sprint plan /

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect automated BVT status.

3.6 Executes Test. Submits Defect.

3.7 Triage defects / change requests

Submit defect against asset

3.8 Confirms the defect is fixed

3.9 team leads Approve the milestone (dev, test)

4 Deliver

4.2 Demo & capture stakeholder feedback

4.3 Approve & Promote the milestone (ba, legal etc)

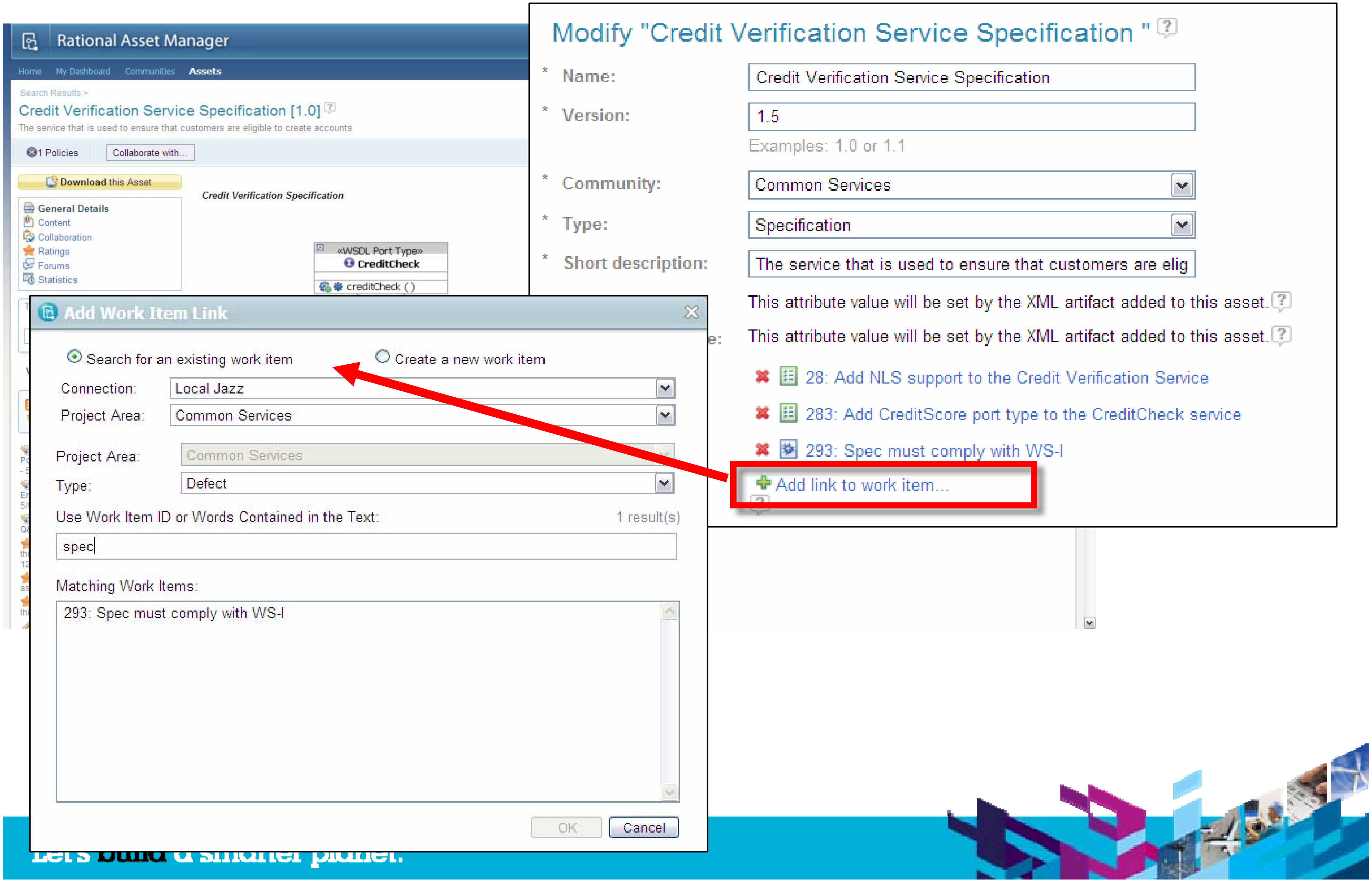
4.4 Package & Publish the release

5 Inspect & Adapt

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures

Submit work items against an Asset



Credit Verification Service Specification [1.0]

The service that is used to ensure that customers are eligible to create accounts

Credit Verification Specification

Add Work Item Link

● Search for an existing work item ○ Create a new work item

Connection: Local Jazz

Project Area: Common Services

Project Area: Common Services

Type: Defect

Use Work Item ID or Words Contained in the Text: 1 result(s)

spec

Matching Work Items:

293: Spec must comply with WS-I

Modify "Credit Verification Service Specification "

* Name: Credit Verification Service Specification

* Version: 1.5

Examples: 1.0 or 1.1

* Community: Common Services

* Type: Specification

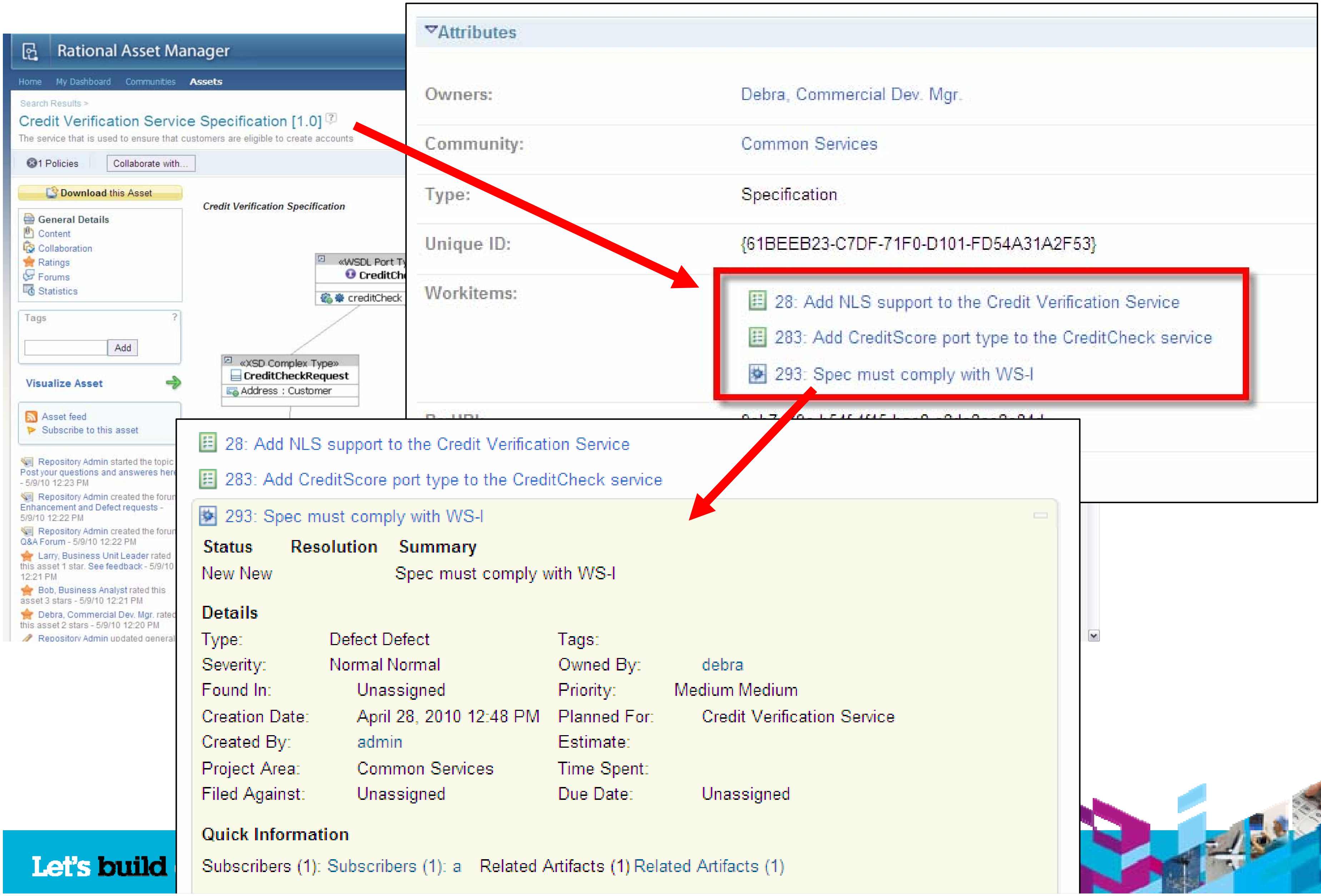
* Short description: The service that is used to ensure that customers are elig

This attribute value will be set by the XML artifact added to this asset.

This attribute value will be set by the XML artifact added to this asset.

- 28: Add NLS support to the Credit Verification Service
- 283: Add CreditScore port type to the CreditCheck service
- 293: Spec must comply with WS-I

Submit work items against an Asset



Rational Asset Manager

Home My Dashboard Communities **Assets**

Search Results >

Credit Verification Service Specification [1.0]

The service that is used to ensure that customers are eligible to create accounts

1 Policies | Collaborate with...

Download this Asset

- **General Details**
- Content
- Collaboration
- Ratings
- Forums
- Statistics

Tags ?

Add

Visualize Asset

- Asset feed
- Subscribe to this asset

Repository Admin started the topic Post your questions and answeres here - 5/9/10 12:23 PM

Repository Admin created the forum Enhancement and Defect requests - 5/9/10 12:22 PM

Repository Admin created the forum Q&A Forum - 5/9/10 12:22 PM

Larry, Business Unit Leader rated this asset 1 star. See feedback - 5/9/10 12:21 PM

Bob, Business Analyst rated this asset 3 stars - 5/9/10 12:21 PM

Debra, Commercial Dev. Mgr. rated this asset 2 stars - 5/9/10 12:20 PM

Repository Admin updated general

Credit Verification Specification

Attributes

Owners: Debra, Commercial Dev. Mgr.

Community: Common Services

Type: Specification

Unique ID: {61BEEB23-C7DF-71F0-D101-FD54A31A2F53}

Workitems:
- 28: Add NLS support to the Credit Verification Service
- 283: Add CreditScore port type to the CreditCheck service
- 293: Spec must comply with WS-I

- 28: Add NLS support to the Credit Verification Service
- 283: Add CreditScore port type to the CreditCheck service
- 293: Spec must comply with WS-I

Status **Resolution** **Summary**
New New Spec must comply with WS-I

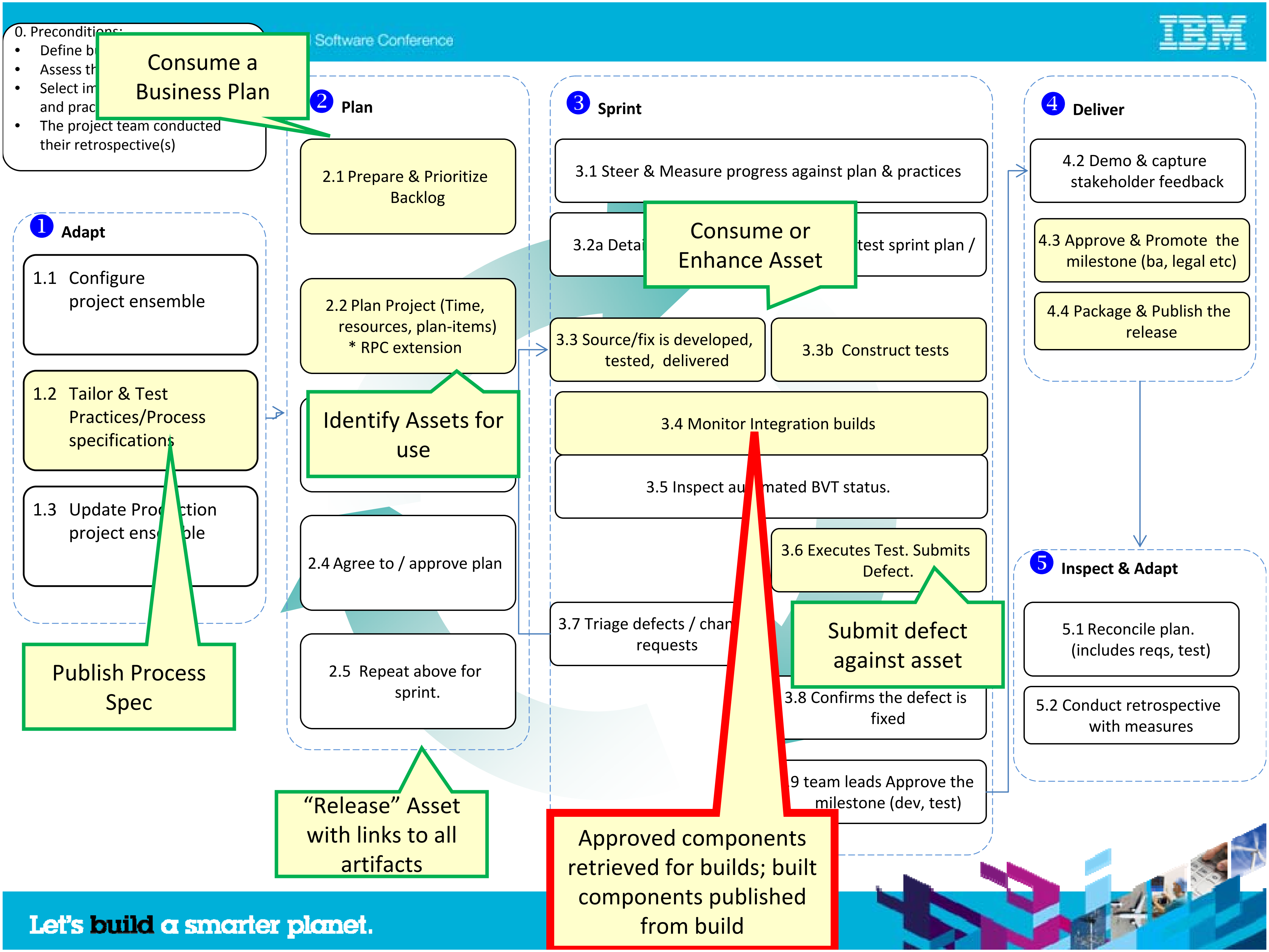
Details

Type: Defect Defect
Severity: Normal Normal
Found In: Unassigned
Creation Date: April 28, 2010 12:48 PM
Created By: admin
Project Area: Common Services
Filed Against: Unassigned

Tags:
Owned By: debra
Priority: Medium Medium
Planned For: Credit Verification Service
Estimate:
Time Spent:
Due Date: Unassigned

Quick Information

Subscribers (1): Subscribers (1): a Related Artifacts (1) Related Artifacts (1)



0. Preconditions:
- Define b
- Assess th
- Select im and prac
- The project team conducted their retrospective(s)

Consume a Business Plan

1 Adapt

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Pro ction project ens ble

Publish Process Spec

2 Plan

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items) * RPC extension

Identify Assets for use

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

"Release" Asset with links to all artifacts

3 Sprint

3.1 Steer & Measure progress against plan & practices

3.2a Detai test sprint plan /

Consume or Enhance Asset

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect au mated BVT status.

3.6 Executes Test. Submits Defect.

3.7 Triage defects / chan requests

Submit defect against asset

3.8 Confirms the defect is fixed

9 team leads Approve the milestone (dev, test)

Approved components retrieved for builds; built components published from build

4 Deliver

4.2 Demo & capture stakeholder feedback

4.3 Approve & Promote the milestone (ba, legal etc)

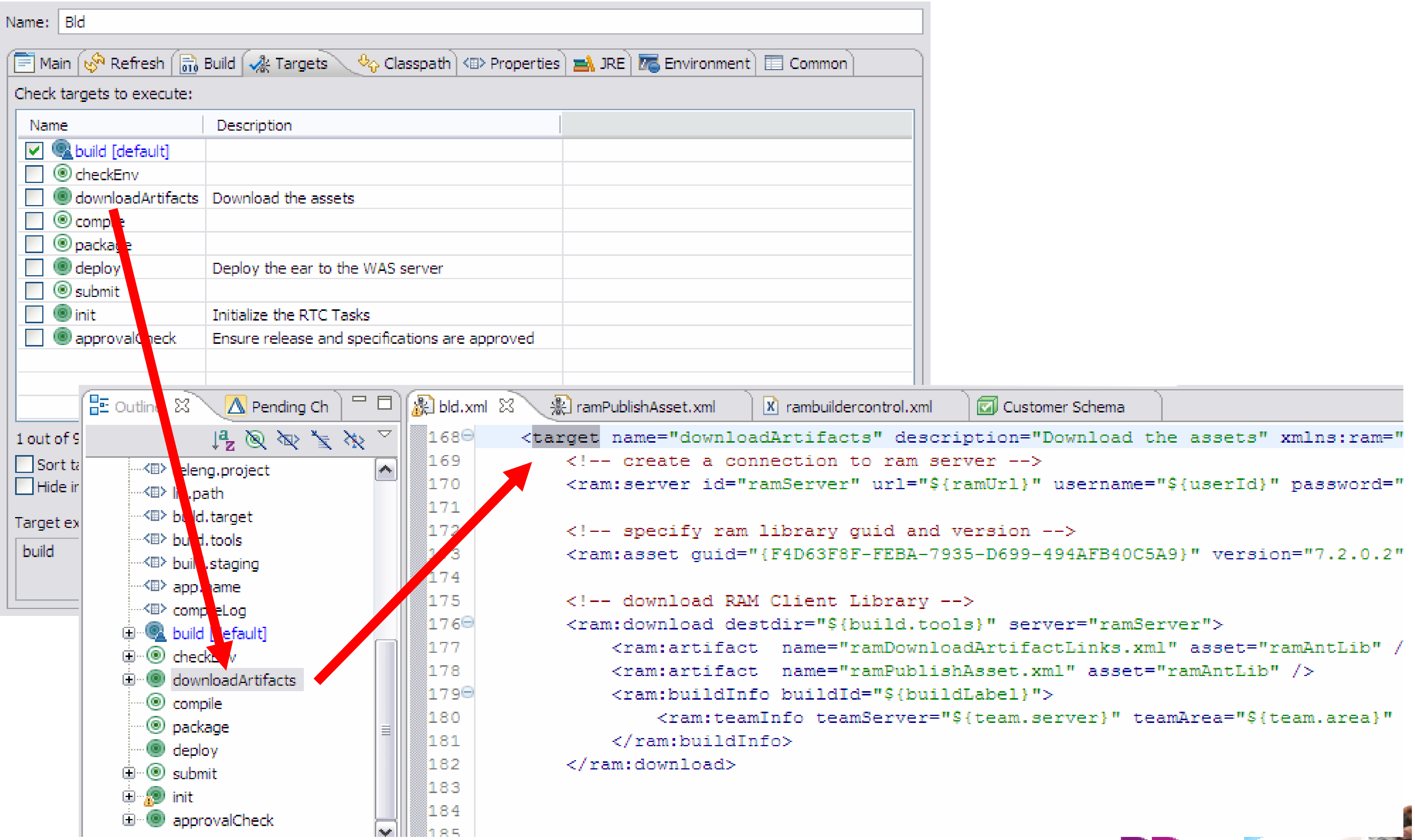
4.4 Package & Publish the release

5 Inspect & Adapt

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures

Approved components *consumed* for build



Name: Bld

Check targets to execute:

Name	Description
☑ build [default]	
□ checkEnv	
□ downloadArtifacts	Download the assets
□ comp e	
□ packa e	
□ deploy	Deploy the ear to the WAS server
□ submit	
□ init	Initialize the RTC Tasks
□ approval eck	Ensure release and specifications are approved

1 out of 9

□ Sort ta

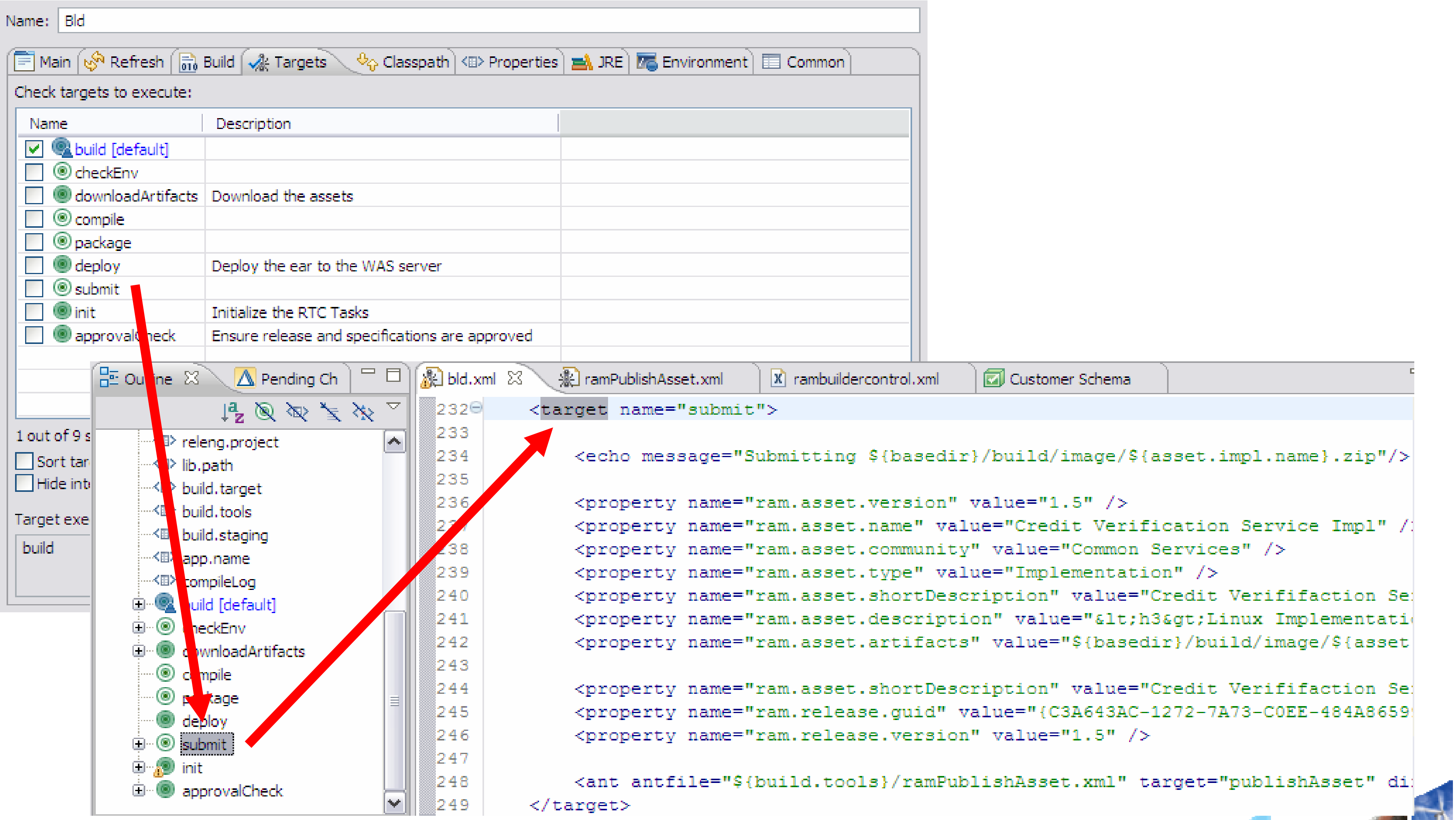
□ Hide ir

Target ex

build

```
    <target name="downloadArtifacts" description="Download the assets" xmlns:ram="
        <!-- create a connection to ram server -->
        <ram:server id="ramServer" url="${ramUrl}" username="${userId}" password="

        <!-- specify ram library guid and version -->
        <ram:asset guid="{F4D63F8F-FEBA-7935-D699-494AFB40C5A9}" version="7.2.0.2"

        <!-- download RAM Client Library -->
        <ram:download destdir="${build.tools}" server="ramServer">
            <ram:artifact  name="ramDownloadArtifactLinks.xml" asset="ramAntLib" /
            <ram:artifact  name="ramPublishAsset.xml" asset="ramAntLib" />
            <ram:buildInfo buildId="${buildLabel}">
                <ram:teamInfo teamServer="${team.server}" teamArea="${team.area}"
            </ram:buildInfo>
        </ram:download>
```

Publish built components



Name: Bld

Check targets to execute:

	Name	Description
☑	build [default]	
☐	checkEnv	
☐	downloadArtifacts	Download the assets
☐	compile	
☐	package	
☐	deploy	Deploy the ear to the WAS server
☐	submit	
☐	init	Initialize the RTC Tasks
☐	approvalCheck	Ensure release and specifications are approved

1 out of 9 s

☐ Sort tar

☐ Hide int

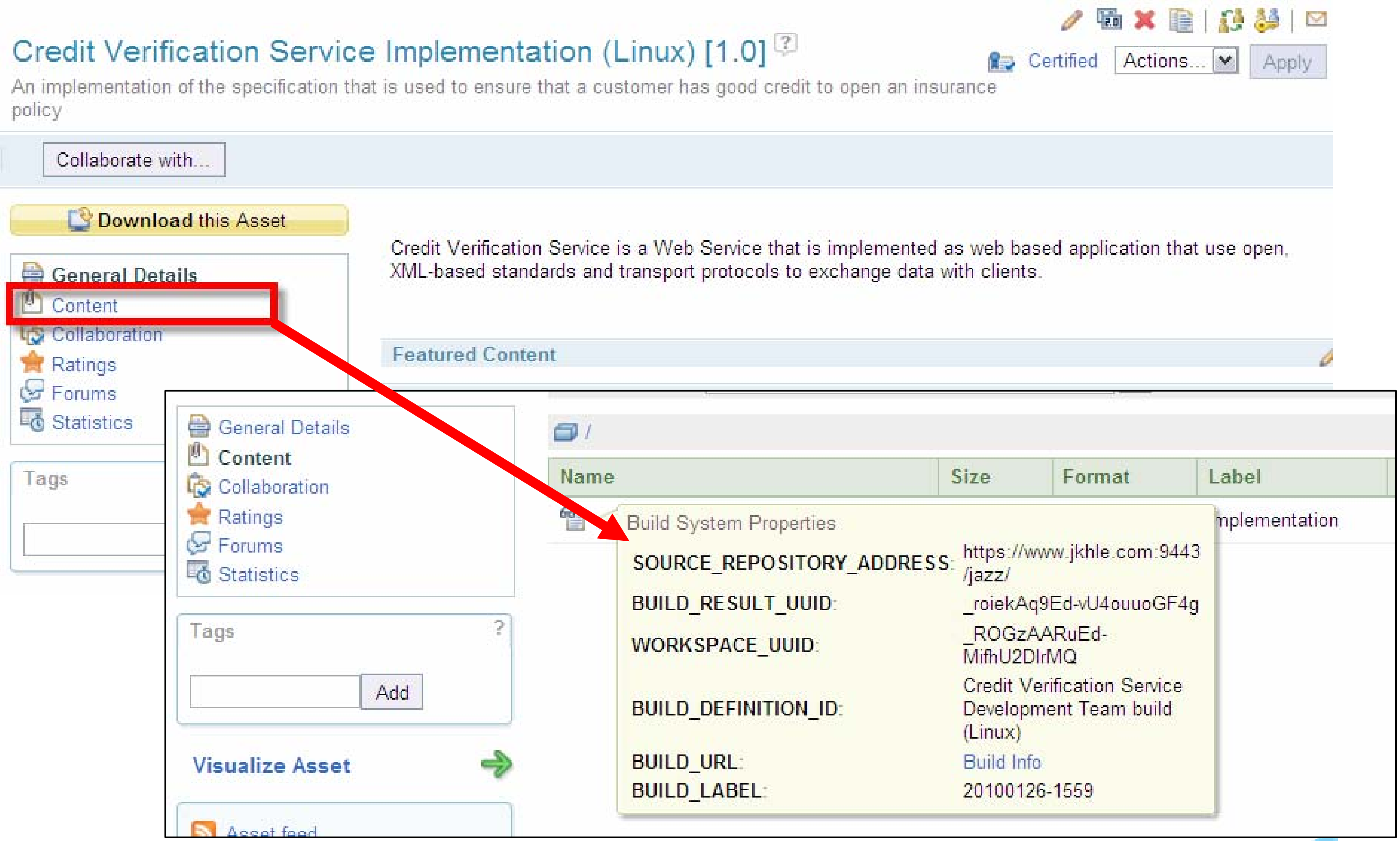
Target exe

build

- releng.project
- lib.path
- build.target
- build.tools
- build.staging
- app.name
- compileLog
- build [default]
- checkEnv
- downloadArtifacts
- compile
- package
- deploy
- submit
- init
- approvalCheck

```
<target name="submit">

    <echo message="Submitting ${basedir}/build/image/${asset.impl.name}.zip"/>

    <property name="ram.asset.version" value="1.5" />
    <property name="ram.asset.name" value="Credit Verification Service Impl" /
    <property name="ram.asset.community" value="Common Services" />
    <property name="ram.asset.type" value="Implementation" />
    <property name="ram.asset.shortDescription" value="Credit Verififaction Se
    <property name="ram.asset.description" value="&lt;h3&gt;Linux Implementati
    <property name="ram.asset.artifacts" value="${basedir}/build/image/${asset

    <property name="ram.asset.shortDescription" value="Credit Verififaction Se
    <property name="ram.release.guid" value="{C3A643AC-1272-7A73-C0EE-484A8659
    <property name="ram.release.version" value="1.5" />

    <ant antfile="${build.tools}/ramPublishAsset.xml" target="publishAsset" di
</target>
```

Approved build is traced back to the build automation



Credit Verification Service Implementation (Linux) [1.0]

An implementation of the specification that is used to ensure that a customer has good credit to open an insurance policy

Certified | Actions... | Apply

Collaborate with...

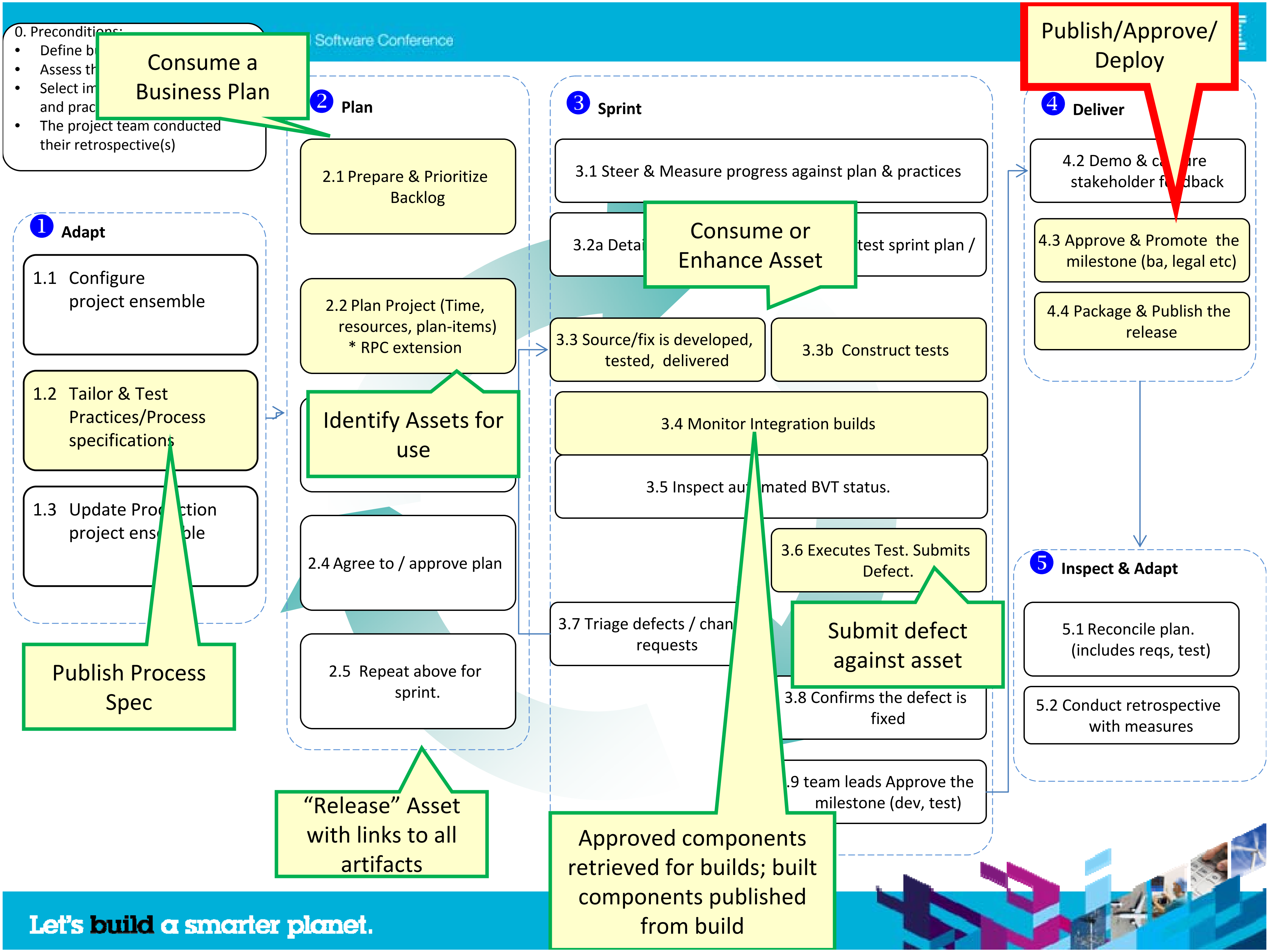
Download this Asset

Credit Verification Service is a Web Service that is implemented as web based application that use open, XML-based standards and transport protocols to exchange data with clients.

Build System Properties

SOURCE_REPOSITORY_ADDRESS:	https://www.jkhle.com:9443 /jazz/
BUILD_RESULT_UUID:	_roiekAq9Ed-vU4ouuoGF4g
WORKSPACE_UUID:	_ROGzAARuEd-MifhU2DIrMQ
BUILD_DEFINITION_ID:	Credit Verification Service Development Team build (Linux)
BUILD_URL:	Build Info
BUILD_LABEL:	20100126-1559

Software Conference



0. Preconditions:
- Define bu
- Assess th
- Select im and prac
- The project team conducted their retrospective(s)

Consume a Business Plan

1 **Adapt**

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Proc ction project ens ble

Publish Process Spec

2 **Plan**

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items) * RPC extension

Identify Assets for use

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

"Release" Asset with links to all artifacts

3 **Sprint**

3.1 Steer & Measure progress against plan & practices

3.2a Detai test sprint plan /

Consume or Enhance Asset

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect au mated BVT status.

3.6 Executes Test. Submits Defect.

Submit defect against asset

3.7 Triage defects / chan requests

3.8 Confirms the defect is fixed

.9 team leads Approve the milestone (dev, test)

Approved components retrieved for builds; built components published from build

4 **Deliver**

Publish/Approve/ Deploy

4.2 Demo & ca re stakeholder f dback

4.3 Approve & Promote the milestone (ba, legal etc)

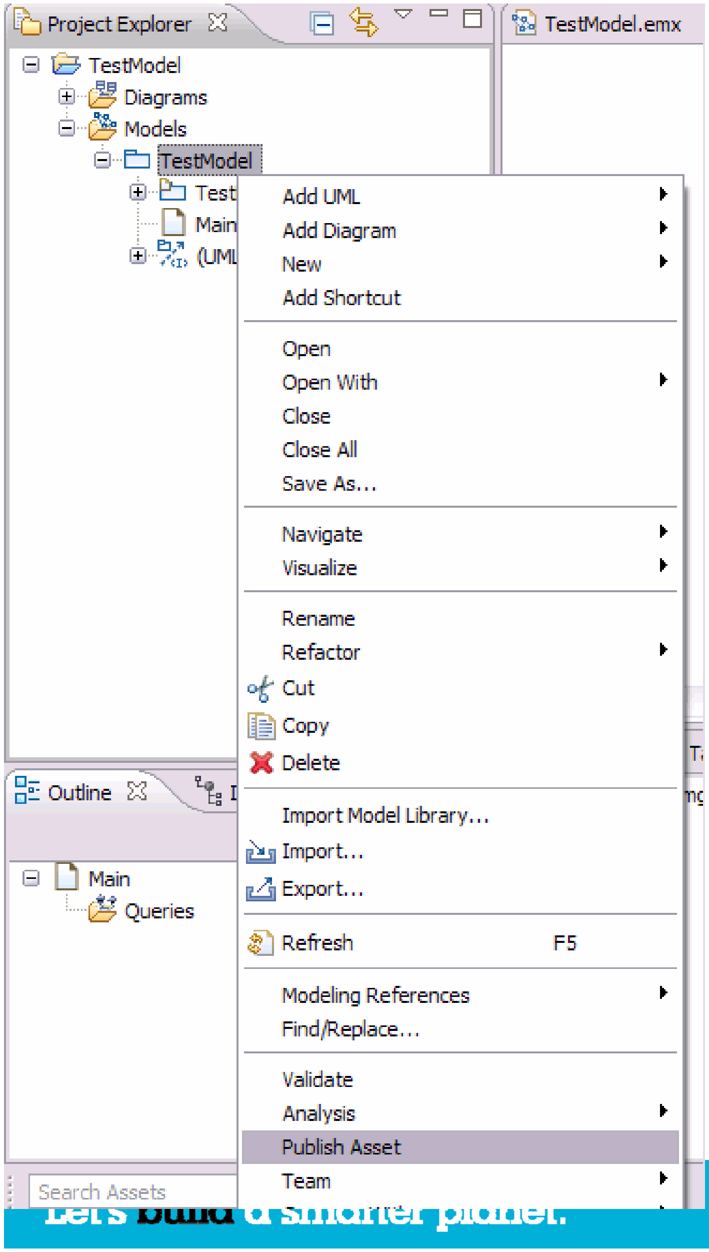
4.4 Package & Publish the release

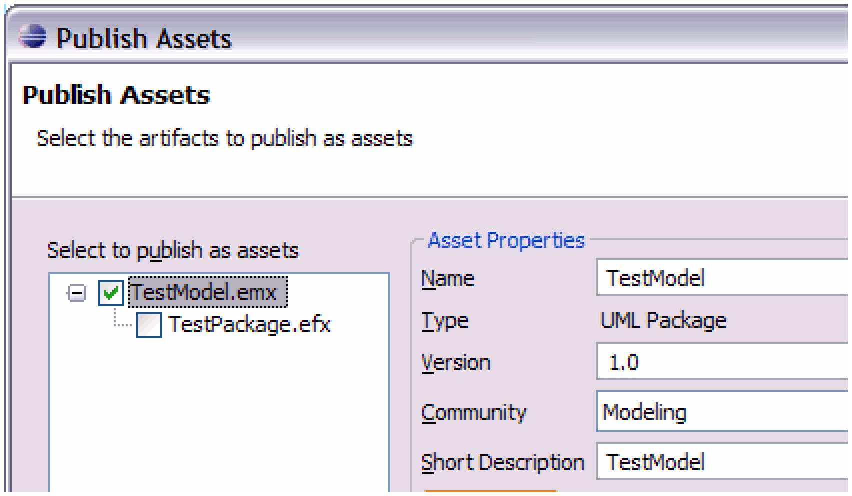
5 **Inspect & Adapt**

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures

Let's build a smarter planet.





Publish Assets

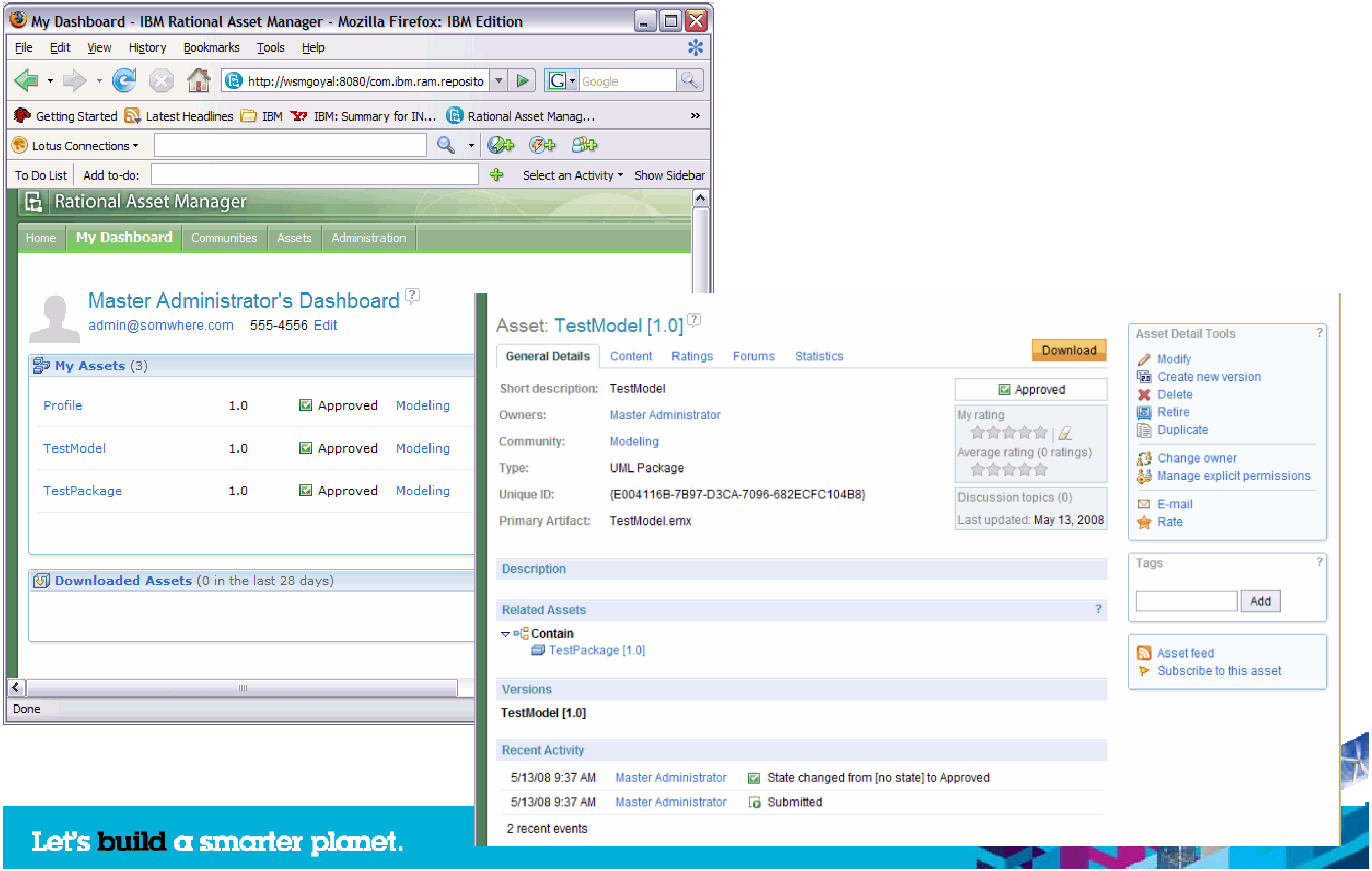
Select the artifacts to publish as assets

Select to publish as assets

- ☑ TestModel.emx
  - □ TestPackage.efx

Asset Properties

Name	TestModel
Type	UML Package
Version	1.0
Community	Modeling
Short Description	TestModel



My Dashboard - IBM Rational Asset Manager - Mozilla Firefox: IBM Edition

File Edit View History Bookmarks Tools Help
http://wsmgoyal:8080/com.ibm.ram.reposito
Google
Getting Started Latest Headlines IBM IBM: Summary for IN... Rational Asset Manag...
Lotus Connections
To Do List Add to-do: Select an Activity Show Sidebar
Rational Asset Manager
Home My Dashboard Communities Assets Administration
Done

Master Administrator's Dashboard

admin@somwhere.com 555-4556 Edit

My Assets (3)

Profile	1.0	☑ Approved	Modeling
TestModel	1.0	☑ Approved	Modeling
TestPackage	1.0	☑ Approved	Modeling

Downloaded Assets (0 in the last 28 days)

Asset: TestModel [1.0]

General Details Content Ratings Forums Statistics
Download

Short description: TestModel

Owners: Master Administrator

Community: Modeling

Type: UML Package

Unique ID: {E004116B-7B97-D3CA-7096-682ECFC104B8}

Primary Artifact: TestModel.emx

☑ Approved

My rating ☆☆☆☆☆

Average rating (0 ratings) ☆☆☆☆☆

Discussion topics (0)

Last updated: May 13, 2008

Description

Related Assets

Contain

TestPackage [1.0]

Versions

TestModel [1.0]

Recent Activity

5/13/08 9:37 AM	Master Administrator	☑ State changed from [no state] to Approved
5/13/08 9:37 AM	Master Administrator	Submitted

2 recent events

Asset Governance Process

Roles and Their Activities For Establishing Asset Governance

Asset-based Development Manager

Create Asset Governance Board
Define Asset Gov and Dev Org
Determine Metrics to Track
…

✓ **Metrics and reporting**

✓ **Review and approval**

Asset Governance Board

Define Asset Review Workflows
Define Asset Type
Specify Asset Version Strategy
…

✓ **Information Model**

✓ **Asset Versioning**

Apply the asset governance process to define the policies and artifacts for implementing governance in the repository.

Best practice

Asset-based Development Administrator

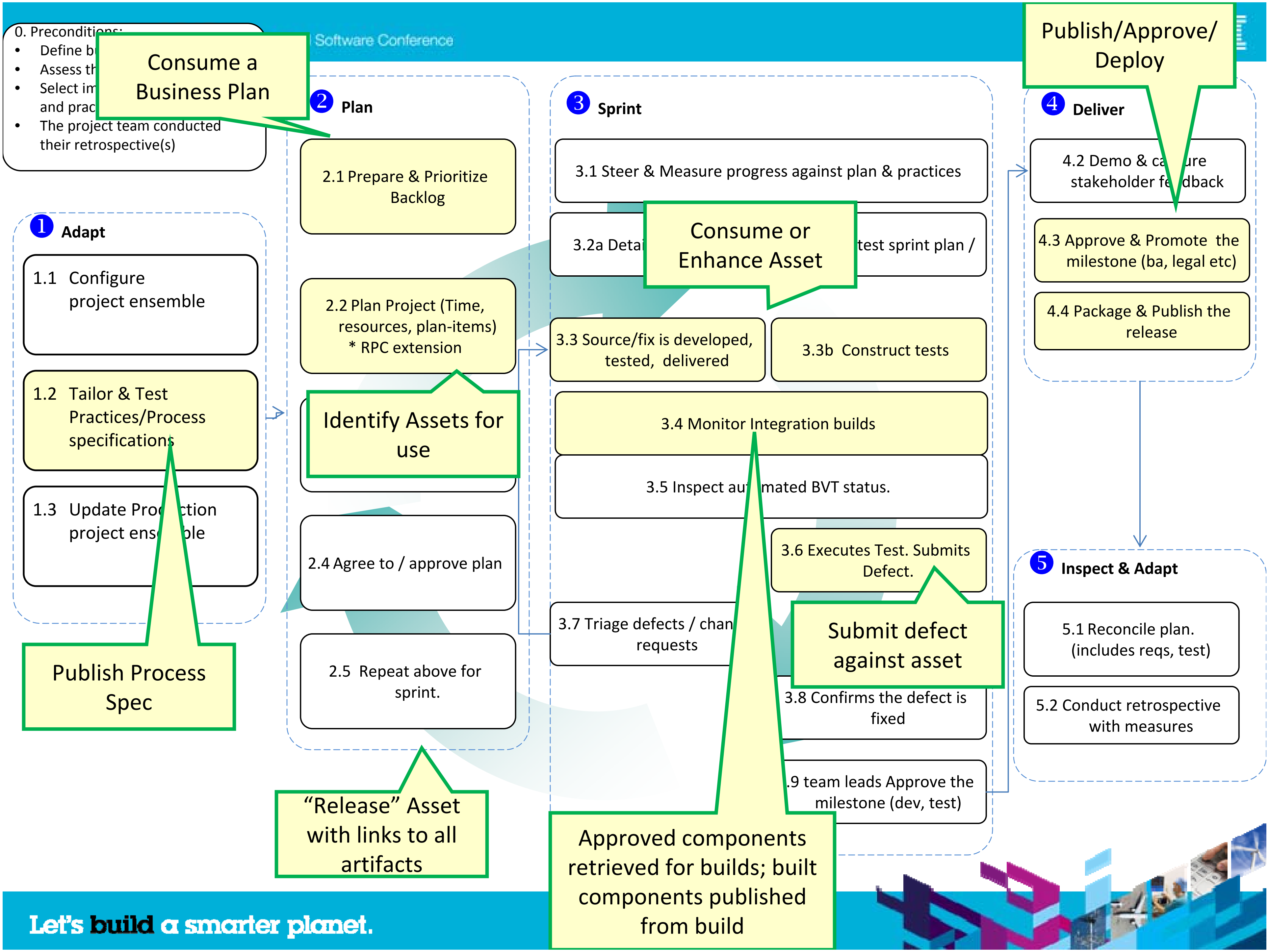
Determine Tools to Support ABD
Specify Repository Community

✓ **Notification, Subscriptions**

✓ **Communities, Access control**

✓ Preparing for Rational Asset Manager governance

Software Conference



0. Preconditions:
- Define b
- Assess th
- Select im and prac
- The project team conducted their retrospective(s)

Consume a Business Plan

1 Adapt

1.1 Configure project ensemble

1.2 Tailor & Test Practices/Process specifications

1.3 Update Prod ction project ens ble

Publish Process Spec

2 Plan

2.1 Prepare & Prioritize Backlog

2.2 Plan Project (Time, resources, plan-items) * RPC extension

Identify Assets for use

2.4 Agree to / approve plan

2.5 Repeat above for sprint.

“Release” Asset with links to all artifacts

3 Sprint

3.1 Steer & Measure progress against plan & practices

3.2a Detai

Consume or Enhance Asset

test sprint plan /

3.3 Source/fix is developed, tested, delivered

3.3b Construct tests

3.4 Monitor Integration builds

3.5 Inspect au mated BVT status.

3.6 Executes Test. Submits Defect.

Submit defect against asset

3.7 Triage defects / chan requests

3.8 Confirms the defect is fixed

.9 team leads Approve the milestone (dev, test)

Approved components retrieved for builds; built components published from build

Publish/Approve/ Deploy

4 Deliver

4.2 Demo & ca ure stakeholder fe dback

4.3 Approve & Promote the milestone (ba, legal etc)

4.4 Package & Publish the release

5 Inspect & Adapt

5.1 Reconcile plan. (includes reqs, test)

5.2 Conduct retrospective with measures

Let’s build a smarter planet.

Summary: End Result when you use Rational Asset Manager in project

- **Supports asset based development, warehousing and asset management**
- **Allows disparate teams to communicate securely and eliminate rework** with asset traceability, categorization and utilization monitoring
- **Implement compliant standards related to your intellectual property**, providing the right level of access to countries, and/or teams.
- Mitigation of **risks** and lower costs of meeting **compliance** mandates by securing **Intellectual property**
- Store information that is useful for collaborating on software development of Assets.
- Provides tools for defining, creating, reviewing and retrieving assets.
- Uses the **Reusable Asset Specification** to describe and classify assets
- **Supports any kind of asset** - source code, DLLs, documentation, presentations, patterns, process flows, etc.
- Enables you to implement asset based development to and succeed with reuse strategies, cutting development costs**. Flexible access to assets via an Eclipse or Web**
- **Integration with other products like RTC, RSA ensure asset based development workflows.**

Questions

www.ibm/software/rational

Managing Assets for Optimal Reuse

- What needs to be done
  - The establishment of
    - **Chains of responsibility** to empower people
    - **Policies** to guide the organization to meet their goals
    - **Control mechanisms** to ensure compliance
    - **Communication** to keep all required parties informed
    - **Measurement** to gauge effectiveness