SERENA

SERENA®
Introduction to Orchestrated ALM

Serena Proprietary and Confidential Information

Publication date: February 2012

Table of Contents

Chapter 1

Welcome to Serena Orchestrated ALM

Introduction to Serena Orchestrated ALM

Serena Orchestrated Application Lifecycle Management (ALM) automates end-to-end application development by helping IT organizations efficiently capture all demand from their stakeholders, manage changes to requirements and software configurations, and deploy releases with confidence. Serena has helped thousands of IT organizations make dramatic improvements to their application development, including greater visibility, faster time to market, higher stakeholder satisfaction, and lower development costs.

Serena Orchestrated ALM is powered by Serena Business Manager (SBM), a powerful platform that allows you to define and automate your organizational workflows. With SBM, you can ensure that all aspects of your development and delivery process are tracked and automated, using fully configurable lifecycles that are modeled after your own processes. The Serena Orchestrated ALM solutions integrate the workflow and automation power of SBM with your critical business systems, including your configuration and demand management systems.

Orchestrated ALM Solutions

Serena delivers Orchestrated ALM with a line of powerful integrated solutions intended to meet the comprehensive needs of different parts of your business. These solutions include Serena Release Manager, Serena Development Manager, and Serena Service Manager.

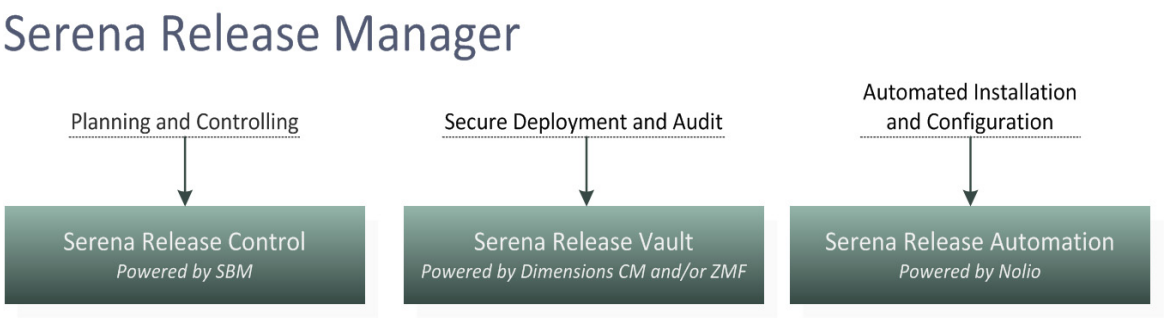
Serena Release Manager

Serena Release Manager helps you manage the flow of change into production. It is the handoff between development, quality assurance, and production operation teams.

The goal of release management is to deploy application changes into production with high quality and without disrupting the business. But this process is often performed manually and is inefficiently connected to the rest of the application lifecycle, leaving a critical gap between application development and operations as well as creating a backlog of changes that must be made.

Serena Release Manager Components

Serena Dimensions CM is an integrated suite that helps you with your release management by enabling you to control, secure, and automate your processes.

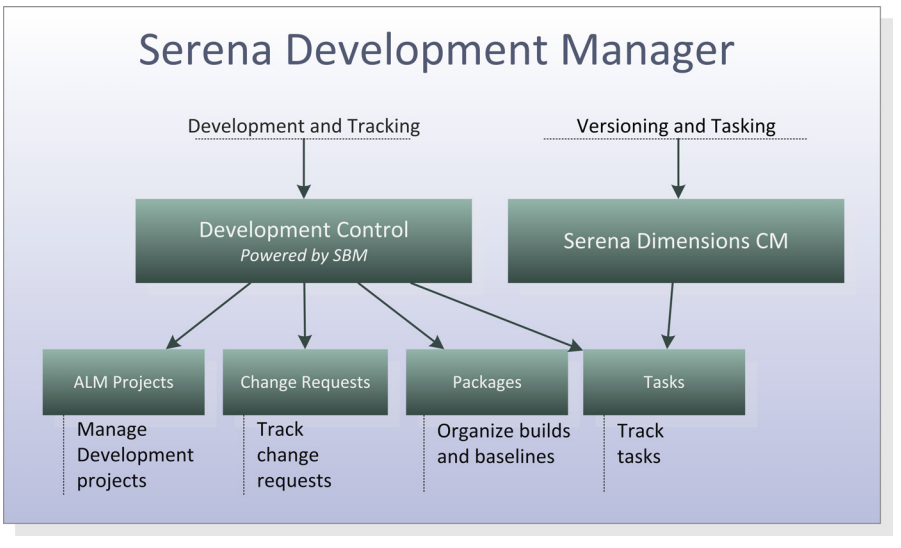


- **Serena Release Control**, powered by Serena Business Manager, helps you plan and control your application release processes across your enterprise, from definition to deployment.
- **Serena Release Vault**, powered by Dimensions CM for distributed systems and ChangeMan ZMF for IBM z/OS systems, ensures a secure and auditable path to production.
  - Secures source for multiple releases
  - Prevents unauthorized changes
  - Enables full traceability and audit trail
- **Serena Release Automation**, powered by Nolio, automates application installation and configuration tasks.
  - Handles a high volume of tasks
  - Cuts deployment time and cost
  - Reduces deployment errors
  - Handles the complexity of multiple release destinations and configurations

Serena Development Manager

Serena Development Manager enables you to orchestrate and monitor your key software development efforts, tracking source code changes and approvals through a central workflow engine. Development Manager uses Serena Business Manager (SBM) to coordinate events across your systems using Web services, integrating application project definition, source code management, test management, and release approvals.

Serena Development Manager Components



Serena Development Manager includes the following:

- ***Development Control,*** a collection of process apps implemented on the Serena Business Manager platform. A process app is a Web application, hosted on a Business Manager server, that you can log into and use from any supported Web browser. With the included process apps, Development Control orchestrates your development work across all of your inter-related systems. The included process apps are:

- **ALM Projects**: Use this app to define and track development projects. This process app sets the workflow for all projects, including states for planning, development, testing, and release. As a project progresses through the workflow, different users work with it using a combination of their own systems.
- **Dev Change Requests**: Define and track development requests. You can relate requests tasks in the Development Tasks process app, which you can then relate to source code assets in Dimensions CM. You can also relate specific types of requests, such as defects, to test cases in your test management system.
- **Dev Tasks**: Define and track development tasks. Manage tasks by relating them to change requests. Track source code changes by relating tasks to source files in Dimensions CM.
- **Dev Packages**: Define and manage development packages. Integrate with your SCM tool to compile and build source code, and integrate with your test management system to test builds before preparing them for release.

- ***Serena Dimensions CM***, a rich enterprise-class version and source code configuration management solution. Dimensions CM provides a wide variety of clients and usage models, with Web and desktop clients, as well as complete scripting and API support. Development Control integrates with Dimensions CM via Web services, enabling you to associate source code assets in the Dimensions CM repository with your tasks in Development Control.

Serena Dashboard

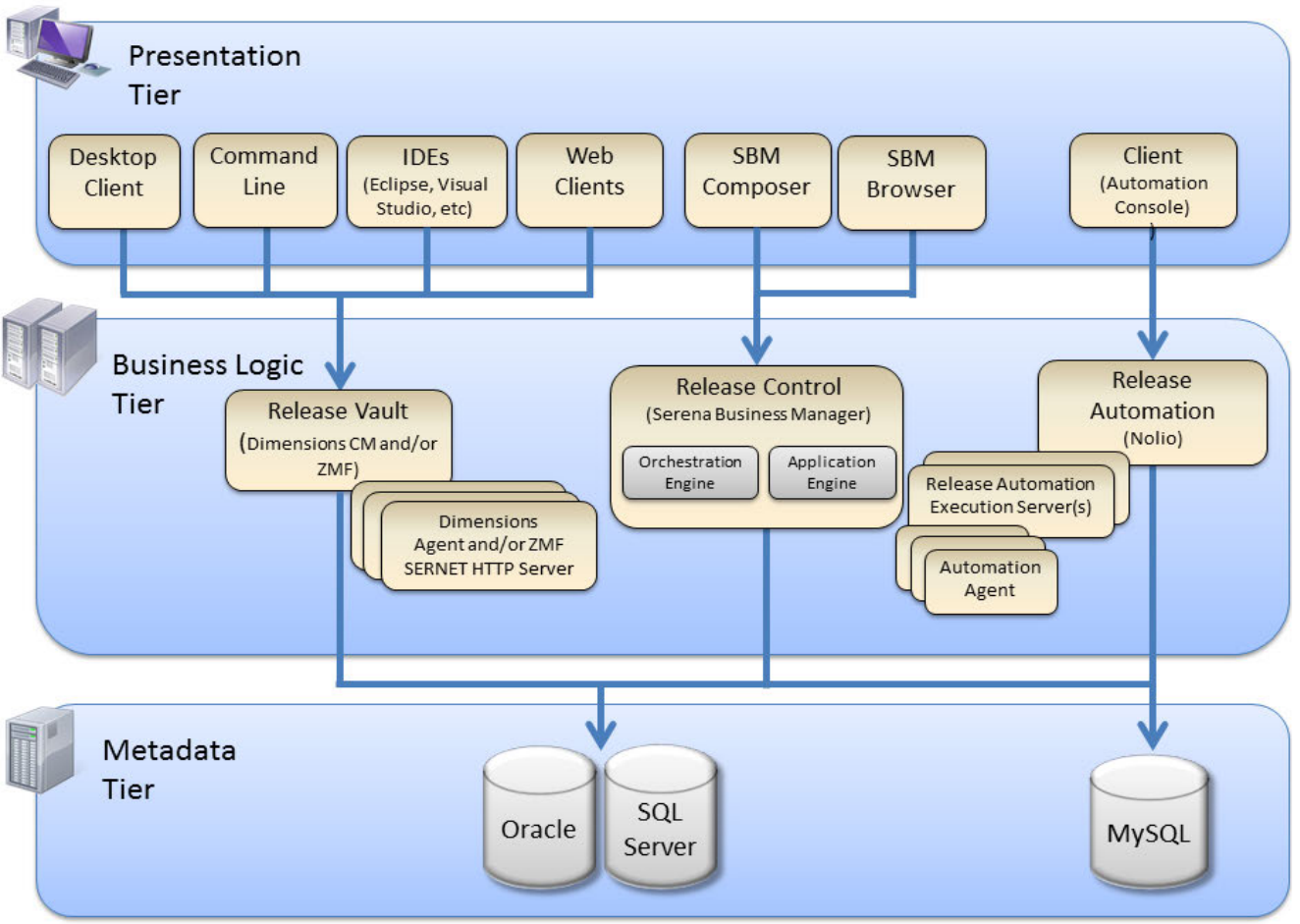
Serena Dashboard is a powerful reporting application that allows users at all levels of the organization to review project metrics that are most immediately relevant to them. Serena Dashboard runs on the Information Builders WebFOCUS platform, a rich business reporting tool that can aggregate key performance data from a variety of critical systems. Serena Dashboard presents data on your most essential key performance indicators (KPIs) with a fully configurable set of graphical charts, tabular data, and more.

Orchestrated ALM System Architecture

The Orchestrated ALM solutions can generally be represented in three tiers: presentation, business logic, and metadata. At the presentation level are the SBM and other clients, such as Dimensions CM. At the business logic layer are the application servers for SBM, Dimensions CM, ZMF, and other integrated systems. At the metadata layer are the database backends. To clarify how these systems work together in each solution, lets take a closer look at the architecture of each.

Serena Release Manager System Architecture

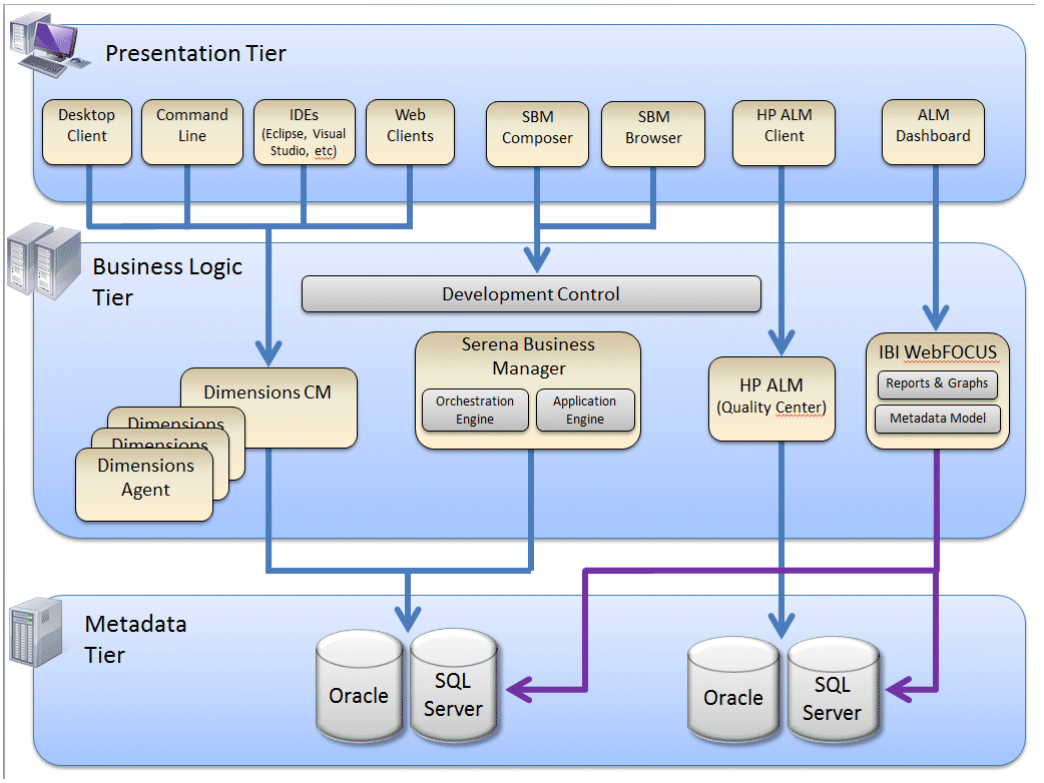
The architecture of the Serena Release Manager suite integrates Serena Release Control, Serena Release Vault, and Release Automation as shown in the following figure.



The integrated power of Release Control, Release Vault, and Release Automation are brought together through the Serena Release Control user interface and its underlying SBM infrastructure.

Serena Development Manager Architecture

The overall architecture of Serena Release Manager is illustrated in the diagram below.



Chapter 2

Serena Orchestrated ALM End-to-End Scenario

About the Serena Orchestrated ALM End-to-End Scenario

Working closely with our users, Serena has built an end-to-end example scenario that represents our ALM best practice recommendations. This scenario addresses the various roles and dependencies throughout the IT software development lifecycle. Although you will find references to Serena solutions throughout the scenario, it is extensible and can apply to 3rd party applications as well.

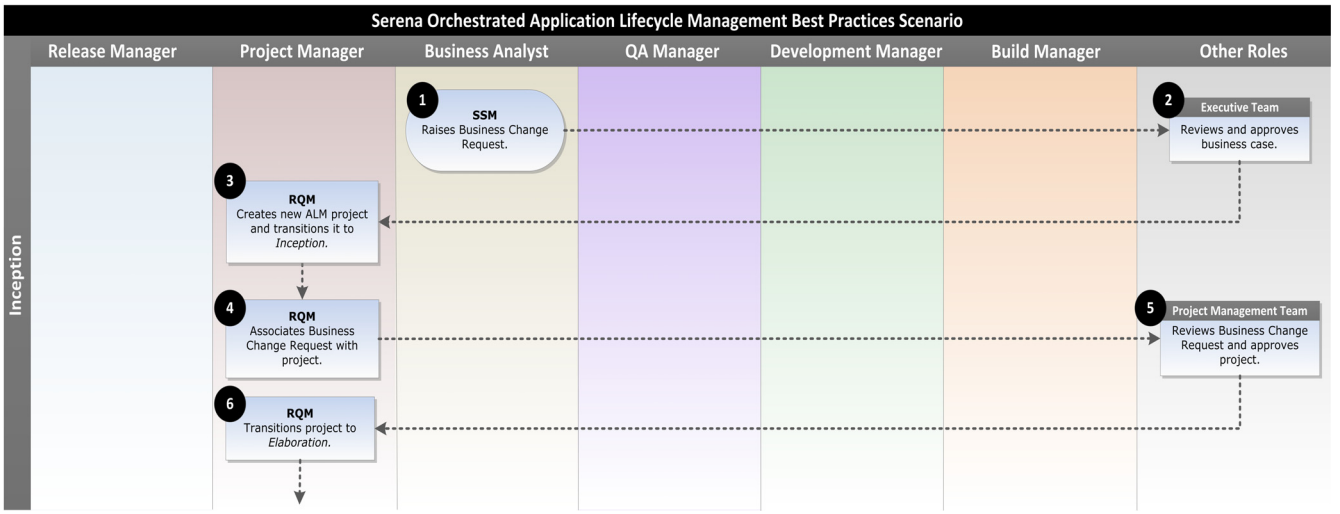
We recommend that you review and consider this scenario as a starting point for understanding how your processes might best map to Serena best practices as implemented by Serena Orchestrated ALM solutions.

This scenario comprises the following high level project lifecycle stages:

- **Inception**
- **Elaboration**
- **Construction**
- **Transition & Completion**

We shall walk through each of these in turn.

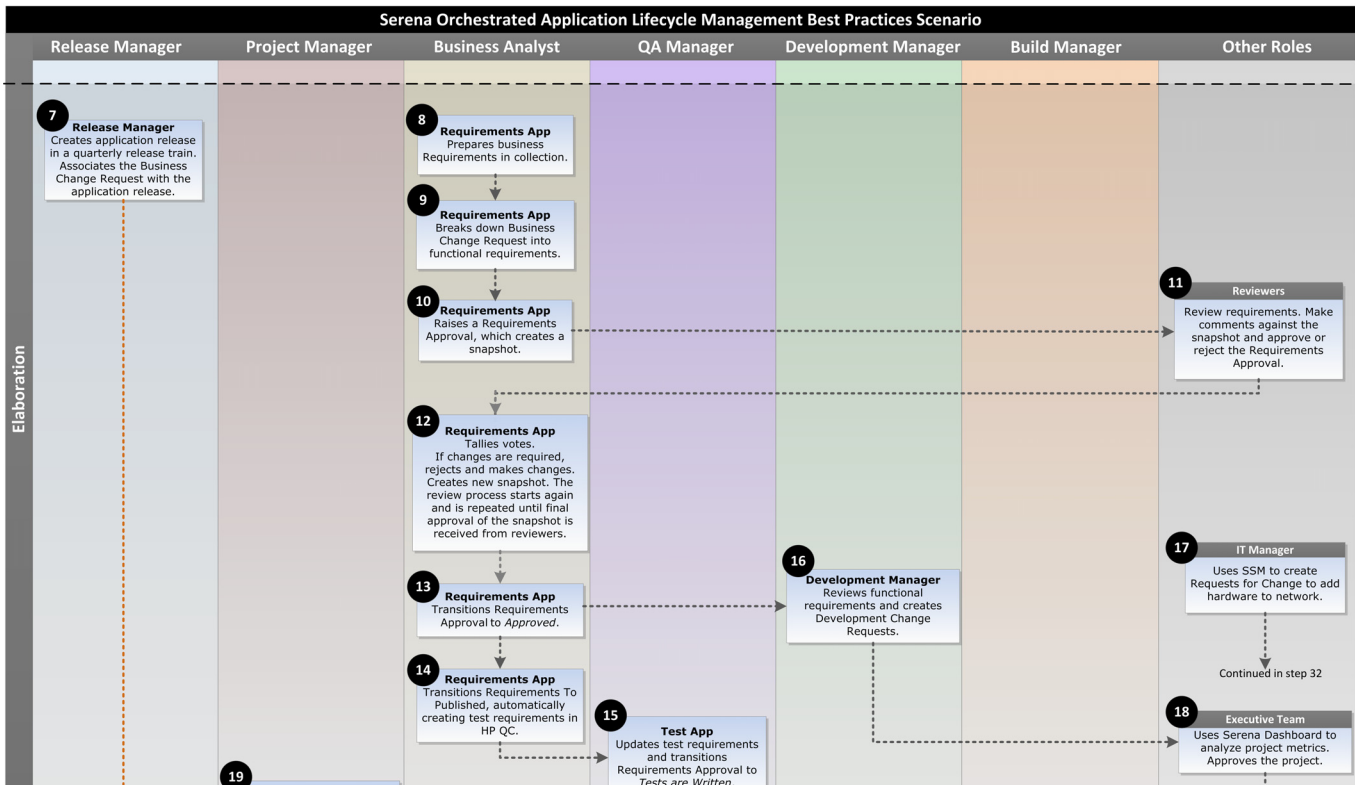
Inception



Step	Description
1	The business analyst uses Serena Service Manager to raise a new Business Change Request to modify an application.
2	The business case for the new version of the application is reviewed and approved by all the relevant stakeholders, for example, the Executive team.

Step	Description
3	The project manager uses Serena RQM to create a new Application Lifecycle Management (ALM) project and transitions it to the Inception state.
4	To address the Business Change Request in a specific project, the project manager uses RQM to associate the Business Change Request with the ALM project.
5	The project management team reviews the Business Change Request and approves the project.ce
6	The project manager uses RQM to transition the project to its next state, Elaboration.

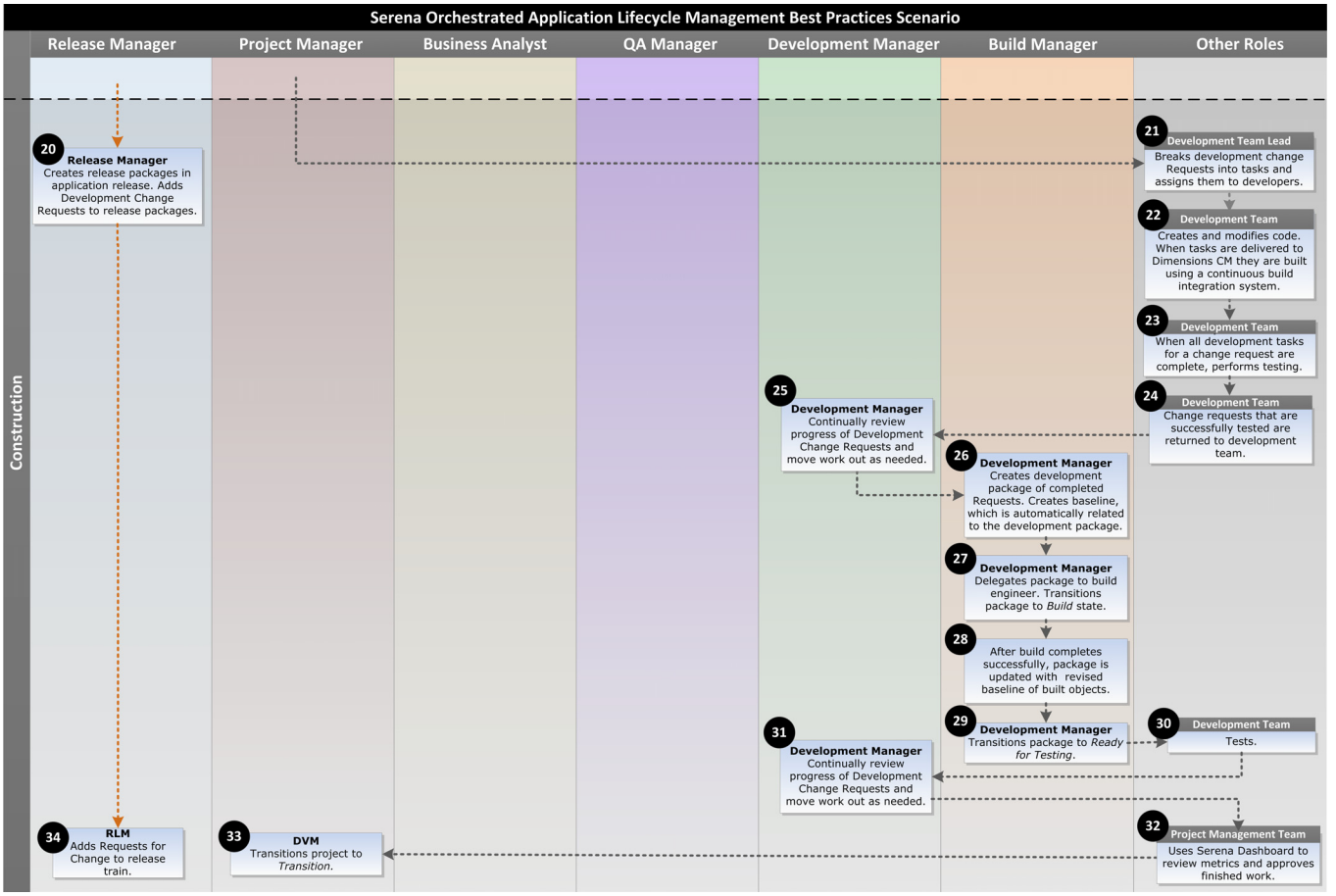
Elaboration



Step	Description
7	To deliver the changes in a specific quarter, the release manager uses Serena Release Manager to create an application release in the appropriate quarterly release train, and associates the Business Change Request with the application release.
8	The business analysis creates a new collection of requirements for the release using the Requirement Management application, such as Dimensions RM.
9	The business analyst the breaks down the Business Change Requests into various classes of functional requirements.

Step	Description
10	The business analyst raises a requirements approval request, which in turn creates a snapshot of the requirements collection.
11	The reviewers of the functional requirements receive notification from RQM and open a web link to a snapshot in RQM. They make review comments against the snapshot and the requirements, and approve or reject the Requirements Approval ticket.
12	The business analyst tallies the reviewers' votes and comments. If changes are required to the snapshot, she rejects the Requirements Approval ticket and makes the changes. She creates a new snapshot of the document, which creates a new Requirements Approval ticket in RQM. The review process starts again. The approval process is repeated until final approval of the snapshot is received from all the reviewers.
13	The business analyst uses RQM to transition the Requirements Approval ticket to the next state, *Approved*, which starts the testing phase.
14	To provide the QA team with test requirements, the business analyst then publishes the requirements which automatically creates test requirements in HP Quality Center.
15	Quality management assets are managed by HP Quality Center or other testing tools. The QA Manager updates test requirements and adds further details. The test requirement definitions are based on the detailed functional requirements. The QA team turns test requirements into test cases and groups them into test suites. The QA Manager transitions the Requirements Approval ticket to its next state, *Tests are Written*.
16	To enable the development team to understand the work that will be required, the development manager uses Serena Development Manager to review the detailed functional requirements and creates Development Change Requests.
17	The IT infrastructure needs to be upgraded to support the anticipated increased traffic to the online application. Therefore, the IT Manager uses Serena Service Manager to raise Requests For Change to install new switches and servers.
18	The Executive management team uses Serena Dashboard to review the project. For example, are there sufficient resources, documentation, and test plans? Are the requirements met? The project is approved for development.
19	The project manager uses Serena Development Manager to transition the project to the next state, Construction.

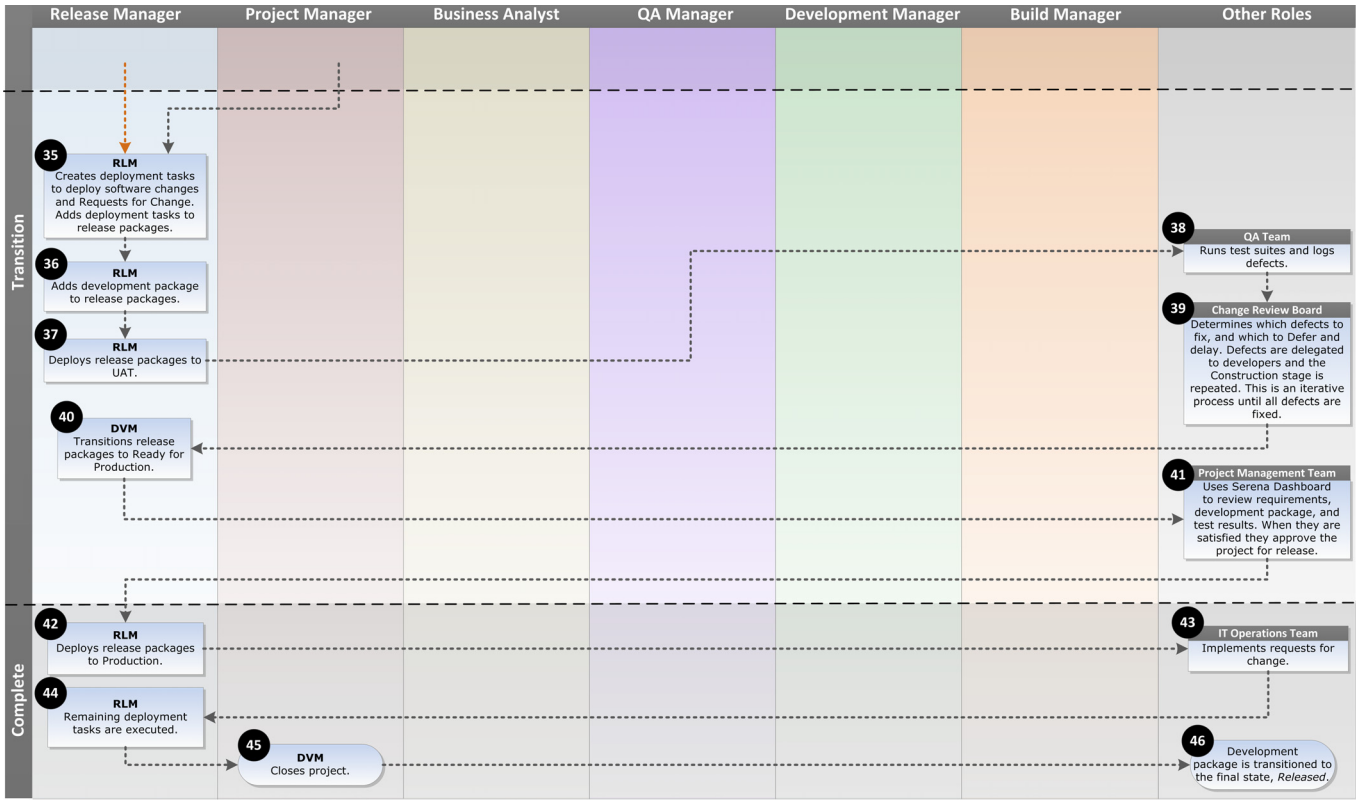
Construction



Step	Description
20	To plan the contents of the release, the release manager creates release packages in the application release and adds the Development Change Requests to the release packages.
21	To assign the work correctly to developers, the development team lead uses Serena Development Manager to break the requests into development tasks. As they are created, the tasks are synchronized with Dimensions CM task requests. When the tasks are ready to be worked on, the team lead moves them into development and assigns them to individual developers.
22	The developers create and update source code using their preferred development environments. Development changes are delivered to Dimensions CM against the tasks. After the tasks are delivered they are built and unit tested using a continuous build integration system, such as Hudson.
23	When all of the development tasks for a change request are completed, the development team performs testing.
24	Change requests are transitioned back to the development team after they have been successfully tested.

Step	Description
25	The development manager and project manager use Serena Development Manager to continually review the list of requests in progress and move work out of the project as needed based on remaining time and business priorities.
26	To build the completed Development Change Requests, the build manager uses Serena Development Manager to create a development package of completed requests. The development package is used to take the changes through the formal build process. The stream where the development work is taking place is automatically related to the package. A baseline of that stream is created and is automatically related to the development package. The baseline assembles all of the source code changes associated with completed Development Change Requests.
27	The build manager delegates the development package to a build engineer and transitions it to its next state, *Build*.
28	After the build has completed successfully, outputs are automatically collected back into Dimensions CM against a build request, and a revised baseline is produced. The development package is automatically updated with the revised baseline of built objects. This is an iterative cycle until all development and build work is successfully completed.
29	The build manager uses Serena Development Manager to transition the development package to its next state, *Ready for Testing.*
30	The development team performs testing. When the testing is complete, the development manager transitions the development package to its next state, Complete.
31	The development manager and project manager use Serena Development Manager to continually review the list of requests in progress and move work out of the project as needed based on remaining time and business priorities.
32	The project management team uses Serena Dashboard to review metrics, such as build successes, and approves the finished work.
33	The project manager uses Serena Development Manager to transition the project to its next state, *Transition*.
34	To ensure that the IT operations team upgrades the hardware for this release, the release manager adds the Requests For Change to the release train.

Transition & Completion



Step	Description
35	The release manager creates deployment tasks and adds them to deployment process templates. The tasks will be used to deploy the software changes and Requests For Change to deployment areas. She then uses the templates to add deployment tasks to the release packages.
36	To pick up the built software from the development team, the release manager adds the development package to the release packages.
37	The release manager deploys the release packages to the UAT state, which automatically runs the deployment tasks
38	The QA team runs tests in the UAT area, which generates a set of test run results. Defects are logged in HP Quality Center.
39	The change review board determines which defects to fix. Deferred defects are moved out of the project to be fixed at a later date. Defects that need to be fixed are automatically created in Serena Development Manager and delegated to developers. The Construction state is repeated. This review process is repeated until all the defects in the project are fixed.
40	When testing is complete, the development manager transitions the development package to its next state, *Ready for Production*.

Step	Description
41	The project management team uses Serena Dashboard to review the requirements, the development package, and the test results. When they are satisfied they approve the project for release to production.
42	The release manager deploys the release packages to deployment.
43	The IT Operations team implements requests for change.
44	The remaining deployment tasks are executed.
45	The project manager closes the ALM project in Development Manager.
46	The package is transitioned to its final state, *Released*.

Chapter 3

Serena Release Manager Objects and Workflows

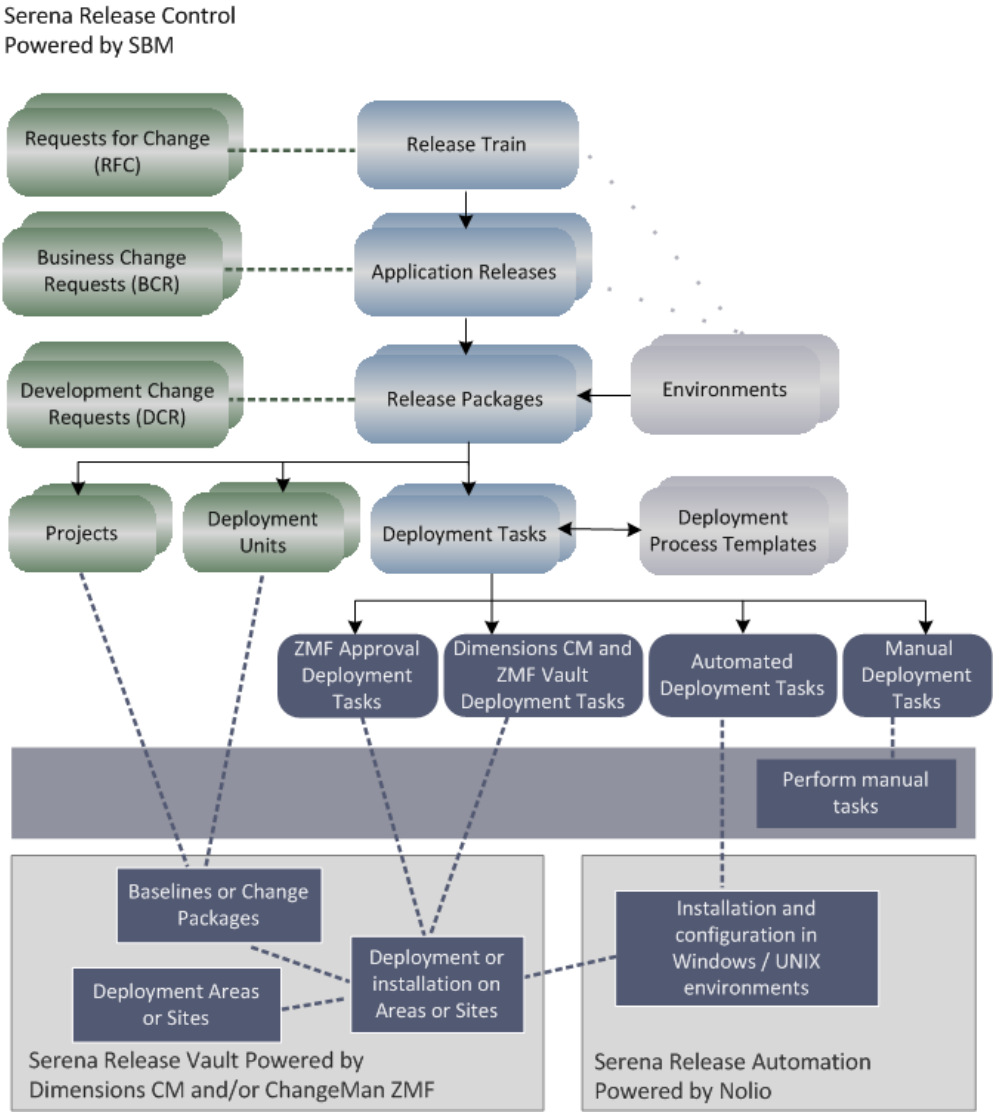
About Serena Release Manager Workflows

In this chapter we will discuss:

- The different types of objects within Serena Release Manager, and a typical workflow for using the different parts of the system
- The specific workflows implemented within SBM for the various process apps that are included with the solution

Serena Release Manager Object Hierarchy Overview

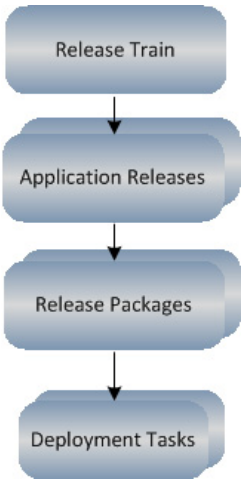
Together, the primary and related entities of Serena Release Control implement a rich release management solution that is highly configurable and customizable to meet your organization's release management needs. The full default Serena Release Control hierarchy and related entities are shown in the following figure.



We will walk through each aspect in turn.

Core Release Management Objects

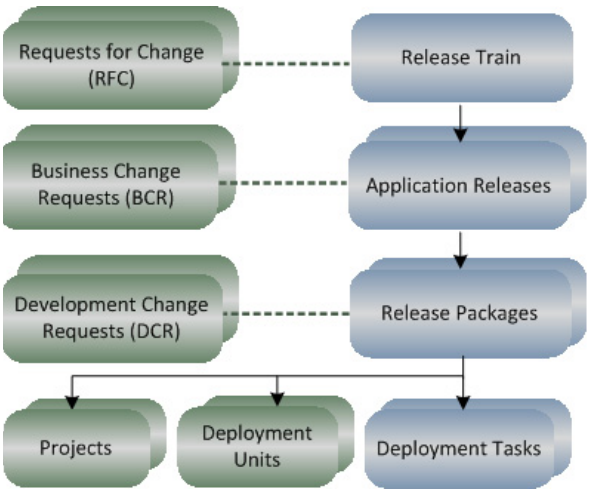
The hierarchical relationships between the primary Serena Release Control objects are shown in the following figure.



- **Release Trains** provide a published schedule of changes to production. One or more application releases are associated with each release train.
- **Application Releases** represent versions of applications or projects, where the application or project architecture is specified by components. One or more release packages are associated with each application release.
- **Release Packages** represent a portion of IT or service infrastructure normally built, deployed, tested, and released together. Release packages define the set of changes to be deployed and drive the deployment processes. One or more development change requests and deployment units are associated with each release package.
- **Deployment Tasks** are actions to be executed as part of the deployment process to deploy a Release Package into a specific environment. Deployment task types include manual, vault, and automation.

Integrated Project and Change Management

Serena Release Control brings in information from integrated providers as needed to support your release control processes. Providers may include SBM, Serena Service Manager, Dimensions CM, and ChangeMan ZMF.



- **Requests for Change** represent operational changes that may affect multiple applications or implement system infrastructure changes within an enterprise. These are associated with release trains and are typically represented as tickets from service management systems such as Serena Service Manager.
- **Business Change Requests** represent customer or business unit change requests that affect specific application releases. They are associated with application releases and are typically represented as tickets from help desk or incident management systems such as Serena Service Manager or other SBM solutions.

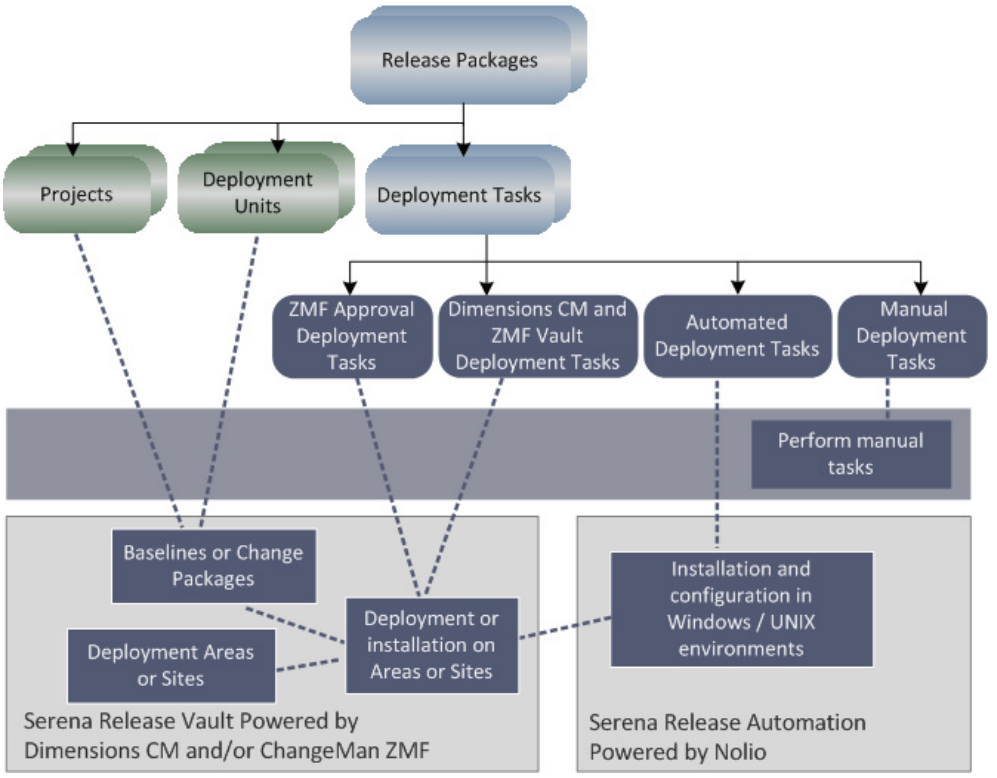
- **Development Change Requests** represent delivered changes from the development process and affect specific release packages. They are associated with release packages and are typically represented as tickets from development management systems such as the SBM Change Manager solution or Dimensions CM request management.
- **Projects** are the organizing entity that Serena Release Control implementation uses to get a collection of related objects and associate them with the primary Serena Release Control objects. For example, Dimensions CM, SSM, and SBM projects, and ChangeMan ZMF applications (projects) are used to get information on the other integrating objects.
- **Deployment Units** represent a set of deployable components, such as Dimensions CM baselines with build outputs and ChangeMan ZMF change packages.

Deployment Relationships

To secure your release deployment and provide an audit trail of changes, Serena Release Manager uses the release vault features of Dimensions CM and ChangeMan ZMF. To automate your release installation and configuration tasks, the default implementation of Serena Release Manager uses Serena Release Automation, powered by Nolio.

Release Packages point to deployment tasks that integrate with release vault and release automation features of integrating products. These deployment tasks automate deployment, installation, and configuration tasks as part of the deployment process.

The flow of information between release control, release vault, and release automation is shown in the following figure.



Serena Release Vault not only secures the code, it also copies the deployment units to the environments defined for the deployment units. For example, when using Dimensions CM as the Deployment Unit Provider, baselines are copied to the deployment areas defined for stages in the Global Stage Lifecycle.

The **release types** and **release stages** define the purpose of the environments where the **deployment units**, such as Dimensions CM baselines and ChangeMan ZMF change packages, are to be copied. The **deployment tasks** define other actions to be done for the deployment, such as approval, vault tasks, and automation tasks. Deployment tasks point to **deployment areas** in Dimensions CM and **sites** in ChangeMan ZMF to specify the exact destinations for the deployment units.

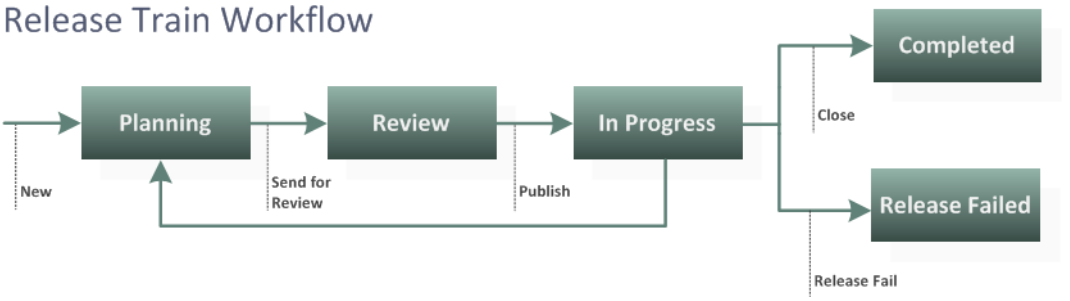
Serena Release Automation performs the installation and configuration tasks once the deployment units are deployed to a specified deployment area or site for a release stage. For example, for each stage, it might update the registries, update configuration files, and restart Web services according to the deployment task definitions. Later, once a release has reached the production deployment stage, Serena Release Automation could be used to distribute files from a production environment to a large number of designated servers, updating the registries, updating configuration files, and restarting Web services for all of these.

Detailed Release Manager Workflows

We will now look at some detailed example workflows for core objects ini Serena Release Manager, such as release trains, releases,

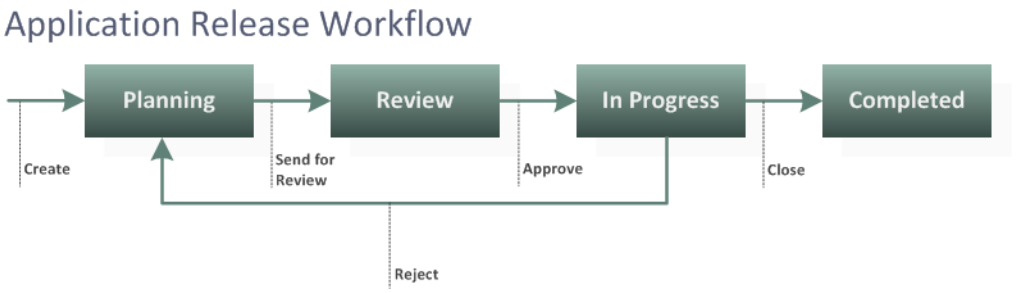
Release Trains

As you work with release trains, you are presented with buttons on the user interface for workflow actions that are available to you. A simplified representation of the default workflow for a release train is shown in the following figure.



Application Releases

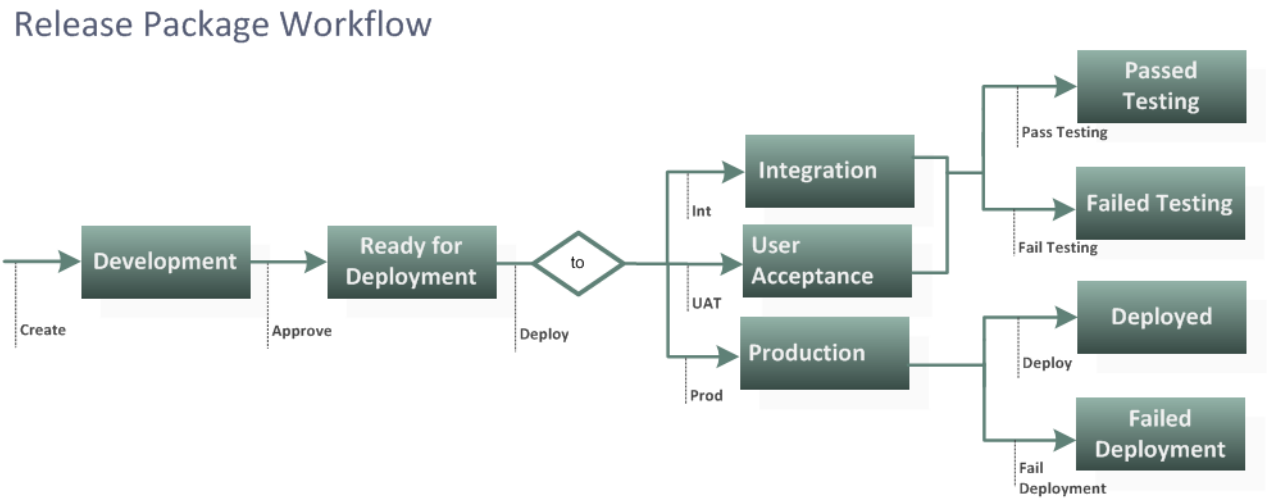
As you work with application releases, you are presented with buttons on the user interface for workflow actions that are available to you. A simplified representation of the default workflow for an application release is shown in the following figure.



Release Packages

As you work with release packages, you are presented with buttons on the user interface for workflow actions that are available to you. The release package workflow is a conditional workflow that allows for default paths for each release type, Major, Minor, and Emergency. The workflow actions available for each release type are the actions needed to support the release stages for that release type.

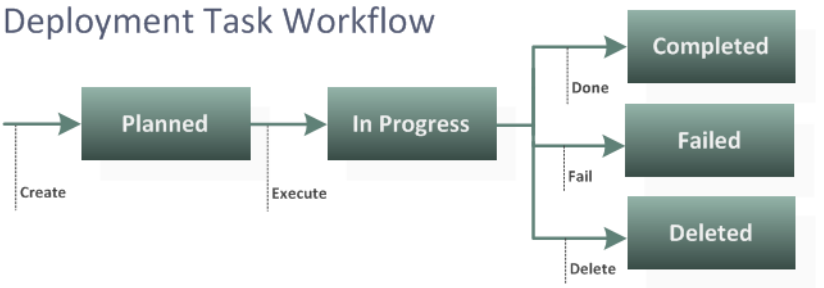
A simplified representation of the default workflow path for a major release package is shown in the following figure.



Deployment Tasks

As you work with deployment tasks, you are presented with buttons on the user interface for workflow actions that are available to you. There are separate workflows for each type of task.

The default workflow for a manual deployment task is shown in the following figure.



Chapter 4

Serena Development Manager Objects and Workflows

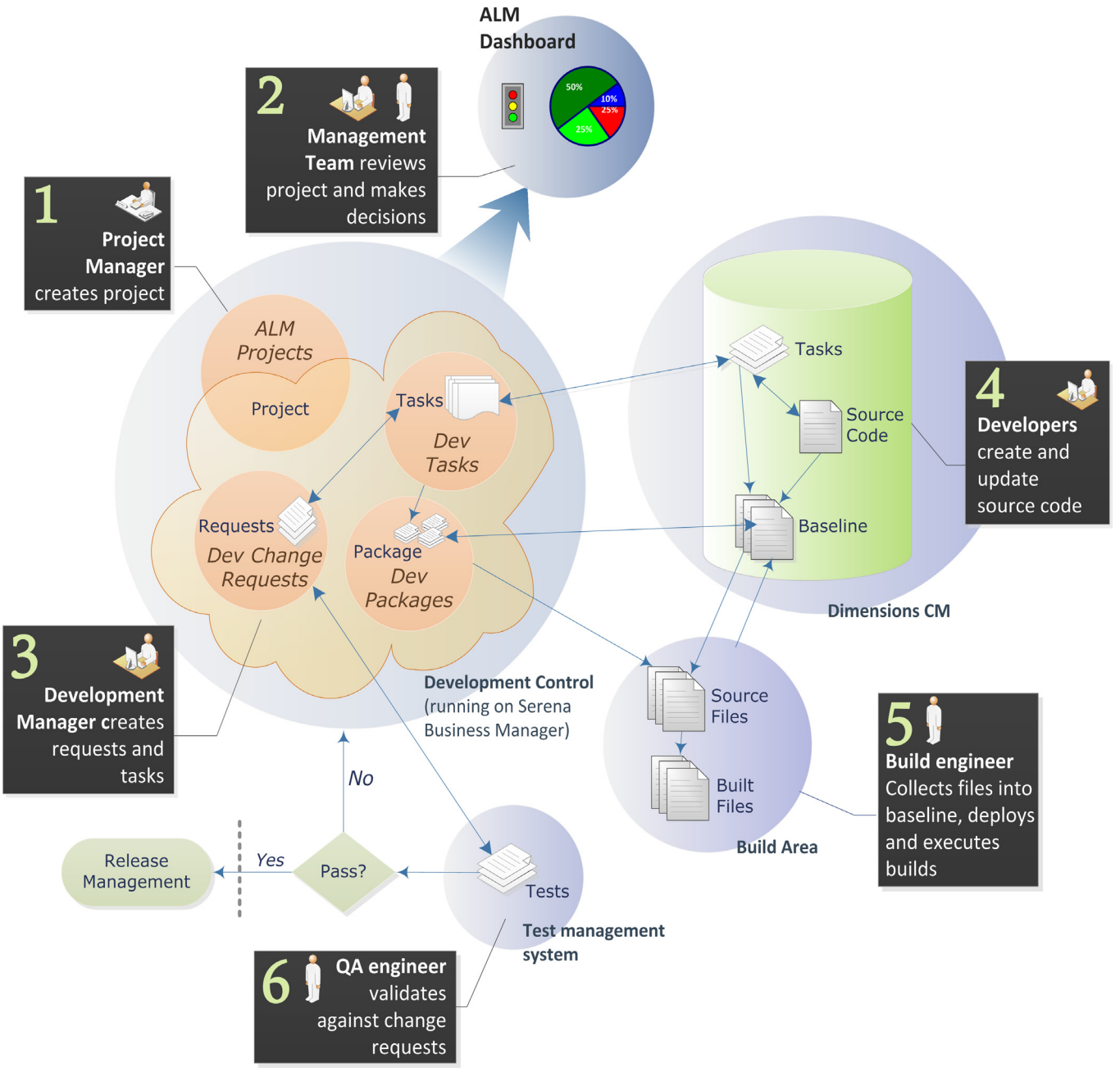
About Serena Development Manager Workflows

In this chapter we will discuss:

- The different types of objects within Serena Development Manager, and a typically workflow for using the different parts of the system
- The specific workflows implemented within SBM for the various process apps that are included with the solutions

Serena Development Manager Usage Overview

The components included with Serena Development Manager work together as defined in the following diagram. This describes a standard configuration, as well as the order in which users interact with it. This only focuses on the end user scenario; configuration is addressed in the *Serena Development Manager Installation and Configuration Guide*.


ALM
Dashboard
50%
10%
25%
25%
2
Management Team reviews project and makes decisions
1
Project Manager creates project
ALM Projects
Project
Tasks
Dev Tasks
Requests
Dev Change Requests
Package
Dev Packages
Development Control
(running on Serena Business Manager)
Tasks
Source Code
Baseline
Dimensions CM
4
Developers create and update source code
3
Development Manager creates requests and tasks
No
Source Files
Built Files
Build Area
5
Build engineer
Collects files into baseline, deploys and executes builds
Release Management
Yes
Pass?
Tests
Test management system
6
QA engineer validates against change requests

Lets step through this diagram.

Step	Description
1	At the beginning of a project, a project manager uses the ALM Projects process app in Development Control (running on Serena Business Manager) to create a new project. This is typically in response to incoming demand, such as requirements from a business analyst for a new feature, or a customer request or defect. For details on the Development Project workflow, please see "ALM Projects Workflow" on page 20. For details on creating projects, please see "Creating and Managing Projects" on page 30.
2	At any point during a project's lifecycle, leads, managers, executives, and others may consult Serena ALM Dashboard to review project status and key performance indicators (KPIs). The reports displayed here may help decision makers choose the correct path forward when work must be prioritized or re-evaluated. For more on viewing and customizing ALM Dashboard views, see "Displaying Project Metrics with the ALM Dashboard" on page 45.
3	The Development Manager creates change requests using the Dev Change Requests process app - or from the project in ALM Projects. The change requests describe the features and other work to be implemented. The change requests are related back to the project, ensuring complete traceability of work. The Development Manager and others also create tasks using the Dev Tasks process app. Tasks can be used to split the work into more manageable units, that can be assigned to individual developers. When you create a development task, the task is synchronized to Dimensions CM, and a new request of type Task is created. To learn about the detailed workflow for change requests and tasks, please see "Dev Change Requests Workflow" on page 21 and "Dev Tasks Workflow" on page 22. For information on creating change requests, please see "Creating and Working on Change Requests" on page 34. For information on working with tasks, please see "Creating and Working on Tasks" on page 37.
4	Developers update the source code using their source control environment. Serena Development Manager includes Serena Dimensions CM. Tasks in Development Control are synchronized to Dimensions CM, and information on all work on files in Dimensions CM is stored in Dimensions CM tasks. This information is then synchronized back to the originating tasks (which are in turn related back to the originating change requests) in Development Control, ensuring a complete audit path of all work completed in context of a project.

Step	Description
5	As work progresses on the project, the build engineer sets up packages using the Dev Packages process app in Development Control. The build engineer relates the packages to baselines in Dimensions CM that collect all of the files associated with change requests in the project and deploy them to a build area. Using these files, the build engineer runs a build and installs it for testing purposes. To learn more about the workflow for the Development Baselines process app, see "Dev Packages Workflow" on page 24. For details on creating and working with development packages, see "Creating and Working with Development Packages" on page 40.
6	Using HP Quality Center / ALM, the QA staff tests the builds, both the nightly builds and release candidate builds. Defects may be tracked in the Dev Change Requests process app, and related to defects in Quality Center. Failed tests are returned to Development Control and the original change requests are returned to developers to fix. When a release candidate build passes testing, the build is turned over to the release engineer who will use a release management solution, such as Serena Release Manager, to deploy the build into all of the required environments. For details on implementing the SBM Connector to Quality Center, see the *SBM Connector for HP Quality Center / ALM Implementation Guide*.

Detailed Development Manager Workflows

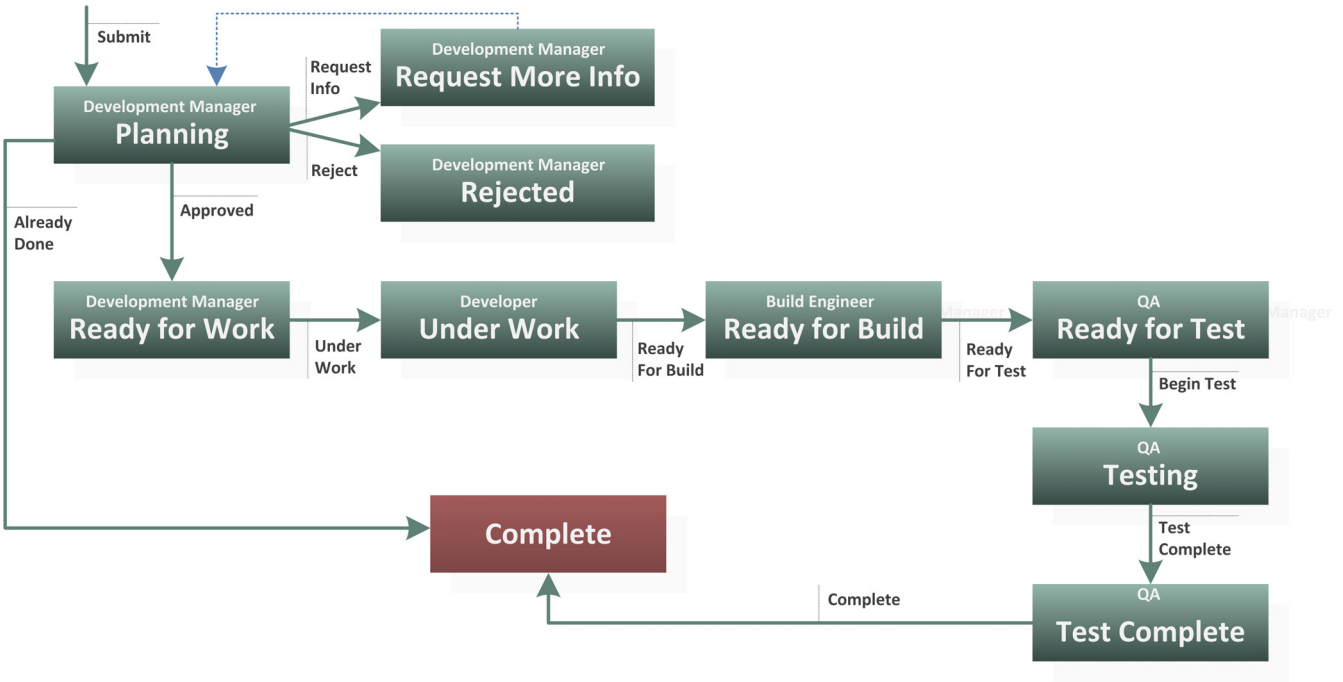
ALM Projects Workflow

ALM Projects provides a high level view of the overall project workflow and status.

Dev Change Requests Workflow

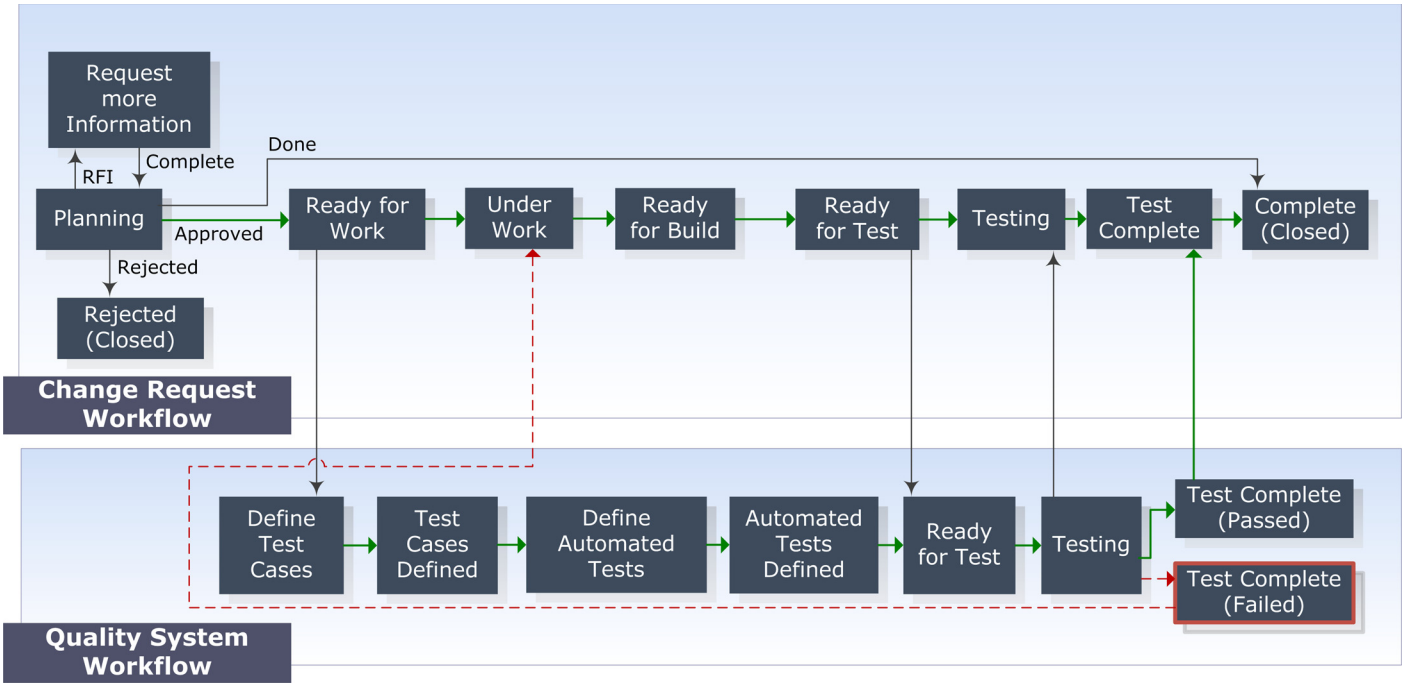
Change requests follow the workflow illustrated below. This workflow is implemented in the Dev Change Requests process app. Note that most work on change requests is expect to happen during the **Develop & Test** state in a project.

Development Change Requests Workflow



Integrating Change Requests with HP Quality Center / ALM

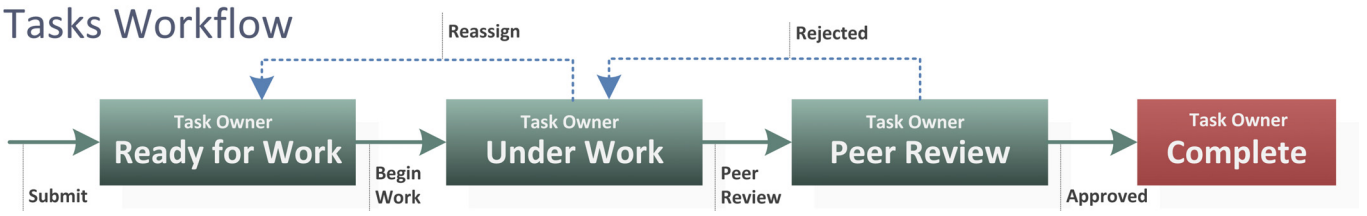
The following diagram illustrates the out-of-the-box integration points between Dev Change Requests and HP Quality Center / ALM.



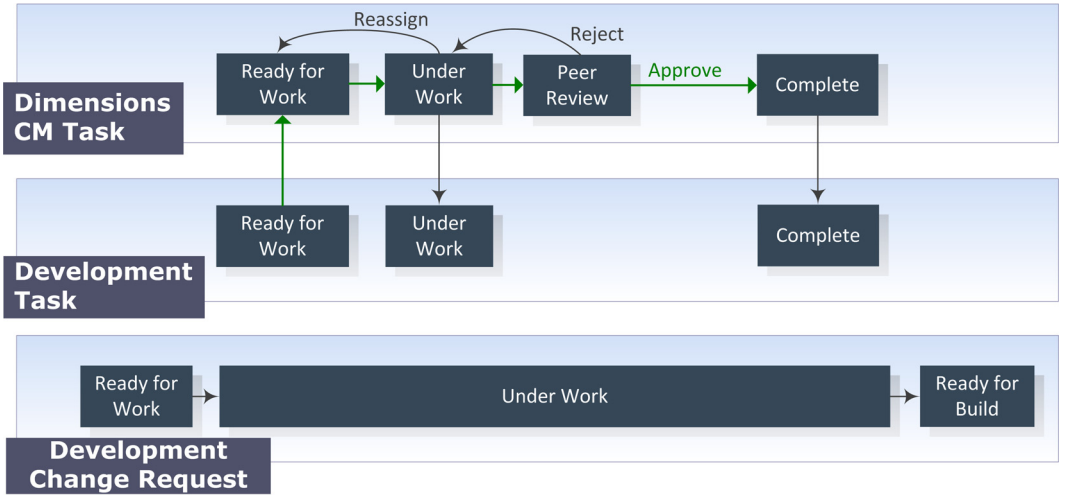
Dev Tasks Workflow

Manage specific developer tasks using the Tasks process app. You start and complete work on a task during the **Under Work** state of the change request that owns it. Development Managers create tasks to break work associated with a change request down into more manageable units that can be assigned to individual developers.

Tasks follow the workflow illustrated below. If you are using Dimensions CM, these states map to parallel states for change requests in Dimensions CM.



The following illustration clarifies the relationship between tasks in Dev Tasks and related tasks in Dimensions CM. Note that this is just one example of how tasks in Development Control might be linked to tasks in Dimensions CM for purposes of tracking work.



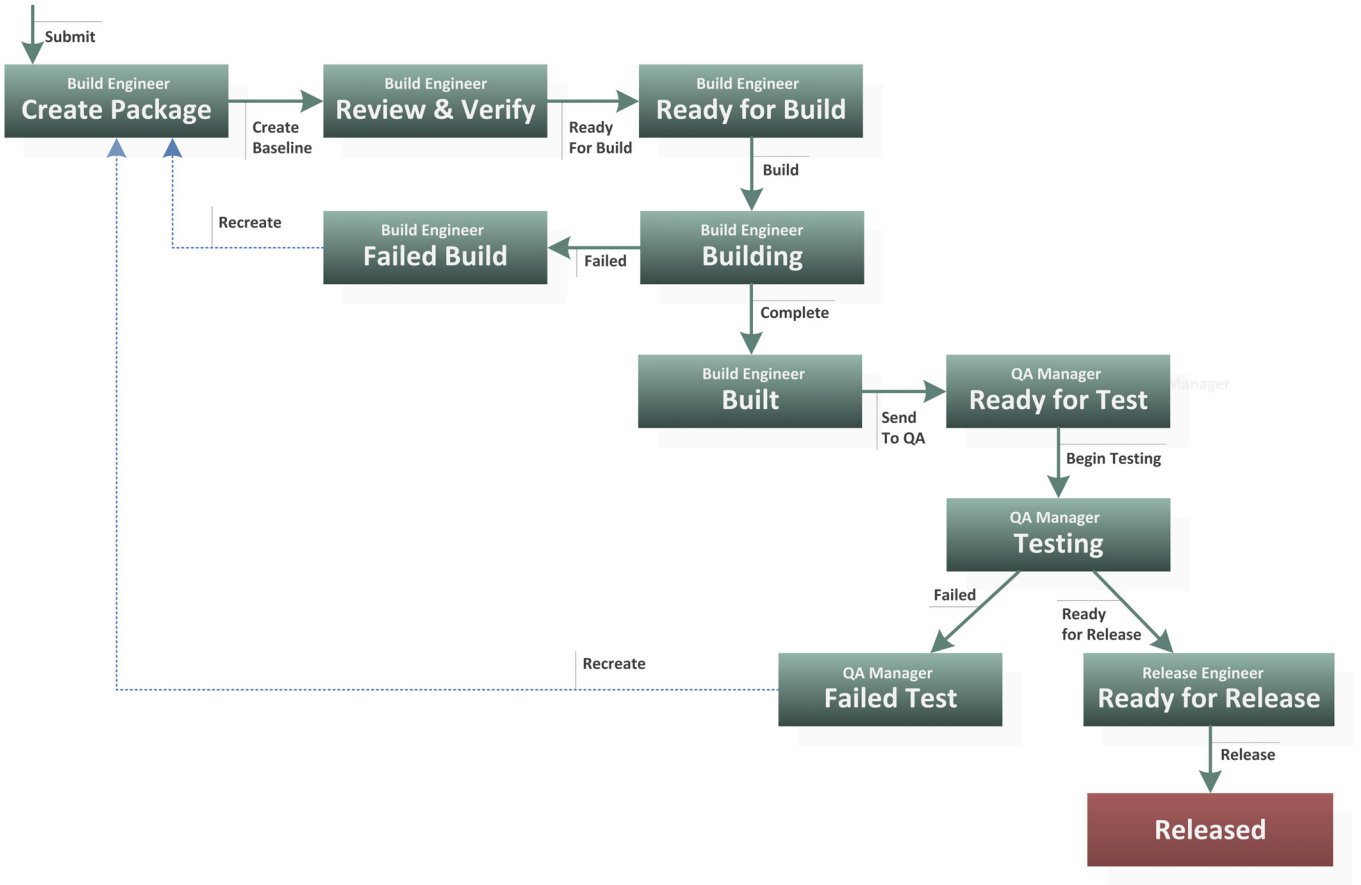
Dev Packages Workflow

Use the Dev Packages process app to manage the creation and validation of release packages and baselines, including testing builds and approving for final release. Most of the work managed within the Dev Packages process app takes place during the **Product Test** project state.

The Dev Packages workflow provides integration points into Dimensions CM, as well as Quality Center. Use Dev Packages to fully automate the build and validation process. Start by building from baselined sources and finish by passing the final build off to your release management system, such as Serena Release Manager.

The following diagram illustrates the workflow for development packages as implemented by the Dev Packages process app in Serena Development Manager.

Development Packages Workflow



The following diagram illustrates the integration points between the Dev Packages process app and Dimensions CM. This is based on the out-of-the-box configuration. In this example, a baseline in Dimensions CM follows a separate workflow. As the package progresses through its workflow states, information and state are synchronized to the related baseline in Dimensions CM.