SERENA®

DEVELOPMENT MANAGER

Getting Started Guide

Serena Proprietary and Confidential Information

Trademarks

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 1850 Gateway Drive, 4th Floor, San Mateo, California, 94404-4061.

Product version: 4.5

Publication date: 2013-05-08

Table of Contents

Welcome to Serena Development Manager

Thank you for choosing Development Manager to plan and control your development.

Serena® Development Manager (Powered by Serena Business Manager) enables you to plan, control, and automate all your development processes from definition to quality assurance and release approval with start-to-finish traceability and end-to-end visibility across distributed environments.

Audience and Scope

This document is intended for personnel who participate in the processes of managing development using Serena Development Manager.

Before You Begin

See the Readme for the latest updates and known issues.

Accessing the Documentation

You can access all documentation and demonstrations for the current release of Serena Development Manager through the Serena ALM Help system.

When you click the **Help** link in Serena ALM, the Serena ALM online Help appears. You can navigate the contents, search, and view the glossary for information on installing, configuring, and using Serena Development Manager. To view and download readme files and PDFs for all supported releases, select **Demos & All Doc** in the online Help menu.

Getting Started with Serena Development Manager

In the default implementation of Serena Development Manager, you can choose from either Agile or Waterfall development methodologies or a mix of both. You can leverage the other solutions in the Serena ALM suite for an end-to-end application lifecycle management solution, including release management and requirements management. Custom implementations may include other SBM process applications (process apps) or solutions.

This documentation tells how to get started with the default implementation of standalone Development Manager.

Logging In to Serena ALM

To work with the Development Control, log in to Serena Development Manager using one of the default user accounts provided as part of your Development Manager Installation.

To log into Development Control:

1. In a supported browser application, open the URL to your Serena Business Mashups server. By default, this is:

   `http://server:`*`port`*`/tmtrack/tmtrack.dll?shell=alm`

   For example, if the server is called dev and the port number is 8085, and you are using the Serena-supplied custom shell:

   `http://dev:8085/tmtrack/tmtrack.dll?shell=alm`

   Don’t forget the question mark at the end of the `.dll`.

2. Enter your login credentials.

You will be presented with your home page that you can customize to your needs.

NOTES

- If your role is one of the Agile development roles, the appropriate Agile user interface elements will appear.
- If your role allows you to use products and components, the Products and Components user interface elements will appear.

Using Agile Development

This section consists of the following topics:

Agile Overview

The Agile functionality provided with Development Manager enables you to use Agile methodologies to manage your software development. This section describes the features available in a default implementation of Development manager. An overview of the terms and concepts involved is given below.

Roles

The roles that are specific to Agile are the Product Owner, the Scrum Master, and the Team Member.

NOTE The Team Member role is not defined as an SBM role in the default implementation of Development Manager. Any user may be a Team Member.

Product Owner

- Creates and manages the Product Backlog
- Defines the features of the product by creating epics and stories based on the customers’ needs
- Prioritizes features according to the business needs or market value
- Decides on release date and content and moves epics and stories from the product backlog to the Release Backlog
- Accepts or rejects work results

Scrum Master

- Adjusts features and priority of tasks for every sprint iteration, as needed
- Assigns tasks to Team Members
- Ensures that the team is fully functional and productive
- Removes impediments and blockers

Team Member

- Creates tasks for epics and stories
- Provides effort estimation for epics, stories, and tasks

- Updates progress and status
- Works with the team on impediments and blockers

Epics, Stories, and Tasks

Product features, requirements, or defects, are defined in terms of epics and stories. These epics and stories have tasks allocated to address them.

Story

A story describes a requirement or product feature in the form of "As a <user type> I want to <do some action> so that <desired result>". For example: "As a wiki user I want a tools menu on the edit screen so that I can easily apply font formatting".

Epic

An epic is a story that is has a large scope. Epics are typically divided into smaller stories, or child stories.

Task

A task is something that needs to be carried out to implement a user story. A user story can have multiple tasks associated with it. In the default implementation of Development manager, a task corresponds to a dev task and follows its workflow. If you have installed the Agile features of Development Manager, there will be some additional fields on the Dev Task forms. A task is estimated in hours.

Backlogs

Product Backlog

The product backlog contains the stories that need to be addressed for the product that have not yet been moved to the release backlog.

Release Backlog

The release backlog consists of stories that are scheduled for the next release but have not yet been moved to the sprint backlog for a specific sprint team.

Sprint Backlog

The sprint backlog is created by the scrum master in conjunction with the development team by selecting stories from the release backlog until the Development Team feels it has enough work to fill the sprint.

Sprint

A time period, for example 4 weeks, in which development occurs on a set of backlog items that the team has committed to.

Monitoring Progress

Taskboard

The taskboard shows the tasks for the current sprint, indicating their status such as whether a story is in elaboration, in progress, or complete, and whether it is blocked.

Blocker

A blocker is something that is preventing a story or task from being worked on, such as a dependency on another task or hardware/software problem. A blocker has some text describing the problem and is visible on the Scrum Master's Landing Page.

Burndown Charts

The sprint burn down chart is a chart showing remaining work in the sprint backlog in story points and task hours. Updated every day, it gives a simple view of the sprint progress.

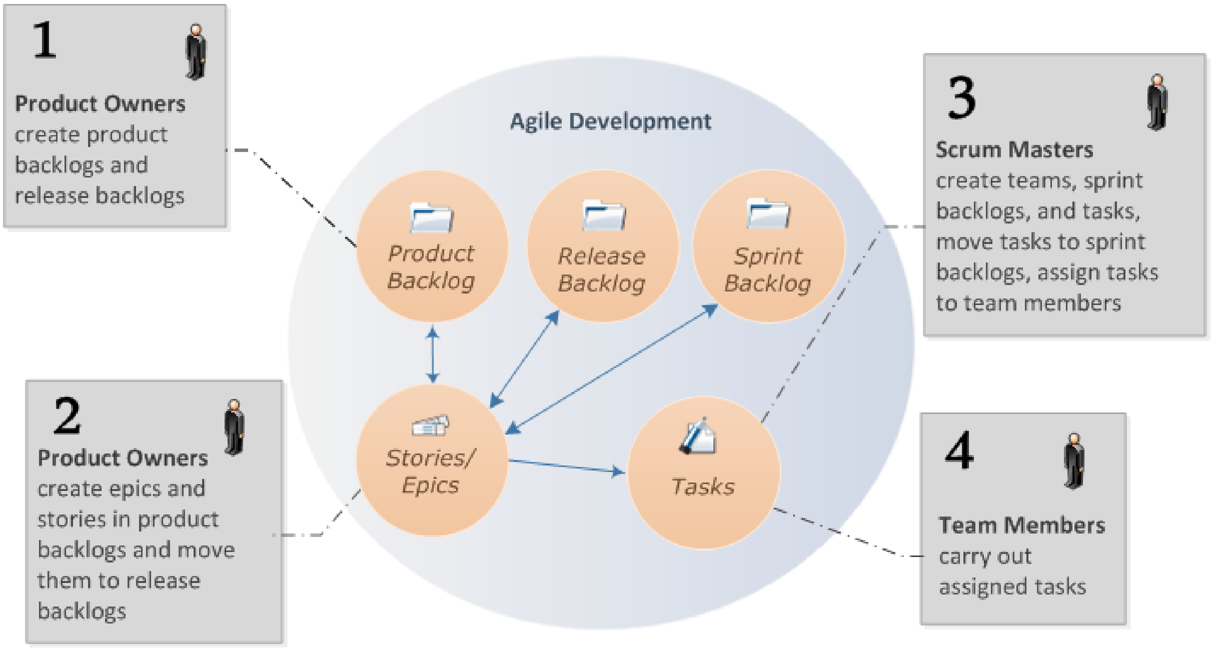
The Release Burndown shows remaining story points for the rest of the product release.

Velocity

The sprint velocity is the total effort a team is capable of in a sprint based on the number of story points completed from the last sprint's stories/features. This is a guideline for the team and assists them in understanding how many stories they can do in a sprint.

Agile Usage Overview

The usage of the default Agile implementation of Serena Development Manager is shown in the following diagram.

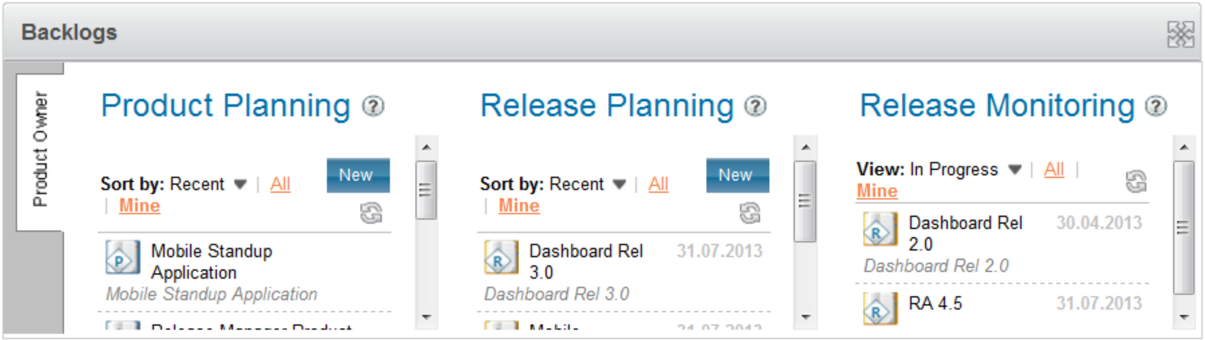

1
Product Owners create product backlogs and release backlogs
Agile Development
Product Backlog
Release Backlog
Sprint Backlog
3
Scrum Masters create teams, sprint backlogs, and tasks, move tasks to sprint backlogs, assign tasks to team members
2
Product Owners create epics and stories in product backlogs and move them to release backlogs
Stories/ Epics
Tasks
4
Team Members carry out assigned tasks

The following table elaborates on the steps in the diagram above and includes references to more detailed information.

Step	Description
1	Product owners create product backlogs and release backlogs. For details, see Using the Agile Landing Page [page 10] and The Backlogs Pane [page 11].
2	Product Owners create epics and stories in product backlogs and move them to release backlogs. For details, see Working With the Product Owners Backlogs Pane [page 12] and Managing a Backlog [page 13].
3	Scrum Masters create teams, sprint backlogs, and tasks, move tasks to sprint backlogs, assign tasks to team members. For details, see Working With the Scrum Masters Backlogs Pane [page 18] and Managing a Sprint [page 20].
4	Team Members carry out assigned tasks.

Using the Agile Landing Page

By default, if you have the Product Owner role, you will be presented on logging in with a Landing page similar to the following page.



This window consists of:

- A pane with your inbox and other Serena ALM applications you are using.
- A pane with a burndown chart. By default, it will display an active sprint if one is available.

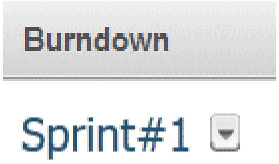
Using this page, you can:

- Display a burndown chart for a selected backlog or sprint.
- Manage backlogs using the Backlogs pane.

The Backlogs pane is different if you are a Scrum Master. For details, see The Backlogs Pane [page 11]

To choose which burndown chart is displayed:

Click the down arrow and select a sprint or release backlog.

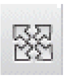


The burndown chart shows the estimated and remaining work in the selected backlog.

The lower part of this window contains the Backlogs pane.

To expand any pane to fill the whole window:

Click the button at its top right.



To collapse an expanded pane:

Click the button again.

The Backlogs Pane

The lower part of the Landing page for a Product Owner or Scrum master contains the Backlogs pane.

The Product Owner's Backlog pane contains:

- Product Planning: A list of product backlogs
- Release Planning: A list of release backlogs

- Release Monitoring: A list of release backlogs according to status

For details of managing backlogs as the Product Owner, see Working With the Product Owners Backlogs Pane [page 12].

If you are a Scrum Master, the upper panes of your default landing page will be similar, but the Backlogs pane will contain:

- Release Backlogs
- Sprint Backlogs
- Blockers

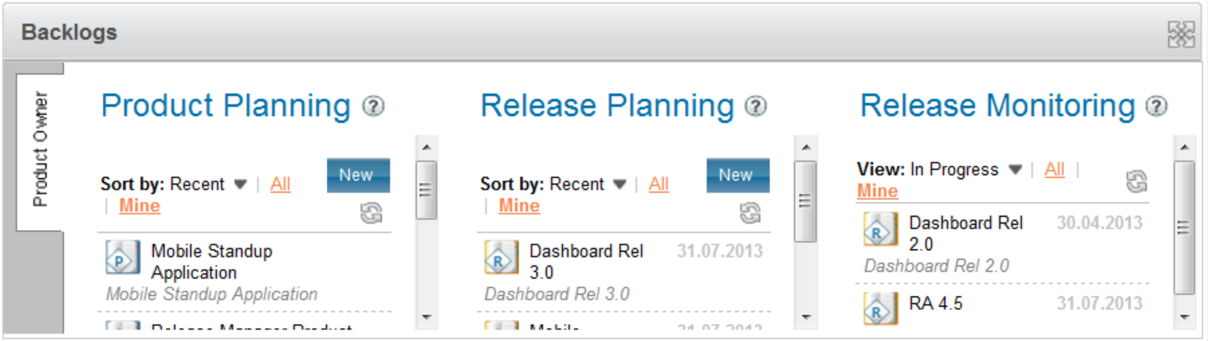
For details of managing backlogs as the Scrum Master, see Working With the Scrum Masters Backlogs Pane [page 18].

Working With the Product Owner's Backlogs Pane

The Backlogs pane allows you to manage product and release backlogs.

From this pane, the Product Owner can:

- List backlogs for Product Planning and Release Planning
- Add a new product or release backlog
- Select a product or release backlog to view or edit its details



Sorting Product or Release Planning Backlogs

To sort Product or Release Planning backlogs:

From the **Sort by:** drop-down menu, select:

- **Recent** to sort the backlogs in descending order of the last time you accessed them
- **Name** to sort in the backlogs in alphabetical order

Filtering Release Monitoring Backlogs

To filter Release Monitoring backlogs:

From the **View:** drop-down menu, select:

- **In Progress**
- **Completed**

Filtering the List of Backlogs

To filter the list of Product or Release Planning backlogs or Release Monitoring backlogs:

Click the link:

- **All** to view all backlogs

- **Mine** to view backlogs created by you

Creating Product Backlogs

To create a product backlog:

1. Under Product Planning, click the **New** Button.
2. On the New Backlog page, enter the **Product Name** and optionally, a **Description**.
3. Click **OK**.

Creating Release Backlogs

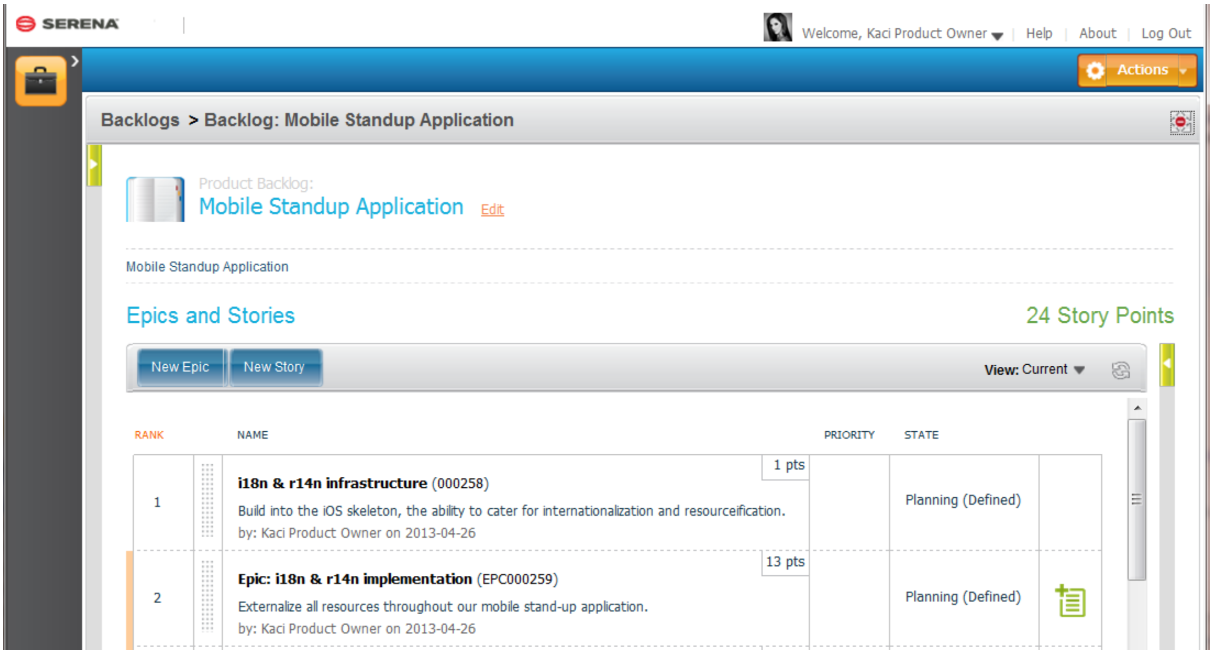
To create a release backlog:

1. Under Release Planning, click the **New** Button.
2. On the New Backlog page, enter the **Release Name** and select the **State**.
3. Optionally, enter a **Description**, a **Start Date**, and an **End Date**.
4. Click **OK**.

Viewing Backlogs

To view the page for a backlog:

Click the row for the backlog. This will display the backlog in the Backlogs pane.



Mobile Standup Application

RANK	NAME	PRIORITY	STATE
1	1 pts **i18n & r14n infrastructure** (000258) Build into the iOS skeleton, the ability to cater for internationalization and resourceification. by: Kaci Product Owner on 2013-04-26		Planning (Defined)
2	13 pts **Epic: i18n & r14n implementation** (EPC000259) Externalize all resources throughout our mobile stand-up application. by: Kaci Product Owner on 2013-04-26		Planning (Defined)

You can return to the backlogs by clicking **Backlogs** in the breadcrumbs.

Editing Backlogs

To edit a backlog:

Click the row for the backlog, then click the **Edit** link. for details, see Managing a Backlog [page 13]

Managing a Backlog

The Backlog page enables you to manage a selected product or release backlog.

This is the page Product Owners and Scrum Masters typically work from.

> **TIP** Example Scenario Users: Log on as Kaci to try this out as the Product Owner. Log on as Sam Scrum Master to try this out as the Scrum Master.

Use the backlog page to:

- Edit backlog details
- Add or edit stories
- Add or edit epics
- Break epics into child stories
- Move epics and stories into the backlog
- Change the ranks of epics and stories
- Display burndown charts for release backlogs

Editing a Product Backlog

To edit a product backlog:

Click the **Edit** link. You can change:

- Product Name
- Description

Editing a Release Backlog

To edit a release backlog:

Click the **Edit** link. You can change:

- Release Name
- Description
- Start Date
- End Date
- State: Planning, In Progress, or Completed

Creating a Story

To create a story:

1. Click the **New Story** button

   The **New Story** page appears.

2. Fill out the form.

   For example, you would select or enter values in the default form for the following:

   - Story Points
   - Name
   - Type
   - Target Due Date
   - Associate to Project
   - Customer Found

- Estimated Effort
- Description
- Acceptance Criteria
- Comments
- Roles
  - Development Manager
  - Developer
  - QA Engineer

3. Click **OK**.

Viewing or Editing a Story

To view or edit a story:

1. Click its name link.

   The Change Request page is displayed to enable you to edit its details.

Creating an Epic

You create epics from the backlog list. While working with epics, you continue working in the backlog list.

To create an epic:

1. Click the **New Epic** button

   The **New Epic** page appears.

2. Fill out the form.

   For example, you would select or enter values in the default form for the following:

   - Story Points
   - Name
   - Target Due Date
   - Associate to Project
   - Customer Found
   - Estimated Effort
   - Description
   - Acceptance Criteria
   - Comments
   - Roles
     - Development Manager
     - Developer
     - QA Engineer

3. Click **OK**.

Creating Child Stories for Epics

Start breaking down the work by creating child stories for an epic.

The child stories appear in the backlog list as well as in the Epic Overview pane; the Epic Overview pane shows the hierarchy, whereas the backlog shows the individual stories, whether they are a child of an epic or a standalone story.

Creating the First Child Story

To create the first child story for an epic:

1. Click the button at the right of the epic; the hover text (tooltip) gives contextual instructions.
2. Create the story using the procedure in Creating a Story [page 14].

   Once you create one story for an epic, the epic disappears out of the backlog and is replaced by the child story. You can still see the epic itself, as the epic still appears in the Epic Overview pane, and the hierarchy of child stories is shown there.

   The child story in the backlog gives both the story item ID and the epic ID.

Creating Additional Child Stories

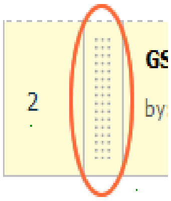
To create additional child stories for an epic:

1. Click the button at the right of the child story; the hover text (tooltip) gives contextual instructions.
2. Create the story using the procedure in in Creating a Story [page 14].

Changing the Rank of a Story

To change the rank of a story:

Drag the area next to the rank number and move it up or down to the required position.



Showing a Burndown Chart

To show the burndown chart for a release backlog:

Click the **Show Burndown** link at the top right. This will display a burndown chart in the backlog page.

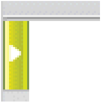
The Stories Panel

Use the Stories panel to move stories into any product, release or sprint backlog to which you have permissions to edit that is currently selected in the Backlog page. You can move stories by selecting them in the Stories panel and dragging them into the backlog.

Showing or Hiding the Stories Panel

To show or hide the Stories panel:

To show the Stories panel, click the arrow to the left of the Backlog pane:



To hide it, click the arrow again.

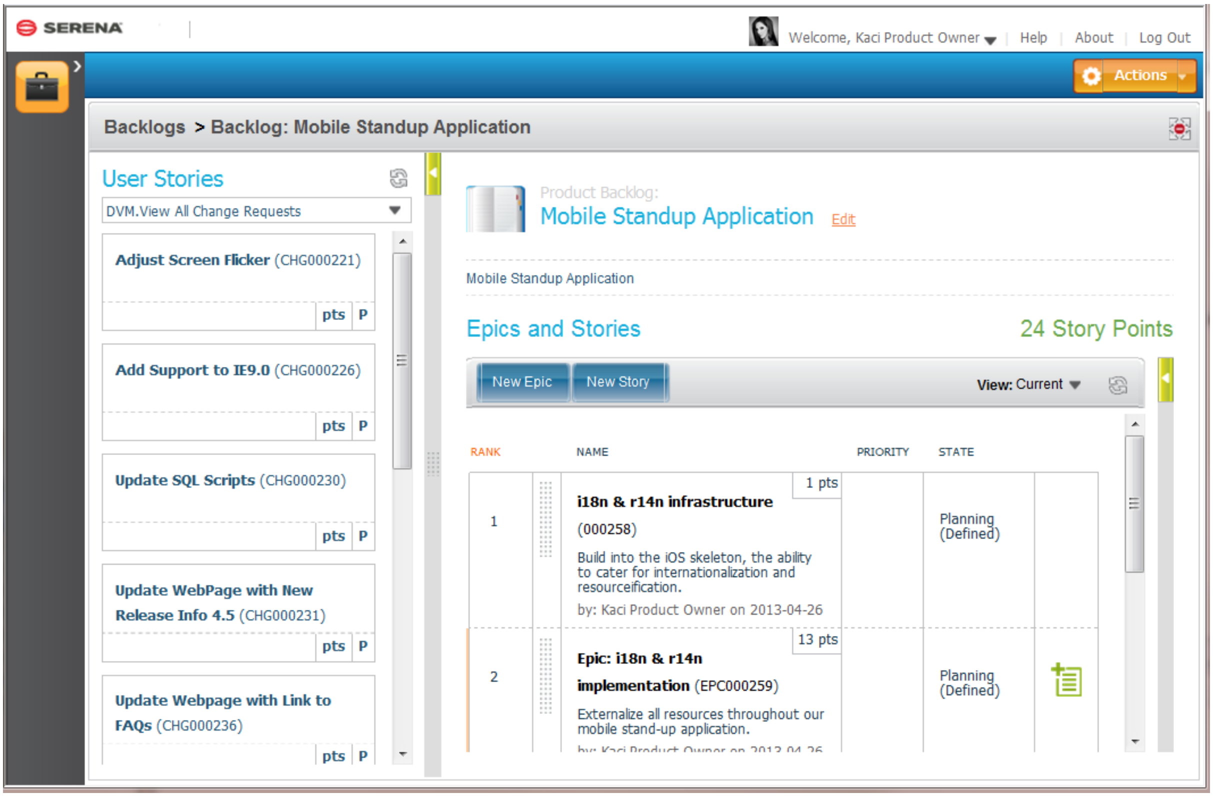
Setting the Stories Panel to Display Stories and Epics

To display stories and epics in the Stories panel:

In the User Stories drop-down list, select a report from one of the following:

- Epics that are not in backlogs
- Stories that are not in backlogs
- Product backlogs
- Release backlogs
- Sprint backlogs

After you select the report, the list of stories for that category appear in the Story panel.


SERENA
Welcome, Kaci Product Owner
Help
About
Log Out
Actions
Backlogs > Backlog: Mobile Standup Application
User Stories
DVM.View All Change Requests
Adjust Screen Flicker (CHG000221)
pts
P
Add Support to IE9.0 (CHG000226)
pts
P
Update SQL Scripts (CHG000230)
pts
P
Update WebPage with New Release Info 4.5 (CHG000231)
pts
P
Update Webpage with Link to FAQs (CHG000236)
pts
P
Product Backlog:
Mobile Standup Application
Edit
Mobile Standup Application
Epics and Stories
24 Story Points
New Epic
New Story
View: Current
RANK
NAME
PRIORITY
STATE

RANK	NAME	PRIORITY	STATE
1	1 pts **i18n & r14n infrastructure** (000258) Build into the iOS skeleton, the ability to cater for internationalization and resourceification. by: Kaci Product Owner on 2013-04-26		Planning (Defined)
2	13 pts **Epic: i18n & r14n implementation** (EPC000259) Externalize all resources throughout our mobile stand-up application.		Planning (Defined)

Moving Epics and Stories to the Backlog

To move an epic or story from the Stories panel into the selected backlog:

Select one or more stories and drag them to the required position in the backlog.

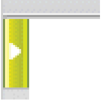
Note that you cannot drag stories back into the Stories panel.

The Epic Overview Pane

Use the Epic Overview pane to see the list of Epics and the child stories associated with each epic.

To show or hide the Epic Overview pane:

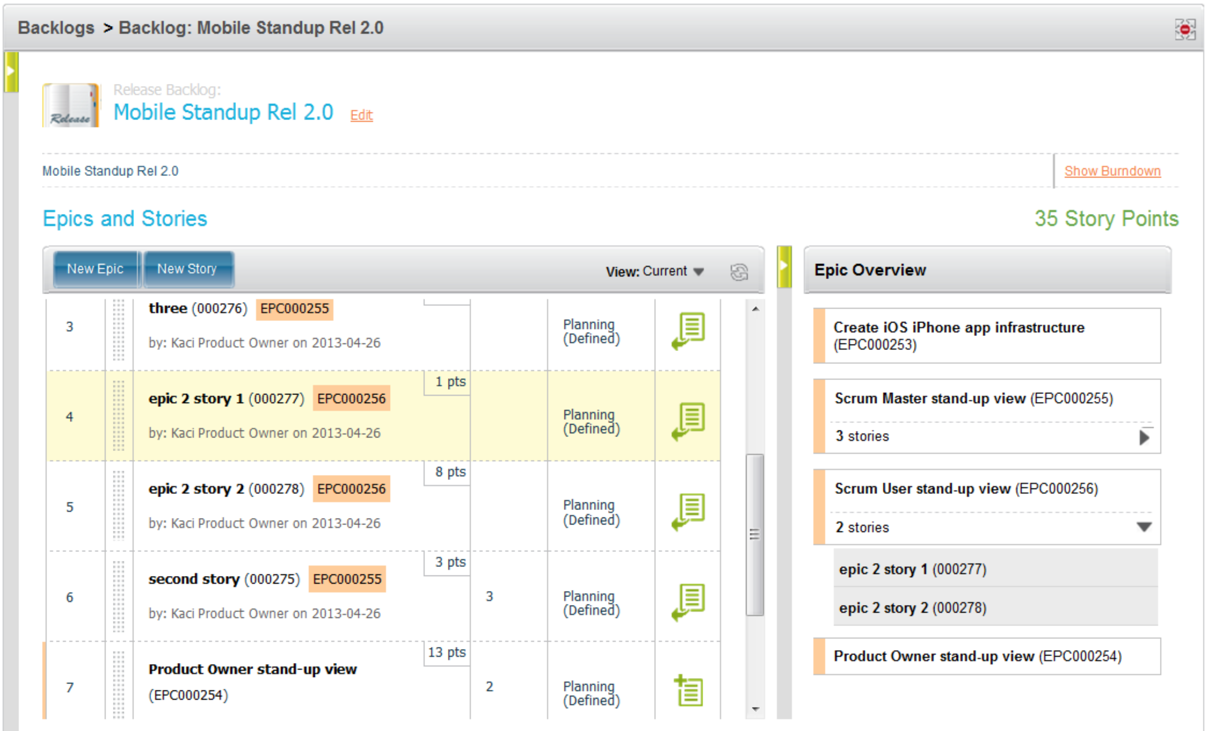
To show the Epic Overview pane, click the arrow to the right of the Backlog pane:



To hide it, click the arrow again.

Showing the Child Stories for an Epic

From the Epic Overview pane, expand an epic to see its child stories.



3	**three** (000276) EPC000255 by: Kaci Product Owner on 2013-04-26			Planning (Defined)
4	**epic 2 story 1** (000277) EPC000256 by: Kaci Product Owner on 2013-04-26	1 pts		Planning (Defined)
5	**epic 2 story 2** (000278) EPC000256 by: Kaci Product Owner on 2013-04-26	8 pts		Planning (Defined)
6	**second story** (000275) EPC000255 by: Kaci Product Owner on 2013-04-26	3 pts	3	Planning (Defined)
7	**Product Owner stand-up view** (EPC000254)	13 pts	2	Planning (Defined)

Create iOS iPhone app infrastructure (EPC000253)

Scrum Master stand-up view (EPC000255)
3 stories

Scrum User stand-up view (EPC000256)
2 stories
- **epic 2 story 1** (000277)
- **epic 2 story 2** (000278)

Product Owner stand-up view (EPC000254)

Working With Epics

When you scroll and select an epic in the Epic Overview pane, the backlog list scrolls automatically to position you at the epic or child story for that epic.

You can view the epic or story content from the backlog list or from the Epic Overview pane. See Managing a Backlog [page 13] for information on working with epics.

Moving Stories to Different Epics

To move a story to a different parent:

1. Expand an epic to see its child stories.
2. Drag a story to another epic within the Epic Overview pane.

Working With the Scrum Master's Backlogs Pane

The Scrum Master's Backlogs pane allows you to manage release and sprint backlogs.

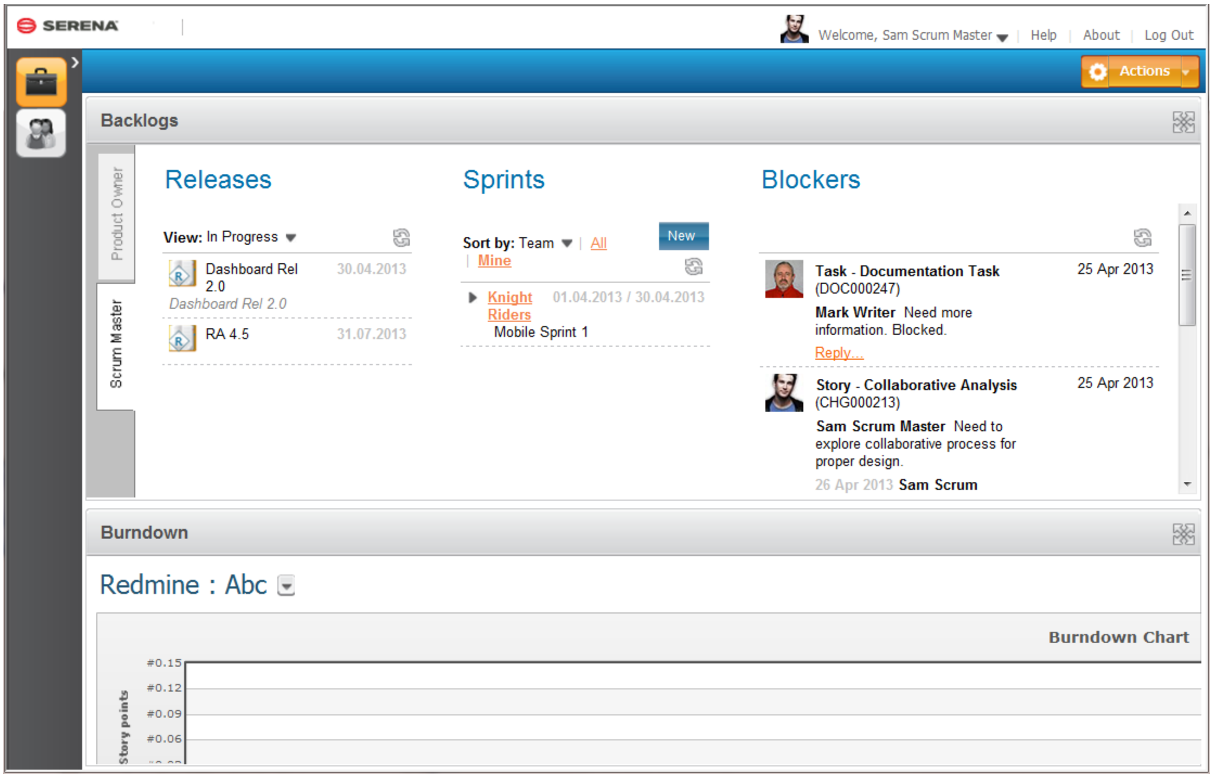
From this pane, the Scrum Master can:

- List backlogs for Releases and Sprints
- Add a new sprint
- Select a release or sprint backlog to view or edit its details
- View any current blockers for stories or tasks

If you are a Scrum Master, your default landing page is similar to that of the Product Owner, except that it shows:

- Release Backlogs for which you are the Scrum Master
- Sprint Backlogs
- Blockers for backlogs for which you are the Scrum Master

As Scrum Master, you cannot create or edit a release backlog but you can create and edit its stories and move stories to it.



Dashboard Rel 2.0 30.04.2013
Dashboard Rel 2.0
RA 4.5 31.07.2013

Knight Riders 01.04.2013 / 30.04.2013
Mobile Sprint 1

Task - Documentation Task (DOC000247) 25 Apr 2013
Mark Writer Need more information. Blocked.
Reply...

Story - Collaborative Analysis (CHG000213) 25 Apr 2013
Sam Scrum Master Need to explore collaborative process for proper design.
26 Apr 2013 Sam Scrum

Redmine : Abc

Sorting Releases

To sort the release backlogs:

Click the arrow in the heading and select:

- **Planning**
- **In Progress**
- **Completed**

Filtering Sprints

To filter the sprint backlogs:

Select:

- **All** to view all backlogs
- **Mine** to view backlogs created by you

Sorting Sprints

To sort the sprint backlogs:

Click the arrow in the heading and select:

- **Recent** to sort the sprints in descending order of the last time you accessed them
- **Name** to sort in the sprints in alphabetical order
- **Team** to sort in order of team

Expanding Sprints for Teams

To expand the list of sprints for a team:

Click the arrow ▶ next to the team name.

To collapse it, click the arrow again.

Creating Sprints

To create a sprint:

1. Under Sprints, click the **New** Button.
2. On the New Backlog page, enter the **Sprint Name** and select a **Team**.
3. Optionally, enter a **Description**, a **Start Date**, and an **End Date**.
4. Click **OK**.

Editing Team Details

To edit the details of a team:

1. Click the name link for the team in the Sprints list.
2. Enter the **Team Name** and select a **Scrum Master**.
3. Optionally, enter the **Velocity** in story points and/or hours, and enter a **Description**.
4. Click **OK**.

Replying to Blocked Stories

To reply to a blocked story with a comment:

Click the blocked story's reply link and enter some text. Your comment is added to the blocker details.

Viewing Blocked Stories

To view a blocked story in a new window:

Click the blocked story's name link.

Viewing Sprint Pages

To View the page for a sprint:

Click the row for the sprint. This will display the sprint items in the Backlogs pane. For details, see Managing a Sprint. [page 20]

You can return to the backlogs by clicking **Backlogs** in the breadcrumb link in the header of the Backlogs pane.

Managing a Sprint

The backlog page for a sprint shows stories and their associated tasks.

On this page you can:

- Edit the details of a sprint
- Create a new sprint

- Add or edit an epic or story
- Add or edit a task
- Change the rank of an epic or story
- Move epics and stories into the backlog
- Display the burndown chart for a sprint

You can choose between two views of the sprint items:

- **Backlog**: a view of the backlog of stories
- **Taskboard**: a view of the tasks listed by status.

Editing Sprint Details

To edit the details of the sprint:

Click the **Edit** link. You can change:

- Sprint Name
- Description
- Start Date
- End Date

Viewing the Sprint Burndown Chart

To show the burndown chart for this sprint:

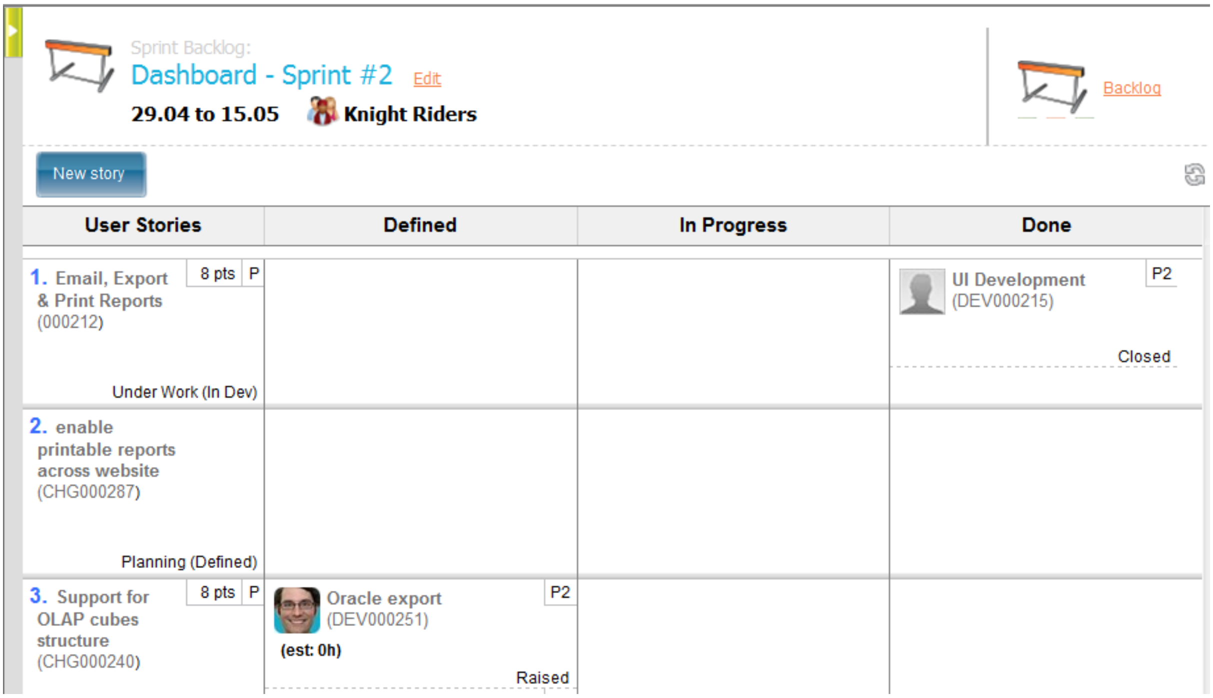
Click the **Show Burndown** link at the top right.

Viewing the Sprint Taskboard

To view the taskboard:

Click the **Taskboard** link in the heading or the icon to the right of the Sprint drop-down, depending on your current page.

This will show tasks listed in columns according to their status.



Sprint Backlog:

Dashboard - Sprint #2 Edit

29.04 to 15.05 Knight Riders

Backlog

New story

User Stories	Defined	In Progress	Done
1. Email, Export & Print Reports (000212) 8 pts P Under Work (In Dev)			UI Development (DEV000215) P2 Closed
2. enable printable reports across website (CHG000287) Planning (Defined)			
3. Support for OLAP cubes structure (CHG000240) 8 pts P	Oracle export (DEV000251) P2 (est: 0h) Raised		

Switching to the Backlog View

To switch to the backlog view:

On the taskboard, click the **Backlog** link in the heading.

Creating Sprints

To create a sprint:

Click the **New sprint** button.

Creating Epics or Stories

To create an epic or story:

Click the **New epic** or **New story** button.

Editing a Story

To edit a story:

Click the name link for the story.

Creating a Task

To create a new task:

1. Click the name link for the story.
2. Select the **Tasks** tab.
3. Click the **New Task** button.

   This page is similar to the Create Dev Task page described in Creating New Tasks [page 36], except that there is a header containing a **Priority** field.
4. On the Create Dev Task page, select a **Priority**. If you want to show the task as blocked, click the Block button and enter some text for the description.

Editing a Task

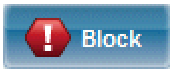
To edit a task:

Click the name link.

Blocking a Story or Task

To block a story or a task:

1. On the Change Request page or the Dev Task page, click the **Block** button.

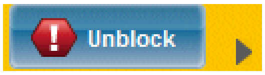
   

2. Enter some text for the description.
3. Click **OK**.

Unblocking a Story or Task

To unblock a story or a task:

Click the **Unblock** button.



Viewing Details of a Blocked Task

To view the details of a blocked story:

Click the arrow to the right of the **Unblock** button. To add a comment, click the **Reply** link and enter some text.

Working with My Tasks

The My Tasks page is available in the My Tasks tab for Team Member to view their assigned tasks.

On this page you can:

- View the list of tasks assigned to you within a selected sprint
- View or edit tasks
- Switch to the Social view to communicate with the team

Switching to the Social View

To switch to the social view:

On the My Tasks page, click the **Social** button in the heading.

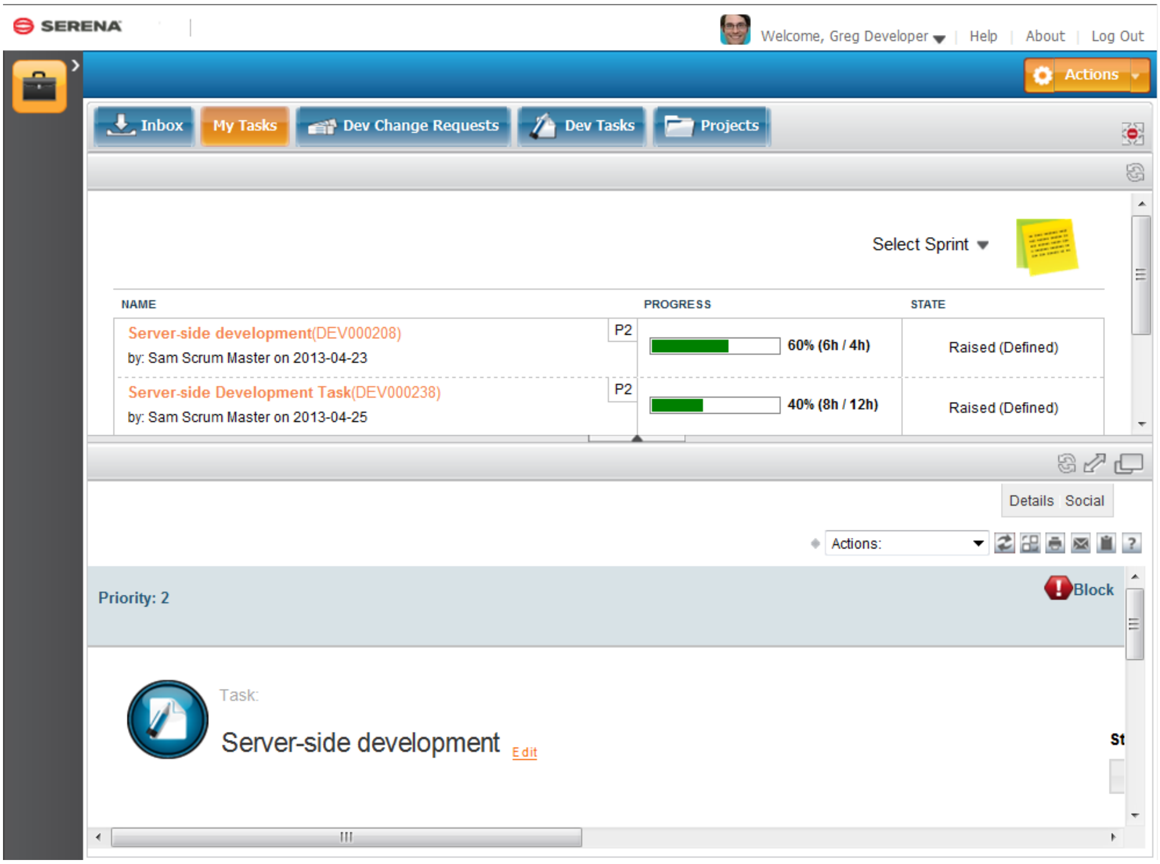
On the social page, you can communicate about the task with other team members.

Viewing the My Tasks Page

To view the My Tasks page:

1. Log on as a Team Member. For example, Greg Developer.
2. Select the **My Tasks** tab.

   The My Tasks page similar to the following appears.



NAME		PROGRESS	STATE
Server-side development(DEV000208) by: Sam Scrum Master on 2013-04-23	P2	60% (6h / 4h)	Raised (Defined)
Server-side Development Task(DEV000238) by: Sam Scrum Master on 2013-04-25	P2	40% (8h / 12h)	Raised (Defined)

Priority: 2

Task:
Server-side development

Viewing a Task

To view a task:

- Select the sprint for the tasks you want to view.
- Click the task if in anchored mode, or double-click if in un-anchored mode.

  The task appears in either the preview pane or a separate window.

Editing a Task

To edit a task:

View the task and then click the Edit link.

You can edit the fields on the task form, change priority, and add comments.

Working on a Task

To work on a task:

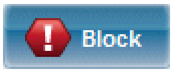
Click the **Begin Work** button to work on the item yourself, or click **Delegate** to delegate the work to someone else.

The buttons appear based on the My Task workflow. The workflow may be different for your implementation.

Blocking a Task

To block a task:

1. On the My Task page, click the **Block** button.

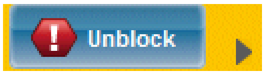


2. Enter some text for the description.
3. Click **OK**.

Unblocking a Task

To unblock a task:

Click the **Unblock** button.



Viewing Details of a Blocked Task

To view the details of a blocked task:

Click the arrow to the right of the **Unblock** button. To add a comment, click the **Reply** link and enter some text.

Development Lifecycle

To get started using Development Manager for the full development lifecycle, see the following sections.

Development Lifecycle Overview

Serena Development Manager enables you to:

- Orchestrate and monitor your key software development efforts, track source code changes, and manage approvals through a central workflow engine.
- Enforce your organizational policies and processes and make them visible to all stakeholders.
- Create organizational expectations for establishing and maintaining baselines, tracking and controlling changes to work products under configuration management, and establishing and maintaining integrity of those baselines.
- Coordinate events across your systems, integrating application project definition, source code management, test management, and release approvals.

Serena Development Manager includes the following components. You can configure and customize the underlying workflows and components based on your organization's needs.

- **Development project lifecycle management.** Promote development activities through lifecycles stages such as planning, code, test, and release. You can customize the development stages according to your organizational practices, define user roles and owners for each stage, and set up notifications.
- **Source configuration management**. Development Manager default provides the capability of using two different Configuration Management tools to manage your source code:
  - **Dimensions CM**, which enables you to control your source code for development tasks and baselines for development packages.

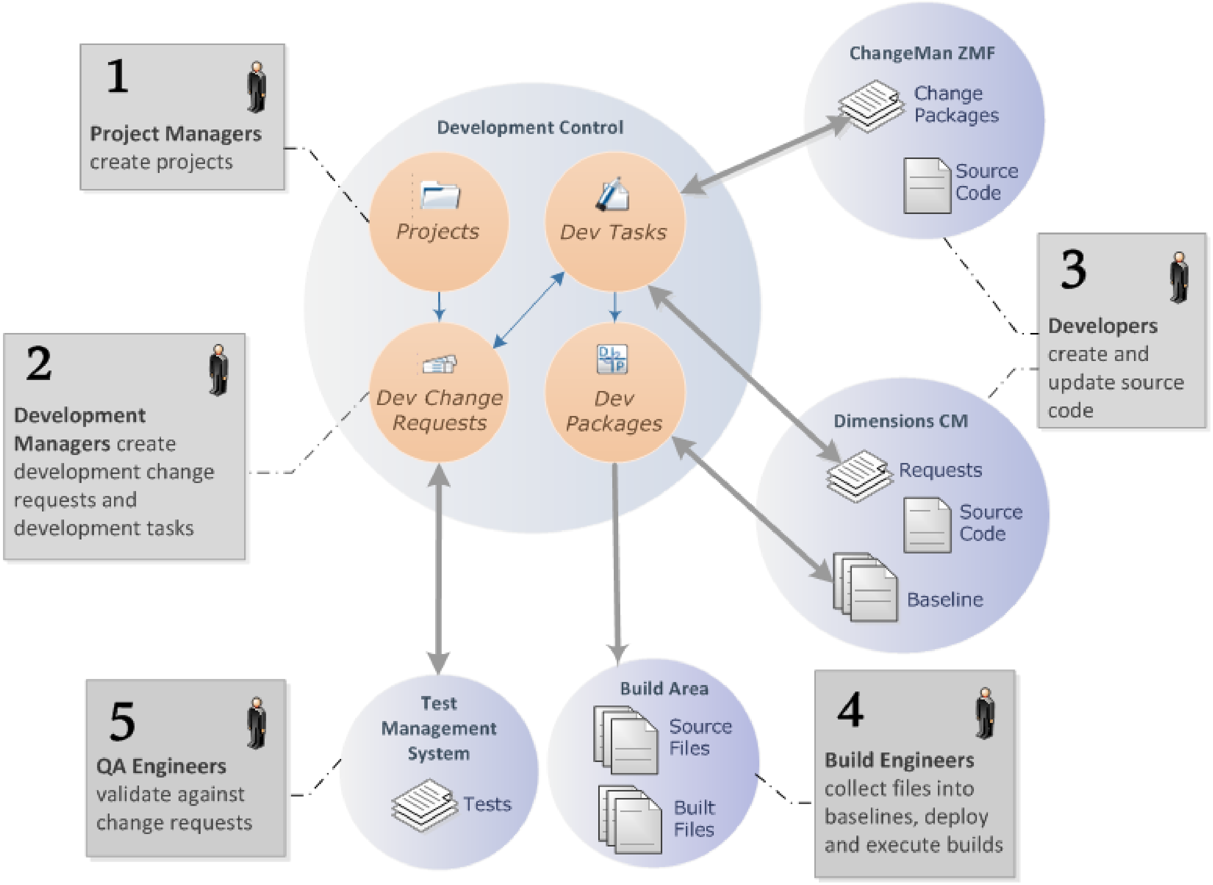
  - **ChangeMan ZMF** which enables you to control your source code for development tasks (but not development packages).

- **Test management system integration**. Monitor your QA team's progress on defects and requirements by integrating Development Manager with HP Quality Center / ALM.

- **Graphical business intelligence dashboard**. As an option, Serena provides a rich, graphical reporting tool, Serena® Dashboard, that allows you to carefully monitor key performance indicators (KPIs). The reports are fully configurable, and can provide graphical information on any aspects of your projects that managers or executives need to track.

Usage Overview

The development lifecycle workflows of 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*.



The following table elaborates on the steps in the diagram above and includes references to more detailed information.

Step	Description

1	At the beginning of a project, a project manager uses the Projects page in Development Control 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 [page 54] . For details on creating projects, please see Creating and Managing Projects [page 33]. If you are using the additional separately licensed Serena Requirements Manager, the requirements that drive the development process will be managed, approved and assigned using this application. For details see Using the Requirements Manager Integration.
2	The Development Manager creates development change requests using the Dev Change Requests page - or from the project page. 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 page. 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 [page 55] and Dev Tasks Workflow [page 56]. For information on creating change requests, please see Creating Dev Change Requests for the Requirements in an ALM Project. For information on working with tasks, please see Creating and Working on Tasks [page 36].
3	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.
4	As work progresses on the project, the build engineer sets up Dev Packages 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 Package Workflow [page 59]. For details on creating and working with development packages, see Creating and Working with Development Packages [page 40].
5	Using HP Quality Center / ALM, the QA staff tests the builds, both the nightly builds and release candidate builds. Defects may be tracked in Dev Change Requests, 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 Example Use Case

We will walk through a scenario that demonstrates what Development Manager can do for you. Keep in mind that this is just an example, and that these steps can all be customized to satisfy the goals of your organization. Customization is one of the key benefits of Serena solutions built on Serena Business Manager; you can adapt the workflows, forms, integrations, and reports to meet your business needs.

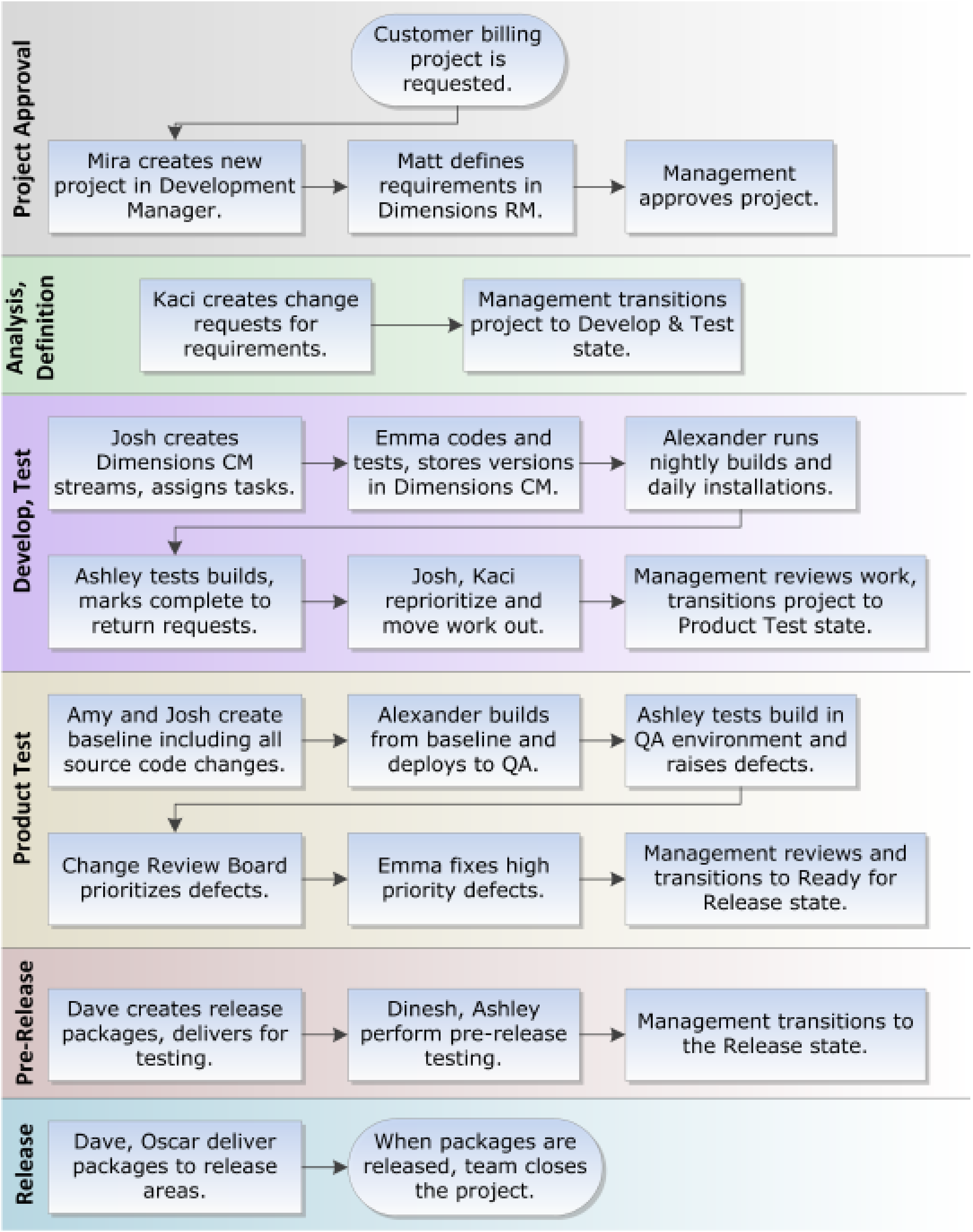
User Roles

Serena Development Manager is designed to meet the needs of key participants in a standard development effort. Roles in your organization may vary. This example scenario includes the following users:

- Mira: Project Manager
- Matt: Business Analyst
- Kaci: Product Owner
- Sam: Scrum Master
- Eva: Component Manager
- Josh: Development Manager
- Emma: Development Engineer
- Amy: Build Manager
- Alexander: Build Engineer
- Dinesh: QA Manager
- Ashley: QA Engineer
- Dave: Release Manager
- Oscar: Release Engineer

Process Example

The following flowchart presents an example scenario using the personas described above. Details about each step follow the diagram.


Project Approval
Customer billing project is requested.
Mira creates new project in Development Manager.
Matt defines requirements in Dimensions RM.
Management approves project.
Analysis, Definition
Kaci creates change requests for requirements.
Management transitions project to Develop & Test state.
Develop, Test
Josh creates Dimensions CM streams, assigns tasks.
Emma codes and tests, stores versions in Dimensions CM.
Alexander runs nightly builds and daily installations.
Ashley tests builds, marks complete to return requests.
Josh, Kaci reprioritize and move work out.
Management reviews work, transitions project to Product Test state.
Product Test
Amy and Josh create baseline including all source code changes.
Alexander builds from baseline and deploys to QA.
Ashley tests build in QA environment and raises defects.
Change Review Board prioritizes defects.
Emma fixes high priority defects.
Management reviews and transitions to Ready for Release state.
Pre-Release
Dave creates release packages, delivers for testing.
Dinesh, Ashley perform pre-release testing.
Management transitions to the Release state.
Release
Dave, Oscar deliver packages to release areas.
When packages are released, team closes the project.

Project Phase	Description	ALM Page or Tool

Project Approval	Matt (*Business Analyst*) asks Mira (*Project Manager*) to create a new project for the next release of the customer billing system. Mira creates the project on the Project page of Development Manager. This project will be used to coordinate development activities.	Projects
	Matt is notified that the project has been created, and uses his requirements management tool to define detailed requirements. He adds requirement links to the project.	External requirements management tool
	The management team (Mira, Matt, Kaci, Josh, Dinesh, Amy, and Dave) reviews the requirements and works through any open questions. When the requirements are approved, they promote the project to the next phase, *Analysis and Definition*.	Projects
Analysis and Definition	Kaci (*Product Owner*) reviews the detailed requirements and creates software change requests for each of them. Her goal is to break the requirements into small, clearly defined units of work.	Projects
	The management team reviews the requirements, stories, and testing requirements. When the requirements and change requests are approved, they promote the project to the next state, *Develop and Test*.	Projects

Develop and Test	Josh (*Development Manager*) reviews the requests and works with Kaci to prioritize the work. Josh also creates the development streams in Dimensions CM where the source code will be managed.	Projects
	Josh, Kaci, Emma (*Software Engineer*), and Ashley (*QA Engineer*) break the requests into development tasks. As they are created, the tasks are synchronized to Dimensions CM requests. Once the requests and tasks are ready to work on, Josh moves them into development. Ashley and Emma can now begin work.	Projects
	Emma develops and tests code changes for each of her tasks. She works with the tasks in Dimensions CM via the Dimensions CM integration to Microsoft Visual Studio. She stores new versions of source files in Dimensions CM. Meanwhile, Ashley defines test cases that can be used to test each story as it is completed.	Dimensions CM/ ChangeMan ZMF
	Alexander (Build Engineer) manages nightly builds and daily installation on the QA test environments.	External build tool
	When she completes all of the tasks for a change request, Emma assigns that request to Ashley for testing.	Dimensions CM/ ChangeMan ZMF
	Ashley runs the test cases for the request that Emma finished coding. Ashley uses the installation that Alexander runs from the nightly build. Ashley returns any request that fails testing back to Emma to fix.	HP Quality Center / ALM
	Once she has finished testing a request, Ashley marks it as complete. This delegates the request back to Josh.	HP Quality Center / ALM
	Josh and Kaci continually review the list of requests in progress, and move work out of the project as needed based on remaining time and business priorities.	Projects
	Once the requests have all been developed and tested, the management team reviews the requirements, requests, and test results. When they approve the completed work, they move the project to the next phase, *Product Test*.	Projects

Product Test	Amy and Josh create a Dev Package, and then relate that package to a baseline in Dimensions CM. The baseline assembles all of the source code changes associated with completed requests. Amy delegates the package to Alexander who then builds the source associated with the Dimensions CM baseline. He collects the build outputs back into Dimensions CM, and deploys them to the QA testing environments.	ALM Dev Packages, Dimensions CM
	Josh assigns the turnover package to Dinesh (*QA Manager*) to coordinate testing.	Projects
	Dinesh prioritizes testing requirements for the project.	HP Quality Center / ALM
	Ashley runs test cases against the QA environments and raises defects as she finds issues.	HP Quality Center / ALM
	The change review board determines which defects need to be fixed. Defects that need to be fixed are delegated to Emma. Deferred defects are moved out of the project to be fixed at a later date.	Projects
	As Emma fixes defects, new packages that include the fixes are forwarded to QA. Ashley re-tests the defects.	HP Quality Center / ALM
	Once all of the requirements and defects have been tested, the management team reviews the requirements, stories, package(s), and test results. Once they approve the completed work, they promote the project to the next phase, *Ready for Release*.	Projects
Ready for Release	Using Serena Release Manager, Dave (*Release Manager*) integrates the packages into the release cycle. He works with Oscar (Release Engineer) to get the package delivered to the appropriate pre-release areas for testing.	Dev Packages
	Dinesh coordinates the pre-release testing and Ashley raises defects as necessary.	HP Quality Center / ALM
	Once the packages have been successfully tested, the management team does a final review of the requirements, completed requests, packages, and test results. Once they approve the release, they promote the project to the next phase, *Release*.	Projects
Released (inactive)	Using Serena Release Manager, Dave works with Oscar to deliver the package to the release team.	Release Control
	Once the packages have been successfully released and the management team approves the final release, they close the project.	Projects

Creating and Managing Projects

Use the **Projects** tab to view and manage your development projects. As new requests and requirements come in, create projects to manage development work. Once you create a project, you can associate change requests and packages with it.

Creating a Project

From the **Actions** list at the top right of the window, select *New Project*.

Note that you will need to be logged in as a user that has the privileges to create a project, such as Mira, the Project Manager.

A **Create Project** form appears.

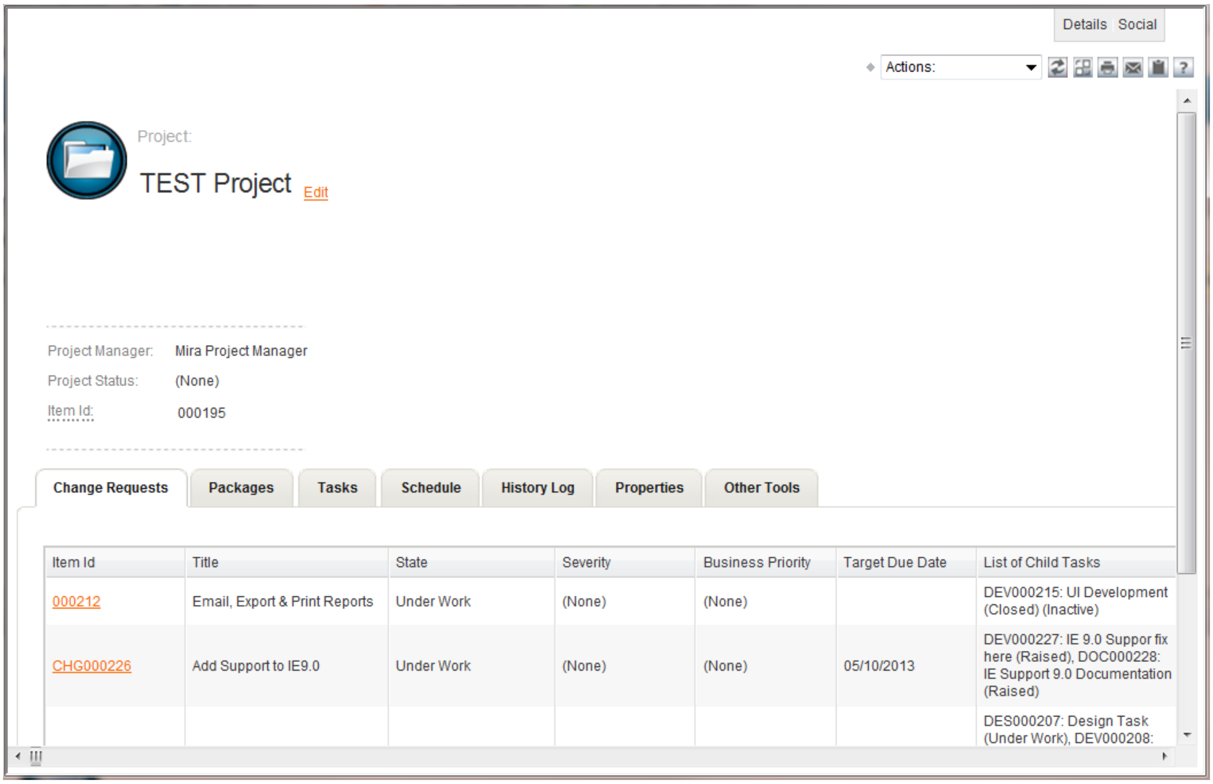
When you create a new project, you set a number of core properties for the project, including:

- The type of project, from the **Project Type** list. In the default implementation, you can choose **Innovation** or **Operational**.
- The title and description.
- Key users involved with the project, such as the Project Manager, Product Owner, Business Analyst, and Development Manager.
- Milestone dates for the project. This includes target start and end dates for each stage in the project, including Inception, Elaboration, Construction, and Transition. You can set actual start and end dates later on as the actual dates are passed
- Configuration Management Settings. You can select ChangeMan ZMF, Dimensions CM, or both as a Configuration Management provider in which to carry out your development work.
- Components: You can relate the project to an affected product and components, and used components. This may be done as an edit, post creation.

NOTE To fill in the "Component" or "Configuration Management Settings" sections of the form for Dimensions CM projects, you will need to be a registered user in Dimensions CM (so it may be the Development Manager who fills in these fields later).

Example Project

Here we see an example project. Click the **Change Requests** tab to view the list of change requests associated with it.



Project:

TEST Project

Project Manager: Mira Project Manager

Project Status: (None)

Item Id: 000195

Item Id	Title	State	Severity	Business Priority	Target Due Date	List of Child Tasks
000212	Email, Export & Print Reports	Under Work	(None)	(None)		DEV000215: UI Development (Closed) (Inactive)
CHG000226	Add Support to IE9.0	Under Work	(None)	(None)	05/10/2013	DEV000227: IE 9.0 Suppor fix here (Raised), DOC000228: IE Support 9.0 Documentation (Raised)
						DES000207: Design Task (Under Work), DEV000208:

- Click the **Tasks** tab to view the list of tasks.
- You can transition a project to the next state in the lifecycle by pressing the appropriate transition button, such as **Begin Elaboration** or **Begin Construction**. Note that you need the

required role, for example Project Manager, to transition the project, otherwise the required button will not be displayed.

- You can update details for the project by clicking the **Edit** link.
- You can display transition history for a project or other Development Manager item by selecting its **History Log** tab.
  - Below the state change history you can view the history of activities by expanding the **Activity Log**.
  - You can expand an entry to reveal more details by clicking the + icon.

Displaying and Creating Related Change Requests

On the **Change Requests** tab, you can list and display details on the change requests associated with the project, as well as create new change requests.

To create new change requests, click the **Create Dev CR** button.

You can also create dev change requests from the **Action** button, but in this case you will need to select the ALM project to which it belongs when you create it.

See Creating New Change Requests [page 35] for details on creating change requests.

Creating and Relating Development Packages

On the **Dev Packages** page, you can list and display details on the development packages associated with the project.

Click the item id of any dev package to display more information about it. Click the **Create Dev Package** button to create a new package.

See Creating New Packages [page 40] for details.

Configuration Management Tools

You can view details of the Configuration Management tools being used for this project by clicking the **Other Tools** tab. You can see whether Dimensions CM or ChangeMan ZMF are being used to track change requests, and for Dimensions CM which product and project/stream is being used.

Project Schedule

You can view the scheduled and actual dates for the project on a project's **Schedule** tab.

Creating and Working on Change Requests

From the Dev Change Request tab, you can list dev change requests and select a specific change request to edit it.

You can filter the list according to various criteria.

By clicking the arrow for the **View** option, you can select:

- *Mine* to view change requests owned by you.
- Select a state to view only dev change requests in a specific lifecycle state.

By clicking the arrow for the Project option, you can select dev change requests that belong to a specific project.

To open a change request, select the corresponding row.

Creating Change Requests

You can submit a change request either:

- By clicking the **Actions** button and selecting New Change Request.
- By selecting an ALM project from the Projects tab, selecting the **Change Requests tab**, and clicking the **Create Dev CR** button. In this case you will not see the **Associate to Project** fields as the change request will automatically be relate to the selected project.

When you submit a new change request, a *Create Dev Change Request* form appears.

If you clicked the **Create Dev CR** button from a selected project, you will not see the **Associate to Project** fields as the change request will automatically be related to the selected project, otherwise you will need to select a project

You set a number of core properties for the request. You can set most of these later on; only the fields with red labels are required. You can set options such as:

- From the **Change Request Type** list, the type of request such as **Defect**.
- The title and description.
- Business priority, severity, and acceptance criteria for resolving the request.
- Specific users in the various user roles, such as the **Development Manager** or **Build Engineer**.

Working on Change Requests

Open a change request to update and work on it.

From a change request, you can:

- Review information about the change request, including its change history, Quality Center relationships, associated tasks, associated change requests, and associated development packages by selecting the relevant tabs.
- Update the title, description, assignment, and other properties by clicking the **Edit** link.
- Transition the request to a different state, for example click the **Under Work** button to transition it to the Under Work state.
- If you are using the Products and Components app you can display the affected product and relate the change request to one or more affected components.
- Click the **Create Task** button on the **Tasks** tab to create a task that will then be related to this request, and that you can then assign to the specific resource who will do the work. View the Activity Log on the **History Log** tab to see the actions that have been performed on the change request. Click the arrows at the left of a row to expand the details.

Creating and Working on Tasks

From the Dev Tasks tab, you can list, update, and report on development tasks. You can relate tasks to projects in the **Projects** tab, and relate tasks to Dimensions CM tasks.

By clicking the arrow for the **View** option, you can select:

- *Mine* to view tasks owned by you.
- Select a state to view only tasks in a specific lifecycle state.

By clicking the arrow for the **Project** option, you can select dev change requests that belong to a specific project.

To open a change request, select the corresponding row.

Creating New Tasks

You can submit a new dev task either:

- By clicking the **Actions** button and selecting *New Task*.
- By selecting a Change Request from the Change Requests tab, selecting the **Tasks tab**, and clicking the **Create Task** button. In this case you will not see the **Associate to Project** fields as the change request will automatically be related to the selected project.

When you create new development tasks, the tasks are synchronized either to new requests in Dimensions CM of type *Task*, or new or existing change packages in ChangeMan ZMF. Development engineers will relate these tasks to the code changes they make in whichever Configuration management tool they are using.

When you submit a new task, a form like the following appears.

When you create a new task, you set core properties for the task, including:

- From the **Task Type** field, the type of task, such as **Design** or **Development**.
- The title and description.
- The owner and estimated effort.
- If you are using the Products and Components app you can relate the change request to one affected component.

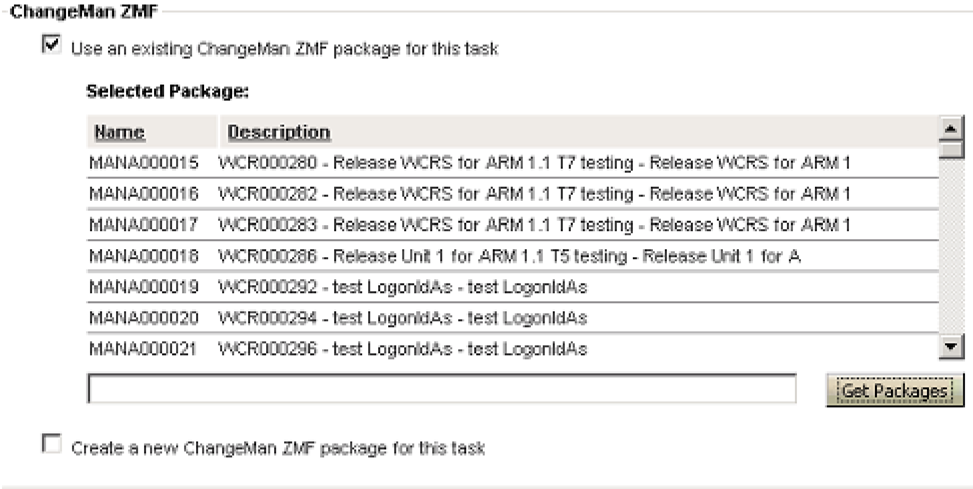
If the task type is Development, you will need to select a **Configuration Management System**. The selection available here depends on what was selected for **Configuration Management Settings** for the owning project, and by default, can be Dimensions CM or ChangeMan ZMF.

Creating a Dimensions CM Task

If you select *Dimensions CM* for the **Configuration Management System**, a new Dimensions CM task request will be created in the associated Dimensions CM project or stream when you save the task.

Creating a ChangeMan ZMF Task

When you have selected *ChangeMan ZMF* for the **Configuration Management System**, you will have the option of either selecting an existing ZMF change package or creating a new one. If you are selecting an existing one, click the **Get Packages** button, and select one from the list. Note that there is by default a one-to-one relationship between ChangeMan ZMF packages and Dev Tasks, so the list will only show packages that are not associated with Development Control.



ChangeMan ZMF

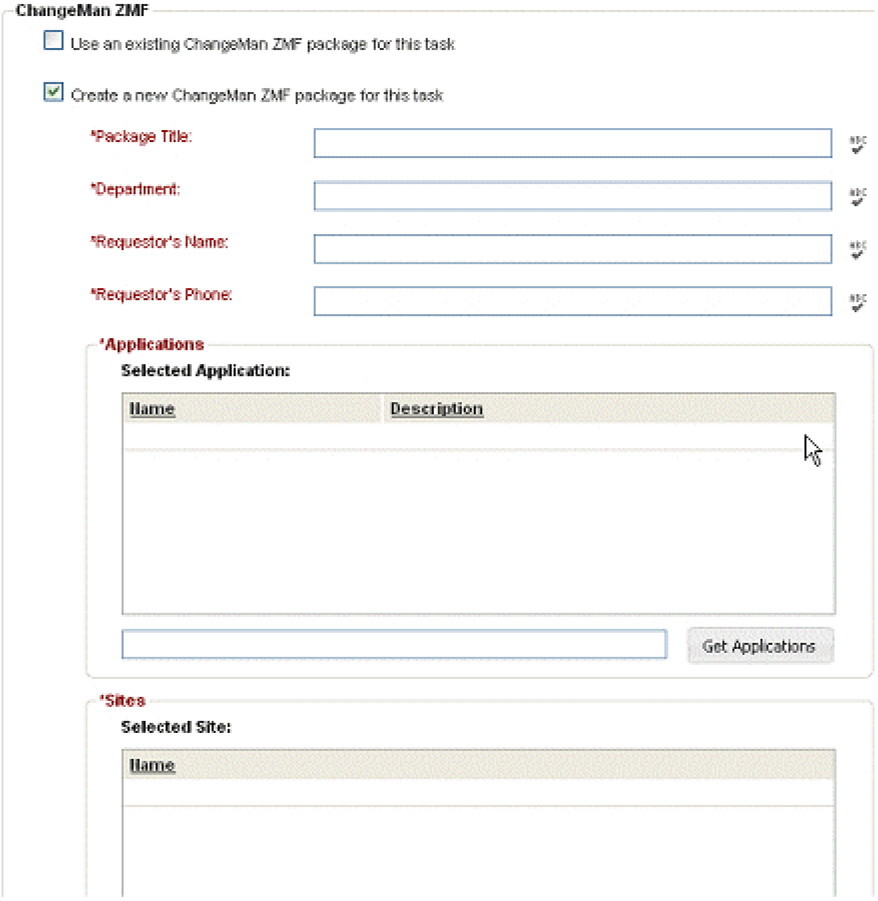
☑ Use an existing ChangeMan ZMF package for this task

Selected Package:

Name	Description
MANA000015	WCR000280 - Release WCRS for ARM 1.1 T7 testing - Release WCRS for ARM 1
MANA000016	WCR000282 - Release WCRS for ARM 1.1 T7 testing - Release WCRS for ARM 1
MANA000017	WCR000283 - Release WCRS for ARM 1.1 T7 testing - Release WCRS for ARM 1
MANA000018	WCR000286 - Release Unit 1 for ARM 1.1 T5 testing - Release Unit 1 for A
MANA000019	WCR000292 - test LogonIdAs - test LogonIdAs
MANA000020	WCR000294 - test LogonIdAs - test LogonIdAs
MANA000021	WCR000296 - test LogonIdAs - test LogonIdAs

☐ Create a new ChangeMan ZMF package for this task

If you select **Create a new ChangeMan ZMF package for this task**, a new one will be created when you save the task. You will need to enter various details for the ZMF change package.



ChangeMan ZMF

□ Use an existing ChangeMan ZMF package for this task

☑ Create a new ChangeMan ZMF package for this task

*Package Title:

*Department:

*Requestor's Name:

*Requestor's Phone:

*Applications

Selected Application:

Name	Description

*Sites

Selected Site:

Name

Working on Tasks

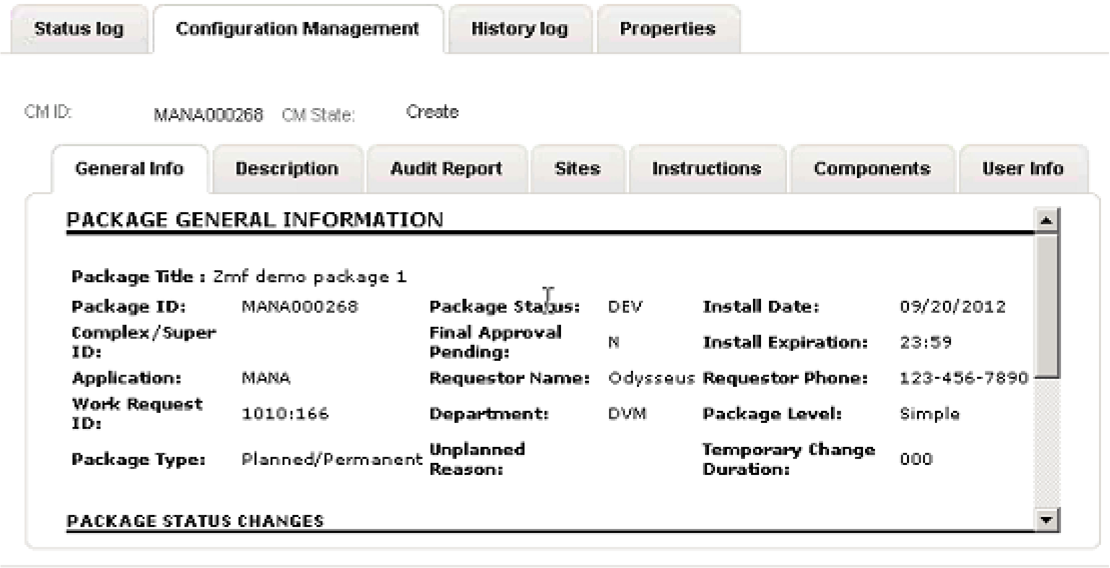
Open a task to update and work on it.

From a task, you can:

- Review information about the task, including its change history, Dimensions CM relationships, Quality Center relationships, associated change requests, and attachments.
- Transition a task from one state to the next. For example, to start working on a new task, click the **Begin Work** button.
- Click the **Edit** link to modify properties such as the owner of the task, estimated work, and attachments.

On the **Configuration Management** tab, you will see the details of the Dimensions CM request or the ChangeMan ZMF package.

For a ZMF package you will see a number of tabs relating to the selected package. The Development Control generated ID relating to the package in ChangeMan ZMF is shown in the **Work Request ID** field.



CM ID: MANA000268 CM State: Create

PACKAGE GENERAL INFORMATION

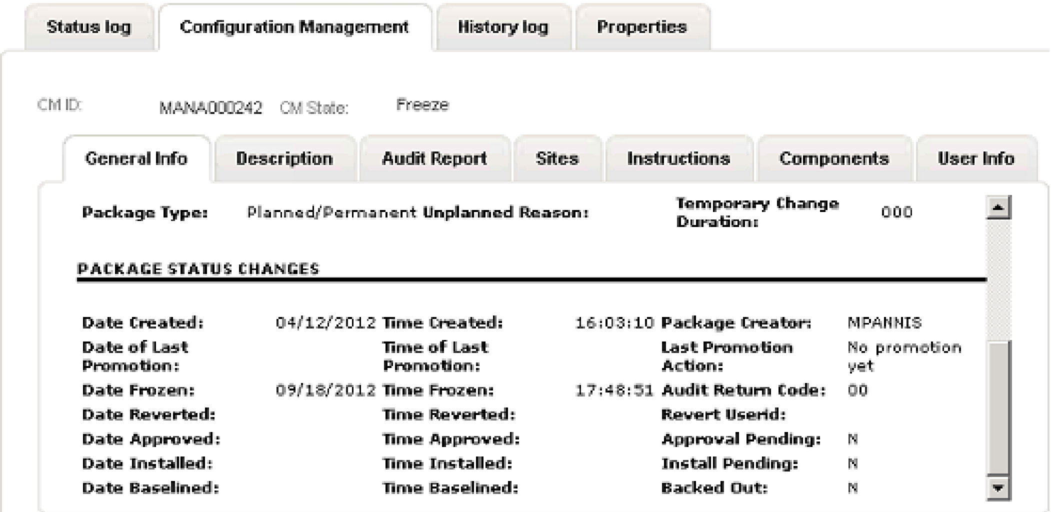
Package Title : Zmf demo package 1

Package ID:	MANA000268	Package Status:	DEV	Install Date:	09/20/2012
Complex/Super ID:		Final Approval Pending:	N	Install Expiration:	23:59
Application:	MANA	Requestor Name:	Odysseus	Requestor Phone:	123-456-7890
Work Request ID:	1010:166	Department:	DVM	Package Level:	Simple
Package Type:	Planned/Permanent	Unplanned Reason:		Temporary Change Duration:	000

PACKAGE STATUS CHANGES

Other fields of note can be seen farther down under Package Status Changes:

- The **Audit Return Code**, if not blank, shows that the package has been audited.
- The **Date Frozen** and **Time Frozen** fields, if present, show that the package has been frozen.



CM ID: MANA000242 CM State: Freeze

Package Type:	Planned/Permanent	Unplanned Reason:		Temporary Change Duration:	000

PACKAGE STATUS CHANGES

Date Created:	04/12/2012	Time Created:	16:03:10	Package Creator:	MPANNIS
Date of Last Promotion:		Time of Last Promotion:		Last Promotion Action:	No promotion yet
Date Frozen:	09/18/2012	Time Frozen:	17:48:51	Audit Return Code:	00
Date Reverted:		Time Reverted:		Revert Userid:	
Date Approved:		Time Approved:		Approval Pending:	N
Date Installed:		Time Installed:		Install Pending:	N
Date Baselined:		Time Baselined:		Backed Out:	N

Lifecycle of Dimensions CM tasks

Once created and delegated to a development engineer, the code is worked on in Dimensions CM and the task is actioned from **Under Work** to **Peer Review** and then to **Complete**. These transitions take place in Dimensions CM.

Lifecycle of ChangeMan ZMF tasks

Once created and delegated to a development engineer, the code is worked on in ChangeMan ZMF. The change package is then Moved to **Frozen**, but it must be **audited** first. These transitions can be performed from either ChangeMan ZMF or Development Control. When the package is **Frozen** it is automatically transitioned to **Complete** in Development Control.

Products and Components

If you are using products and components and the Dev Task affects a component that has a related CM design part, a CM request will be created in the corresponding design part.

Creating and Working with Development Packages

From the **Dev Packages** tab, you can list, update, and report on development packages. Development packages collect together all work associated with a release. You can gather a group of change requests related to a project in a dev package. You can also associate them with a baseline in Dimensions CM. This allows you to deploy the specific files in Dimensions CM that are related to the change requests, and run any build processes required to test the changes.

By clicking the arrow for the **View** option, you can select:

- *Mine* to view dev packages owned by you.
- Select a state to view only dev packages in a specific lifecycle state.

By clicking the arrow for the **Project** option, you can select dev change requests that belong to a specific project.

To open a change request, select the corresponding row.

You can also find and track specific packages by running and creating reports from the **Reports** view. For more on reporting in SBM, see the SBM online help.

Creating New Packages

You can submit a new dev package either:

- By clicking the **Actions** button and selecting *New Dev Package*.
- By selecting an ALM project from the Projects tab, selecting the **Packages tab**, and clicking the **Create Dev Package** button. In this case you will not see the **Associate to Project** fields as the dev package will automatically be related to the selected project.

When you create a new dev package, a page like the following appears.

When you create a new package, you set core properties for the package, including:

- The title and description.
- The Release Engineer, Build Engineer, and QA Manager to assign to the package.
- Associated ALM project. This then supplies the name of the Dimensions product and project / stream.

Working on Packages

Open a package to update and work on it.

From a package, you can:

- Review information about the package, including its change history, associated Dimensions CM baselines, and associated ALM project.
- Transition the package to the next state in the workflow, for example if the package is in the **Ready To Build** state, you can click the **Build** button to transition it to the **Building** state. Note that you need the appropriate role, such as Build manager, to transition the package, otherwise the required button will not appear.
- Click the **Edit** button to change core properties for the package, including associated change requests, title and description, and the ALM project that is related to the package.

Creating New Dimensions CM Baselines

You can create new Dimensions CM baselines from a selected dev package on the Dev Packages tab. When you create new baselines, you choose which change requests to associate with the new baseline. You can also create a new revised baseline.

To create new Dimensions CM baselines:

1. From a development package that is in the **Create Package** state, click the **Create Baseline** button.

2. Enter the baseline name in the **Baseline Name** field. This is the name that will appear on the baseline in Dimensions CM. This must be defined in all capital letters.

3. In the **Selected Design Part field**, click the **Get Design Parts** button to list all options. Select a design part from which to scope items selected for the baseline..

4. In the **Selected Baseline Type** field, click the **Get Baseline Types** button and select a baseline type.

5. In the **Selected Baseline Template** field, click the **Get Baseline Templates** button and select a baseline template.

6. Search for change requests that you want to associate with the baseline, by selecting them in the **Change Requests for this Baseline** field.

   Tasks associated with these change requests are in turn related to corresponding tasks in Dimensions CM; and files that have been updated in Dimensions CM in response to those tasks will be included in the baseline.

   To use this field, enter part of the name of the change requests and click the **Find** button to locate them. Then select the required change request and click the arrow to move them to the list at the right.

   If that doesn't work, click the → button. On the dialog box that appears, click the **Lookup** button to display a list of all available change requests, or enter a specific ID or title in the Item Id or Title field.

7. If you are revising an existing baseline, you must complete the options under Revised Baseline Settings:

   - Under **Previous Baseline Project**, locate the project to which the package and baseline belong.

   - Then, locate the dev package to which the baseline is associated, under **Previous Baseline Dev Package**.

   - Finally, locate the existing baseline in the **Previous Baseline** field.

   - Based on the above searches, you can now update the change requests associated with this baseline. This will create a new revised baseline in Dimensions CM. The changes you make to associated change requests will in turn modify the tasks and associated versioned files related to the baselines in Dimensions CM.

8. Once you create the new baseline, it will be associated with the current package.

9. Once the baseline has been successfully created in Dimensions CM, the dev package will be transitioned to *Review and Verify*. At this point the Build Manager will verify the baseline and either select **Failed Verification**, meaning that another baseline will need to be created for the dev package, or select **Verified**, meaning that it is *Ready for Build*.

Starting a Build

You can start a build from development packages that have been verified and are in the *Ready for Build* state.

To start a build:

1. From a package in the **Ready for Build** state, click the **Build** button. Typically you will need to be logged in as the Build Manager to do this.

2. The build will be based on the baseline in the **Current Baseline** field, and will include the change requests listed in the **CRs in this Baseline** list.

3. Click the **Get CM Build Configurations** button and select a build configuration
4. Click the **Get CM Work Areas** button and select a work area.
5. Click **OK**.
6. The package is transitioned to the **Building** state while the build task is executed by Dimensions CM. Once the build task is complete, the package is transitioned either to the **Failed Build** state (if the build failed), or the **Built** state (if the build succeeded). If the build failed, you must start again with a fresh package, once the issues have been addressed. If the build succeeded, you can transition the package to the next state, **Ready For Test**, by clicking the **Send to QA** button.

Turning Over a Dev Package Using the Release Vault Integration

If you are using the Release Vault integration, you have the option of publishing the dev package for release using Serena Release Manager. For this you will need to have installed the RLM - Vault Request process app, and have created the necessary templates. For further details, see the *Release Manager Getting Started Guide* and the *Release Manager Installation and Configuration Guide*.

To create a vault template:

1. On the **Vault Requests** tab of the Serena ALM UI, click the **Actions** button, and select *New Vault Template*.
2. Enter a name for the Vault Template
3. Select **Vault**.
4. On the Options tab, enter any required filters for the **Included filters.**
   a. Type a filter expression that can include wildcard expressions. For example `**/*.`exe to include .exe files.
   b. Click **Add** to add it to the Included Filters text box. Repeat this to add further filters.
5. Repeat the above for any required **Excluded filters**.
6. Click the **Get design part** button and select a design part.
7. Click the **Get vault stage** button and select a stage to which the baseline is to be deployed.
8. Enter a **Baseline stem** for the prefix to be included in the name of the baseline.
9. On the Target tab, Click the **Get DimCM Project** button and select a project from the list. This is the project to which the baseline will be deployed.
10. Click **Save**.

You can also create a template by creating a vault request and saving it as a template.

To turn over a release package:

When a dev package has passed testing and reached the **Ready for Turnover** state the final transition in Development Manager is **Turnover**.

1. Click the **Turnover** button. This displays the **Turnover** page:
2. If you want to publish the dev package to the Release Vault, select **Yes** for **Add to Vault**
3. Select a template from the **Vault Template** list.

   The vault template determines which files are included in the baseline for turnover. You can click the button next to the template list to view or edit the template.

The **Target Product** is specified in the configuration file `dm_qlarius.properties`. For details, see *Designating the Details for Each Provider* in the *Development Manager Installation and Configuration Guide*.

The **Options** specify which items are included or excluded, the design part and the stage.

4. Click **OK** on the **Turnover** page. The dev package will move to **Turnover Complete**.

A baseline will be created in Dimensions CM under the specified target product.

The baseline will now be available to be specified as a deployment unit in Serena Release manager. Note that Serena Release Manager identifies and lists the deployment units based on the deployment unit projects that are selected for the release package, and any filters on status that are defined in the configuration files.

For details, see the *Serena Release Manager Getting Started Guide*.

Displaying Project Metrics with the Serena® Dashboard

Serena Development Manager can be used with Serena Dashboard, a rich, graphical reporting tool that allows you to carefully monitor key performance indicators (KPIs). The Dashboard is fully configurable, and can provide graphical information on any aspects of your projects that managers or executives need to track.

Documentation is provided separately for Serena Dashboard.

Getting to Know Dimensions CM

With Dimensions CM, you can manage your source code assets by collaborating on shared development projects and streams with your team members. You can synchronize your local workspace with Dimensions CM, automatically delivering your changes to the shared repository, as well as copying other users' changes to your workspace. With Dimensions CM, you can collect files related to a milestone, such as a release build, into a baseline that can be deployed for testing or release.

With Development Manager, you can synchronize tasks from the Dev Tasks to Dimensions CM requests. Developers can then work on tasks in context of their favorite integrated development environment (IDE). Dimensions CM provides a number of different clients to choose from, to best suite the different working styles on your team. You can use the Dimensions Web client, desktop client, IDE clients, or even command-line clients depending on what best suits you.

To get started learning more about Dimensions CM, please see *Getting Started with Dimensions CM*.

Integrating with HP Quality Center

Use the Development Manager Connector for HP Quality Center to synchronize defects between Dev Change Requests and Quality Center. When users submit and update defects in one system, the defects are automatically submitted and updated in the other system. In this way, if your QA teams use Quality Center to record test results, you can ensure that all work against a project is tracked in Serena Development Manager.

The heart of the Connector for HP Quality Center is a set of Web Services that provide an API to key Quality Center features, such as submitting and updating defects. SBM process apps can interact with Quality Center using these Web services.For instructions on installing and configuring the Connector, please see the *Development Manager Connector for HP Quality Center / ALM*.

Using Products and Components with Development Manager

Introduction to Products and Components

The Products and Components workflow can be used by any teams performing OALM or IT development. It can also be used to develop software that is integrated and embedded into an electronic system, which in turn is integrated with mechanical systems. Using Products and Components allows you to:

- Model the logical structure of the software using components.
- Bring control and visibility to changes being made to the software.

About Products

A *product* identifies a logical software system, typically something that can be shipped, for example, a vehicle climate control system. Products can represent systems that may continue to exist for many years over the duration of multiple projects. For example, different versions of a vehicle climate control system may be installed in the current and future versions of a vehicle model. Each version may be a different project, and the climate control system itself is a product. A product is typically comprised of a hierarchy of multiple components and sub-components.

Products Lines

Products can have an optional *Product Line* that enables your organisation to group specific categories of products. For example, there might be a Sports product line that is used to group the *Lightweight Sports Climate Control System*, *High-powered Sports Climate Control System*, and *Luxury Sports Climate Control System*.

Product Lines are auxiliary items that you can edit in the SBM Administration Portal. Alternatively, if you have permission, you can create a Product Line in the Create/Edit Product dialog box.

About Components

A *component* identifies a standalone piece of code, usually something that can be built, tested and delivered. For example, a file containing the power settings for the blower in the vehicle climate control system. Components can:

- Be broken down into multiple sub-components. The parent component must be in the state *In Use*.
- Reuse other components.

Components can be linked to design parts in Dimensions CM and have a similar purpose. A product can be linked to a Dimensions CM product and a component can be linked to a CM design part. These links are not mandatory.

Note: You can use components without Dimensions CM however you will not benefit from the extra value that CM provides.

Component Types

Components have an optional Component Type that allows you to group your components into logical sets. For example:

- In-House
- Application
- Contract
- COTS (Common off the Shelf)
- OEM
- OSS (Open Source Software)

Component Types are auxiliary items that you can edit in the SBM Administration Portal. Alternatively, if you have permission, you can create a Component Type in the Create/Edit Component dialog box.

Suppliers

Components have an optional Supplier that allows you to keep track of which components came from third-party suppliers. Supplier is an auxiliary item that you can edit in the SBM Administration Portal. Alternatively, if you have permission, click the arrow in the Supplier section in the Create/ Edit Component dialog box.

Part Numbers

Each component is assigned a part number. You can enter a number manually or allow the system to generate one automatically.

Component Relationships

All components are owned by one parent product. Sub-components are owned by top-level components or other sub-components. This relationship is created automatically when a sub-component is created from a parent component. Only top level components do not have an owning component.

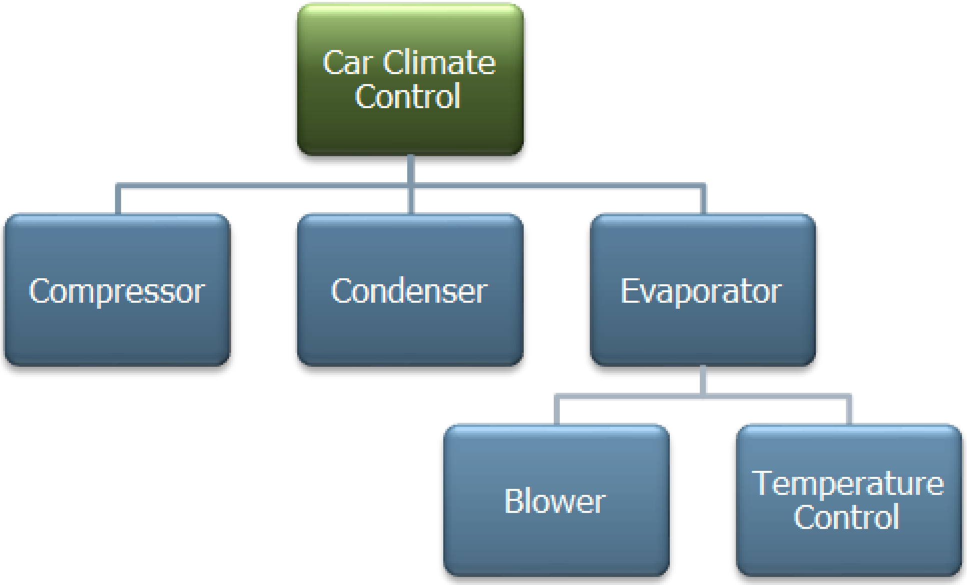
The "Used in" relationship between two components allows you to model the reuse of components within a product, or between products.

Products and Components Example

In the following example, the product Car Climate Control has three top level components.

- Compressor
- Condenser
- Evaporator

These top level components have the “Owned by” relationship with the product.



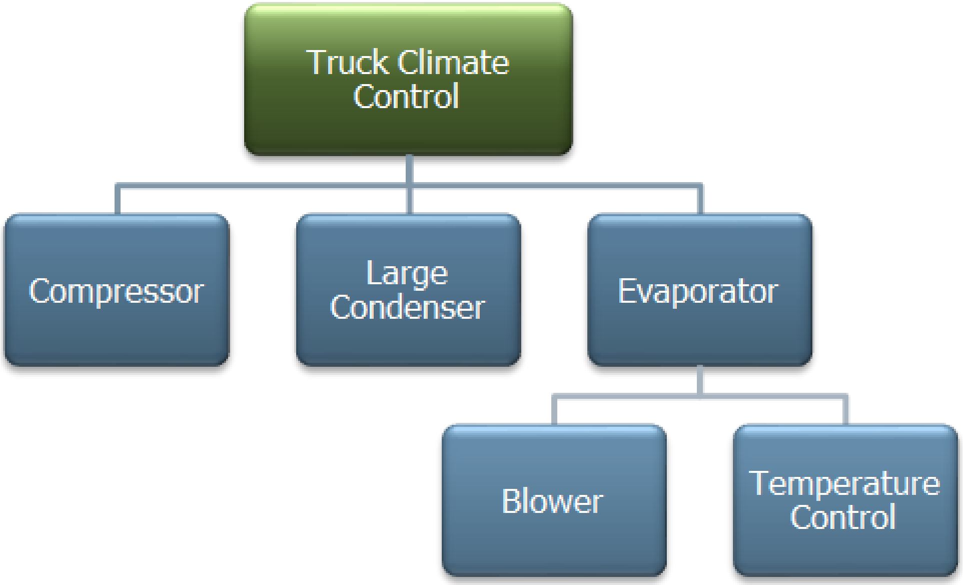
The Evaporator parent component has two sub-components:

- Blower
- Temperature Control

In this example the sub-components have the "Used in" relationship with the Evaporator parent component.

Re-Use of Components

In the following example, the Truck Climate Control product reuses all of the components from the Car Climate Control product. None of the components are modified apart from the Condenser



The truck requires a different, larger condenser than the one used in cars. This requires modifications to the software

What are the Benefits of Using Products and Components?

Products and components bring multiple benefits to your software development process.

Increased Visibility

You can use Serena Dashboard to display reporting and metrics about:

- Products or components that have the most defects.
- Work product approval trends.
- Action closure rate.
- Defect and error trends.

Control and Track Changes

You can control and keep track of what can be changed.

- You can relate development change requests (DCR) and development tasks to projects, products and components.
- You can indicate which components a project can change. You can limit which components the development change requests in a project can change. You can also limit the components that development tasks in a development change request can change.

Increase Structure and Process

Products and components bring increased structure and process to your software development.

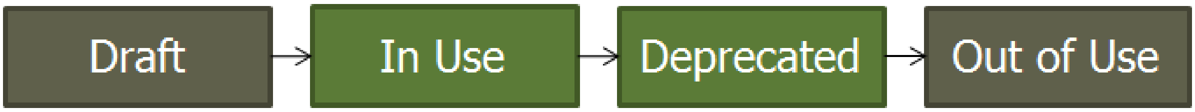
Uses the SBM Role System

Products and components use the following SBM roles:

- Component Manager
  - Can create products and components.
  - Can update products and components.
  - Can be a candidate for the Component Manager field on a component.
- Component Authority
  - Can be a candidate for the Component Authority field on a component.
  - Is the subject matter expert for a component.
  - Can give this role to members of a development team.
- Viewer
  - Can view products and components.
  - Is a required role to see products and components in ALM projects, DCRs, and development tasks.

Shared Workflow

Products and Components share a workflow and states:



The *In Use* and *Deprecated* states are active; all the other states are inactive.

This workflow:

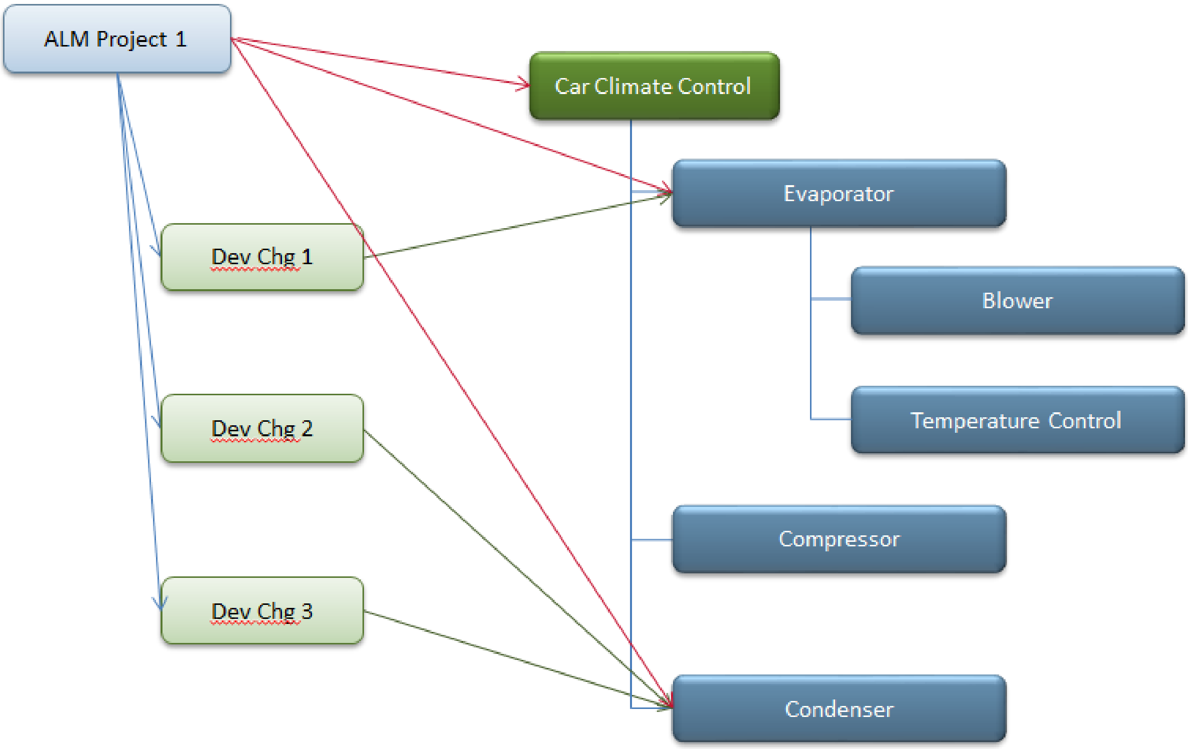
- Allows for a simple maturity lifecycle controlled by the Component Manager.
- Affects how you can use a product or component

Linking to ALM Projects and Development Change Requests

After you have modelled your product and components you can relate them to an ALM project. This gives you control and visibility over the parts of the system that the project will change.

Note: If you are relating ALM projects, DCRs, and development tasks to Dimensions CM design parts you must have a Dimensions CM user account.

The illustration below shows the link between an ALM project and its DCRs, and the product and its hierarchy of components:



Linking with ALM Projects

An ALM Project has a single “Affected Product”. This is the shippable software system that you update with a project. The product is inherited by the child DCRs and development tasks and influences which components are affected.

An ALM Project has a set of “Affected Components”. These are the components that are available to be affected by DCRs and development tasks.

An ALM Project has a set of “Used Components”. These are the components the ALM project expects to make use of in builds, but does not anticipate will change. “Used Components” are for information only.

Each “Affected Component” for a project must be:

- Part of the affected product, or,

- Be a component from another product that has a "Used In" relationship with a component from the affected product.

The same applies to "Used Components".

Linking with Development Change Requests

A development change request can have a set of "Affected Components". These are the components that the DCR is allowed to change. Each affected component for a DCR must be an affected component of the DCR's ALM parent project (or their child components).

Note: Only "Affected Components" can be changed by child development tasks.

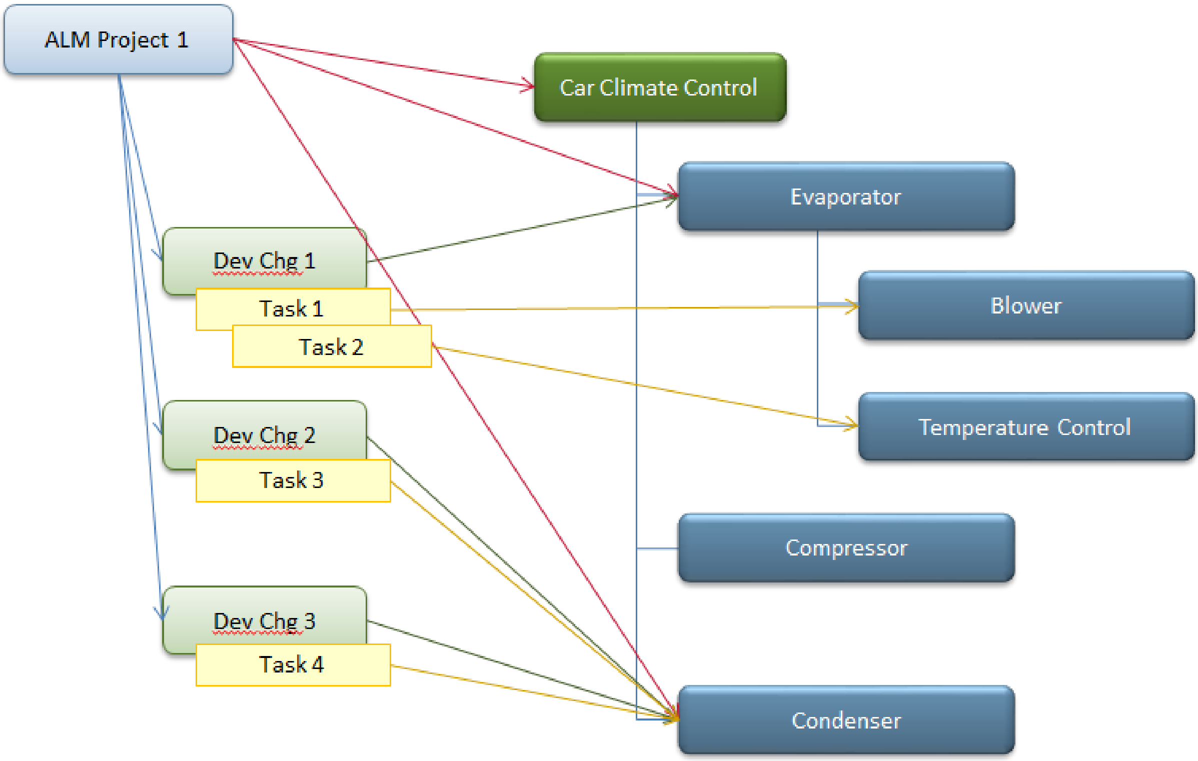
Linking to Development Tasks and Dimensions CM

A product can be linked to a Dimensions CM product. However, the link to CM is optional and products and components can be used independently of source control system.

After you have created development change requests you can create child development tasks. A development task can only have a single "Affected Component" and must be one of the affected components of the parent DCR (or their child components). The development task is then used to modify the relevant source code.

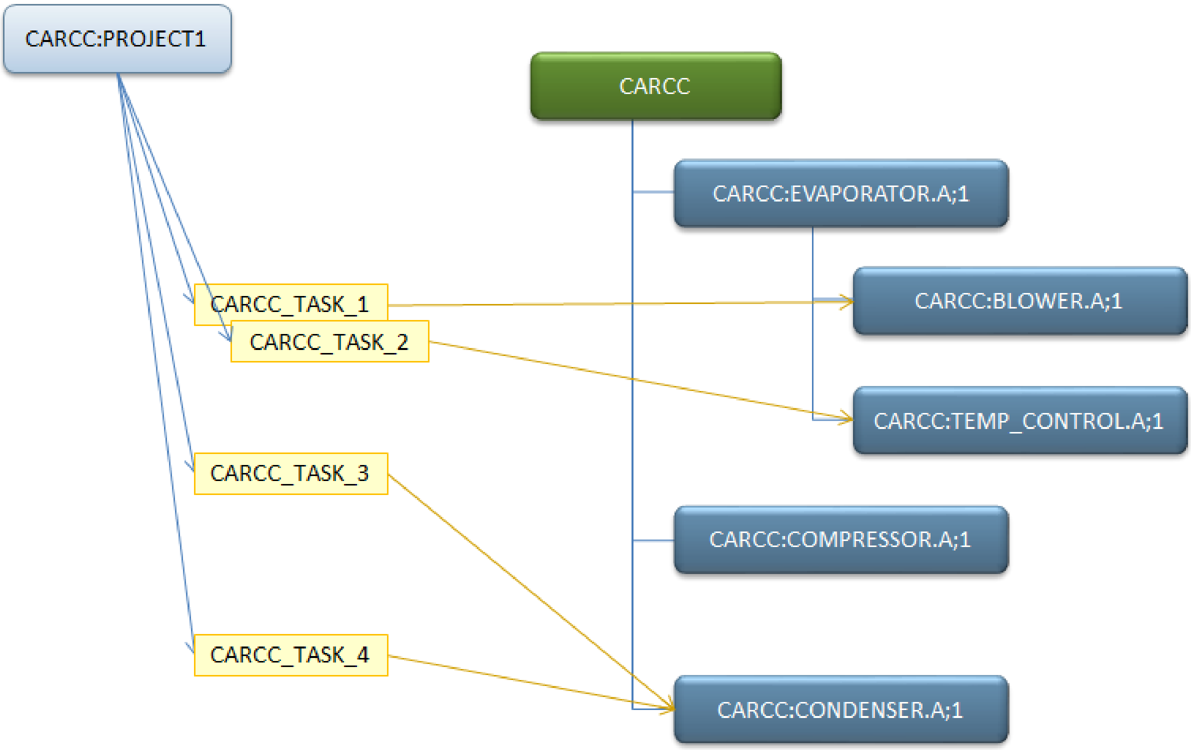
When you create a development task a Dimensions CM task request is automatically created and related to that task. If a development task has an "Affected Component" that is linked to a valid CM design part, the CM task request will also be related to that design part.

The illustration below shows the links in DVM between development tasks and affected components:



For example, the development change request *Dev Chg 1* has one affected component, the Evaporator. Therefore its child development task *Task 1* can affect the Evaporator but not the Compressor or Condenser components.

The illustration below shows the links in CM between task requests (on the left) and design parts (on the right):



Working with Products and Components

Note: Unless otherwise specified, to perform the actions described below you need to have the Component Manager role.

To create a new product:

1. Click Actions and select New Product. The Create Product dialog box appears.
2. In the Name field enter a name for the new product.
3. To specify a Product Line, optionally enter a search value in the field, and click Find. You can also use the advanced search. Select a product line from the list.
4. In the Description field enter a description for the new product.
5. From the Component Manager list select a user who will be responsible for the product's components.
6. From the State list select In Use. This allows components to be created for the product.
7. Click Get CM Products and select a Dimensions CM product from the list. **Note**: The user completing this field must be a registered Dimensions CM user.
8. To create the new product click OK.

To view and edit a product's properties:

1. On the Products tab do one of the following:
   - To view in a new window, double-click the product.
   - To view in the bottom pane, select the product.
2. To edit the product click Edit. The Edit dialog box appears.
3. Edit the properties as required.
4. Click OK. If you have opened a new window, close it.

To create a new component:

1. Click Actions and select New Component. The Create Component dialog box appears.
2. In the Name field enter a name for the new component.
3. To specify a Component Type, optionally enter a value, and click Find. You can also use the advanced search. Select a component type from the list.
4. In the Description field enter a description for the new component.
5. From the Component Authority list select a user who will be the subject matter expert for this component.
6. To specify a Supplier, optionally enter a value, and click Find. You can also use the advanced search. Select a supplier from the list.
7. In the Part Number field enter a part number. Or leave it blank and a number will be generated automatically.
8. In the Owned by Product field, optionally enter a value, and click Find. You can also use the advanced search. Select a product from the list that will own this component.
9. Click Get CM Design Parts and select a design part to be related to the component.

   Notes:

   - The user completing this field must be a registered Dimensions CM user.
   - The product must have one or more design parts.
10. To create the new component click OK.

To edit a component:

1. On the Components tab do one of the following:
   - To view in a new window, double-click the product.
   - To view in the bottom pane, select the product.
2. To edit the component click Edit. The Edit dialog box appears.
3. Edit the properties as required.
4. Click OK. If you have opened a new window, close it.

To create a sub-component:

1. On the Components tab select a parent component.
2. If the parent component is in the Draft state, click In Use.
3. Click Create Sub-Component.
4. Complete the fields, see here [page 51] for details. **Note**: You cannot change the Owned by Product field as this is inherited from the parent component.
5. To create the sub-component, click OK.

To delete a component or sub-component:

You can only delete a component or sub-component if it is at the Draft or Out of Use state.

1. On the Components tab select the component or sub-component that you want to delete.
2. Click Delete.
3. To confirm the deletion, click Delete again.

To create or remove a used-in relationship between components:

1. Select a component or sub-component and click Edit.

2. From the Used in Components list select, or unselect, the parent component(s) that will use this sub-component.
3. Click OK.

To change the ownership of a component or sub-component:

1. Select a component or sub-component and click Move.
2. In the Move Component dialog box select the new owning component.
3. Click OK.

To add components to an ALM project:

Typically, a project manager creates an ALM project. A development manager can then edit the project and add affected and used components.

1. As a development manager, select an ALM project and click Edit.
2. In the Components section, find and select an affected product.
3. Click Get Components.
4. Select the affected and used components for this project.
5. Click OK.

To display a project's component information:

1. On the Projects tab select a project. The project is displayed in the bottom pane.
2. Select the Components tab.

To add affected components to a Development Change Request:

As a development manager, create or edit a development change request in the usual way. Perform these additional steps to add, or remove, components that will be affected by the DCR:

1. (If you are creating a new DCR) In the Components section click Get Affected Components.
2. Select or deselect affected components.

Note: You cannot change the affected product as this is inherited from the ALM product.

To add affected components to a Development Task:

The associated development change request must be related to at least one affected component. You can only select one component to be affected by a development task.

As a development manager, create or edit a development task in the usual way. Perform these additional steps change components that will be affected by the task:

1. (If you are creating a new task) In the Components section click Get Affected Components.
2. Select or deselect an affected component.

Understanding the Default Workflows

Object Types and Relationships

With the included workflows, Development Control orchestrates your development work across all of your inter-related systems.

NOTE The SBM process app names in which the workflows are defined include a prefix that indicates the solution to which they belong. The internal names for workflows and projects may differ from the process app name.

The workflows of Development Manager include:

- **ALM Projects**: Use this workflow to define and track development projects. This is 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 workflow, which you can then relate to source code assets in Dimensions CM or ChangeMan ZMF. 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 or ChangeMan ZMF.

- **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.

- **Dev Control Orchestrations**: This workflow ties together the other workflows using Serena Business Manager orchestrated workflows.

- **Products and Components**: Define and manage a logical view of the parts, or components, of your system. Bring control and visibility to the changes you are making to these components.

- **Agile Planning**: Use Agile planning methodology to manage backlogs of stories and tasks, and optionally link them to your development change requests and tasks.

The SBM workflows included with Development Control enable you to manage projects, change requests, tasks, and release packages. Each of these object types can be related to specific items in external systems.

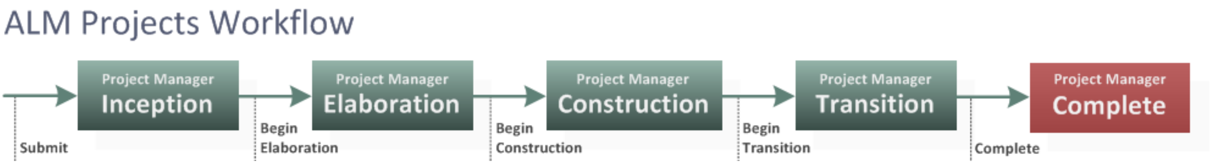
- **Projects**: You manage projects using the ALM Projects workflow. You can relate projects as follows:
  - By relating a project to Dimensions CM streams and projects, you enable the association of source code and other managed assets related to the project.

  - By relating a project in ALM Projects to change requests in the Dev Change Requests workflow, you ensure that all work to be completed in context of the project is associated with the project. You also provide scope for the creation of Dimensions CM tasks and items that are needed for this project.
  - By relating a project to packages in the Dev Packages workflow, you ensure that all completed work included in releases are associated with the project.

- **Change Requests**: You manage change requests using the Dev Change Requests workflow. You can relate change requests as follows:
  - Relate change requests to tasks in the Dev Tasks workflow. You can then manage all development work from the related tasks.
  - If you use HP Quality Center / ALM to manage defects, associate requirements and defects to defect type change requests in Development Control.
  - If you use Serena Requirements Manager, associate requirements to dev change requests. You can associate multiple requirements to a single dev change request or multiple dev change requests to a single requirement.
- **Tasks**: You manage tasks using the Dev Tasks workflow. You can relate tasks as follows:
  - Relate tasks to change requests in the Dev Change Requests workflow to maintain a history of which tasks were associated with the requests.
  - Relate tasks to requests in Dimensions CM in order to track specific code changes. All updates to files stored in Dimensions CM are tracked in Dimensions CM tasks; this information is then synchronized to tasks (and their associated requests) in the Dev Tasks workflow.
- **Development packages**: You manage development packages using the Dev Packages workflow. Packages define the scope of a release by collecting all included change requests and tasks into one place. You can relate packages as follows:
  - Relate packages to baselines in Dimensions CM. Dimensions CM baselines collect all source code (or compiled deliverables) associated with a particular project or release. Those baselines can be deployed, using Dimensions deployment features or using Serena Release Manager.
  - Packages are also related to specific change requests. The change requests define the scope of the code included in the baseline.
- **Products**: You manage products using the Products and Components workflow. Products represent long-lived shippable software systems, which may continue to exist over the duration of multiple ALM Projects.
- **Components**: You manage components using the Products and Components workflow. Components are reusable, buildable sub-pieces of a larger software system. Components are composed together to form Products.

ALM Projects Workflow

ALM Projects provides a high level view of the overall project workflow and status. In the default workflow, the Project Manager is the primary owner of each project state, but other participants could be made the primary owners in your implementation.



The states include the following:

1. **Inception**: During this state, the Project Manager creates the project.

2. **Elaboration**: During this state, the Business Analyst defines change requests for the project based on the approved requirements.

3. **Construction**: At this point, the Development Manager or Lead prioritizes the change requests and creates tasks related to the requests. The tasks are assigned to developers to implement. Developers work on the tasks, storing new versions of files in their configuration management system (such as Dimensions CM). Regular builds are compiled and installed for testing, and requests are assigned to QA to test. Once all of the requests have been implemented, the Development Manager can transition the project to the next state.

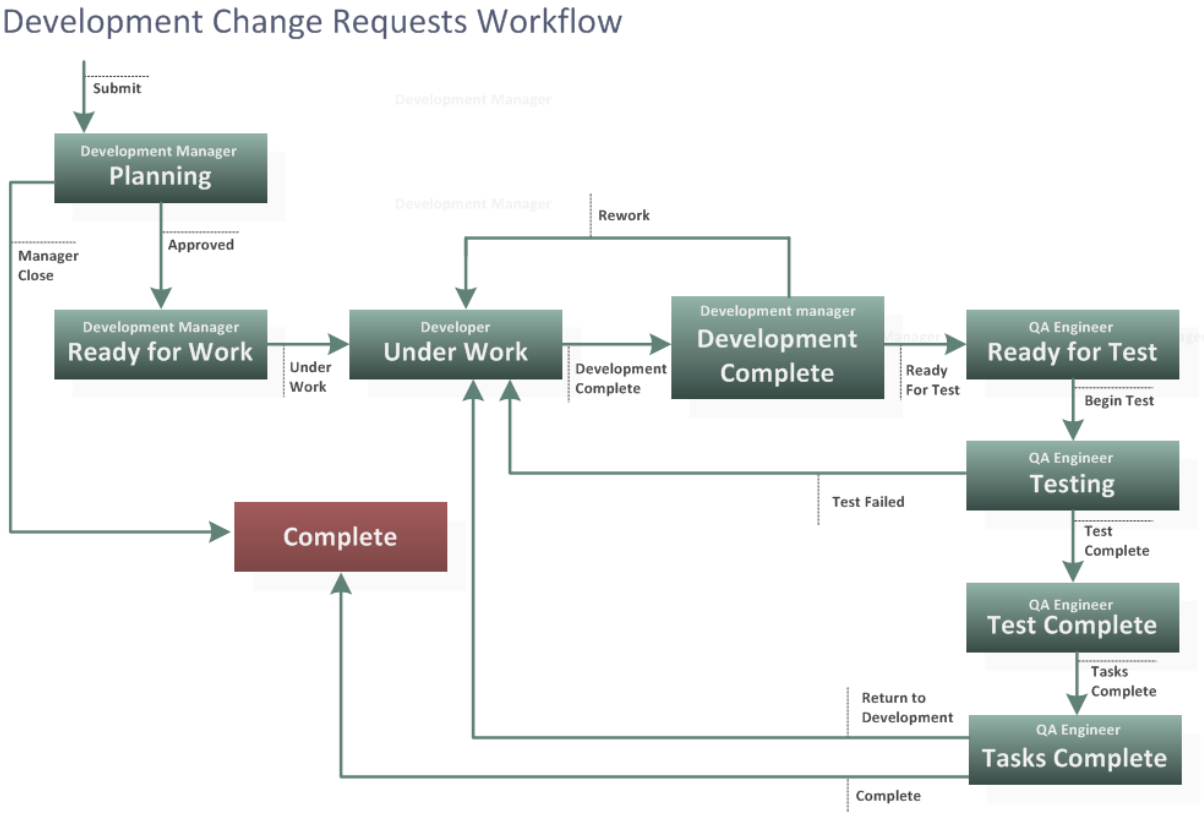
4. **Transition**: During this state, QA perform robust testing of the completed product or features, recording defects as they find them. QA may use a test management system such as HP Quality Center to track test case execution. QA submits defects, and the defects are assigned to developers to fix. QA validates fixes and closes defects as they are resolved. When all critical defects are complete, QA can transition the project into the Complete state.

5. **Complete**: During this state, the finished product is prepared for release. The final builds are collected and packaged for deployment, perhaps using Serena Release Manager. When the product is ready to deploy or release, the project can move to the final state.

Dev Change Requests Workflow

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

SBM Workflow for Change Requests



1. **Planning**: During this state, the Product Owner and Development Manager define the content of change requests based on the requirements for the project. Once the request is approved, it can be moved into the **Ready for Work** state.

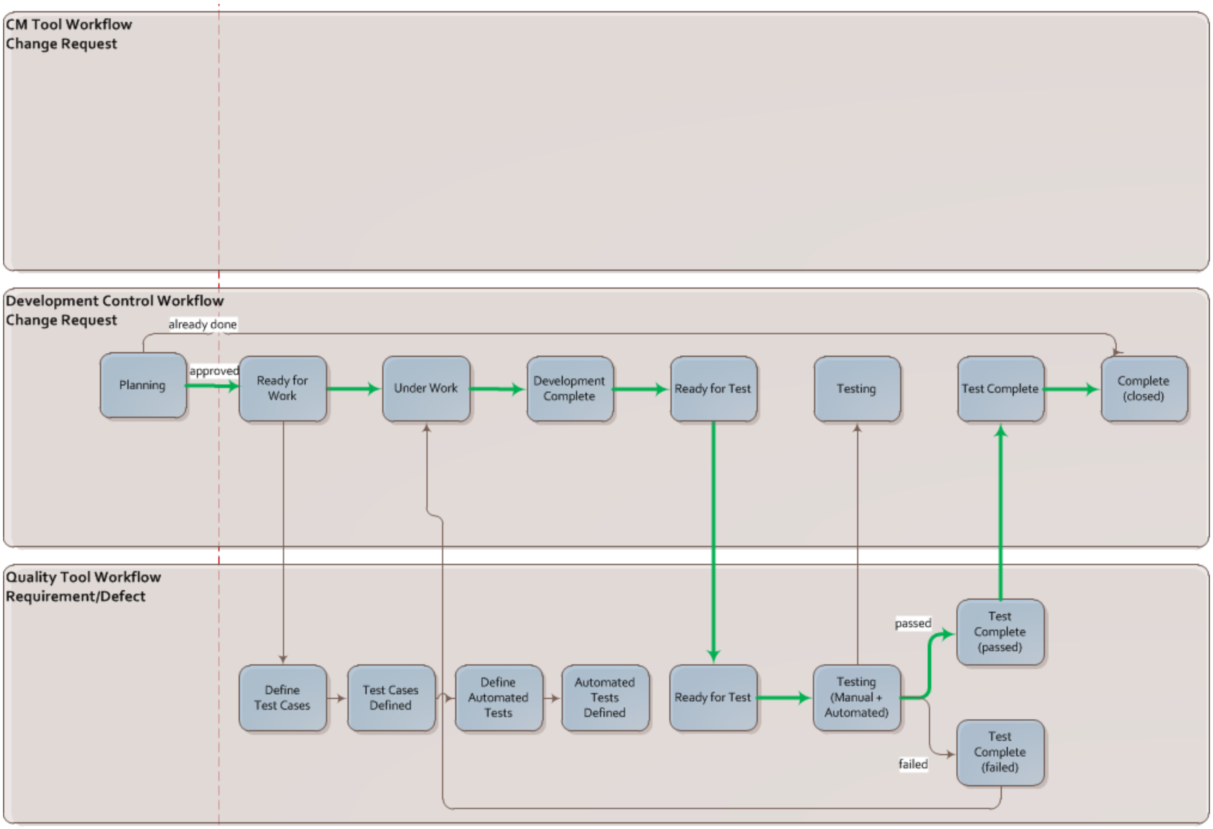
2. **Under Work**: Developers can now work on tasks associated with the request. As work is complete the developer checks updated source code files into Dimensions CM. Once this work associated with the request (and its related tasks) is ready to be compiled and tested, the request is transitioned by the developer to the next state.

3. **Development Complete**: At this state, the build engineer starts a build using the code that was created or updated during the **Under Work** state. When the build successfully completes and the request is ready to test, the build engineer transitions the request to the next state.

4. **Ready for Test**: The QA staff can now test the code that was developed and compiled to satisfy this change request. The QA manager or responsible QA engineer transitions the request to the **Testing** state. For dev change requests of type defect, testing can then be managed in HP Quality Center. If a test fails in the test case management system, the request is returned to the **Under Work** state. When testing is complete, the request can be transitioned to the **Test Complete** state, and then to the **Complete** state.

Integrating Change Requests with HP Quality Center / ALM

The following diagram illustrates the default integration points between Dev Change Requests and HP Quality Center / ALM.

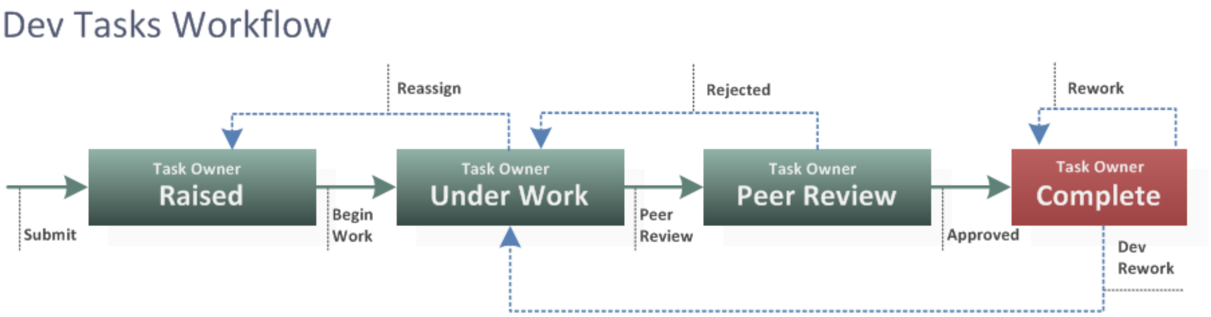


The test states in the change request workflow are mapped to workflow states in the Quality Center / ALM workflow. When the change request enters the *Ready for Test* state, it is associated with test cases in Quality Center.

Dev Tasks Workflow

Manage specific developer tasks using the Tasks workflow. 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 or ChangeMan ZMF, these states map to parallel states for change requests in Dimensions CM or change packages in ChangeMan ZMF.



The states include the following:

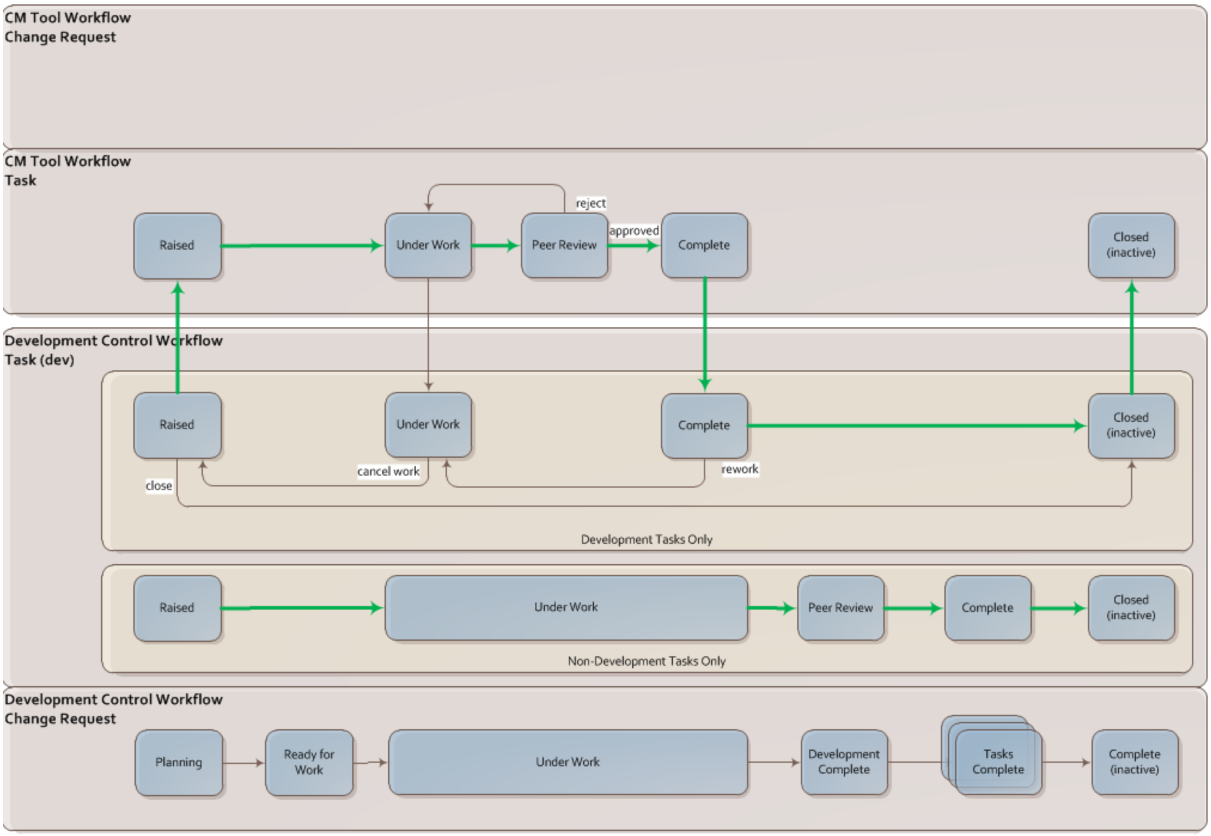
1. **Raised**: The task is submitted. If you are using Dimensions CM, a related change request is also created in Dimensions CM or ChangeMan ZMF. To start work, the Development Manager assigns an owner and transitions to the **Under Work** state. The related item in Dimensions CM or ChangeMan ZMF is also transitioned to the parallel **Under Work** state in Dimensions CM or **Development** in ChangeMan ZMF.

2. **Under Work**: As work is complete the developer checks updated source code files into Dimensions CM or ChangeMan ZMF. Information about the new and updated files is stored in the application, and that data is synchronized back to the task in Dev Tasks. All information about affected files is therefore captured in context of the change request or change package that the task belongs to.

3. **Peer Review**: A software development best practice is to include review of changes by another developer before turning the new or updated code over for testing. Once the peer has approved the changes, the the Dimensions CM task can be actioned to the **Complete** state. At this point the corresponding dev task in Development Control is also transitioned to the **Complete** state.

Task Workflow Using Dimensions CM

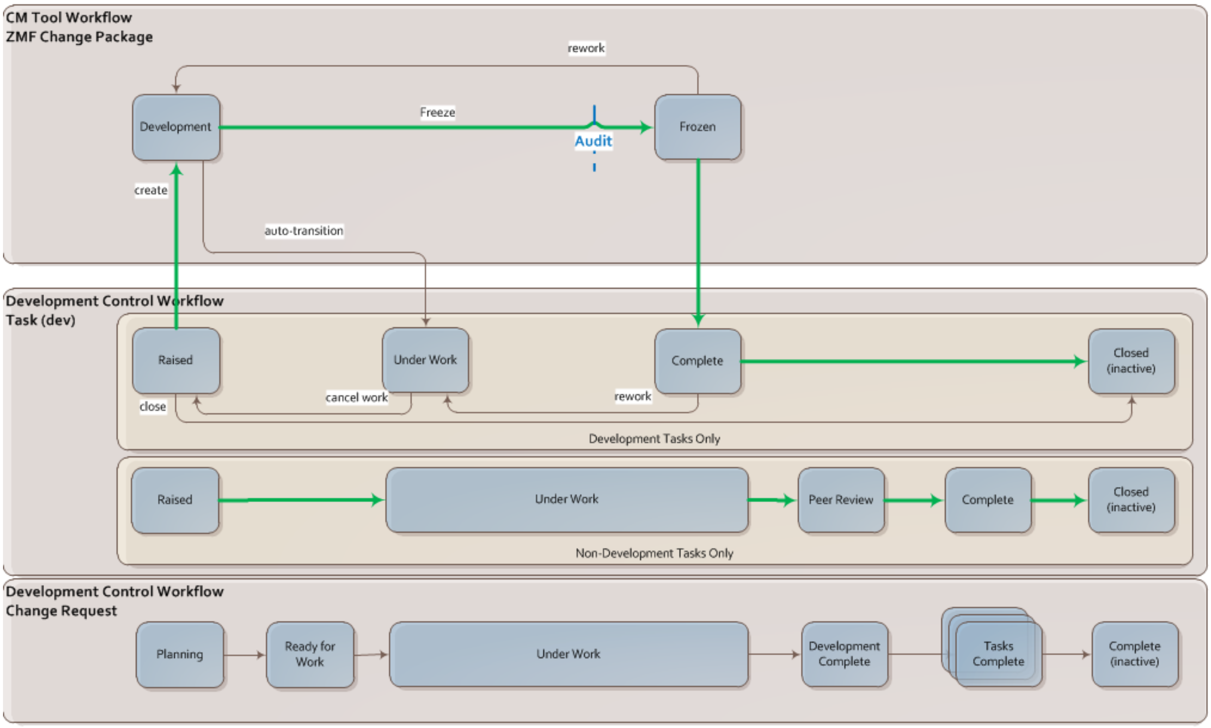
The following illustration clarifies the relationship between tasks in Dev Tasks and related tasks in Dimensions CM. Note that this is just one example of how tasks in Development Control might be linked to tasks in Dimensions CM for purposes of tracking work.



In this example, the associated task in Dimensions CM follows a parallel workflow. The request may be a specific type that is designed to synchronize with tasks in the Dev Tasks workflow. In this example, the task in Dev Tasks is transitioned to the completed state once the related task in Dimensions CM is completed, requiring all work completed in Dimensions CM to be peer reviewed and approved before the task can be closed. Transitioning the request in Dimensions CM to the **Under Work** and **Complete** states in turn automatically transitions the related task in Dev Tasks.

Task Workflow Using ChangeMan ZMF

The following illustration shows the relationship between tasks in Dev Tasks and the corresponding change package in ChangeMan ZMF. In this case, the change package is under development until work is complete, when it is frozen. In order to be frozen it must have first been audited. Freezing the change package in ChangeMan ZMF automatically transitions the task in Dev tasks to **Complete**.

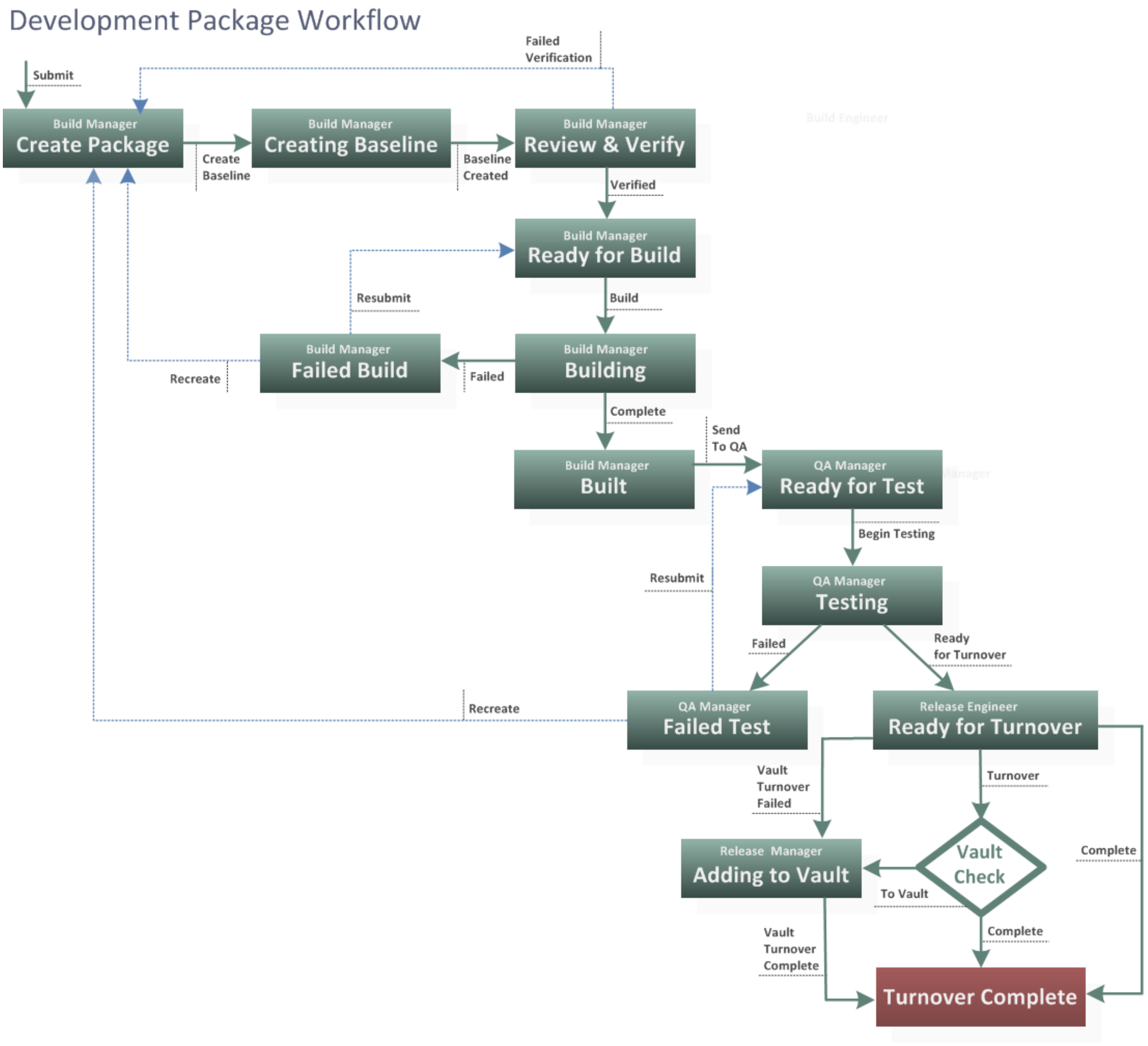


Dev Package Workflow

Use the Dev Packages workflow to manage the creation and validation of release packages and baselines, including testing builds and approving for final turnover. Most of the work managed within the Dev Packages workflow 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.

Dev Package Workflow Diagram



The workflow states include the following:

1. **Create Package**: The release engineer creates a package that defines the scope of a release. The package includes links to change requests that define the content of the package.

2. **Creating Baseline** and **Review and Verify**: A baseline is created in Dimensions CM that includes all of the files that are to be included in the release. The baseline is related to the package in Dev Packages, and the baseline follows a parallel workflow. Once the baseline and package are approved for testing, the package is transitioned to the next state.

3. **Ready for Build** and **Building**: The build manager compiles the build from the package that the release engineer turned over. If the build fails, the release engineer reviews and updates the package as needed. The baseline may need to be recreated and the build run again. If the build succeeds, the build manager transitions the package to the **Built** state, and then deploys the compiled deliverables for QA to test.

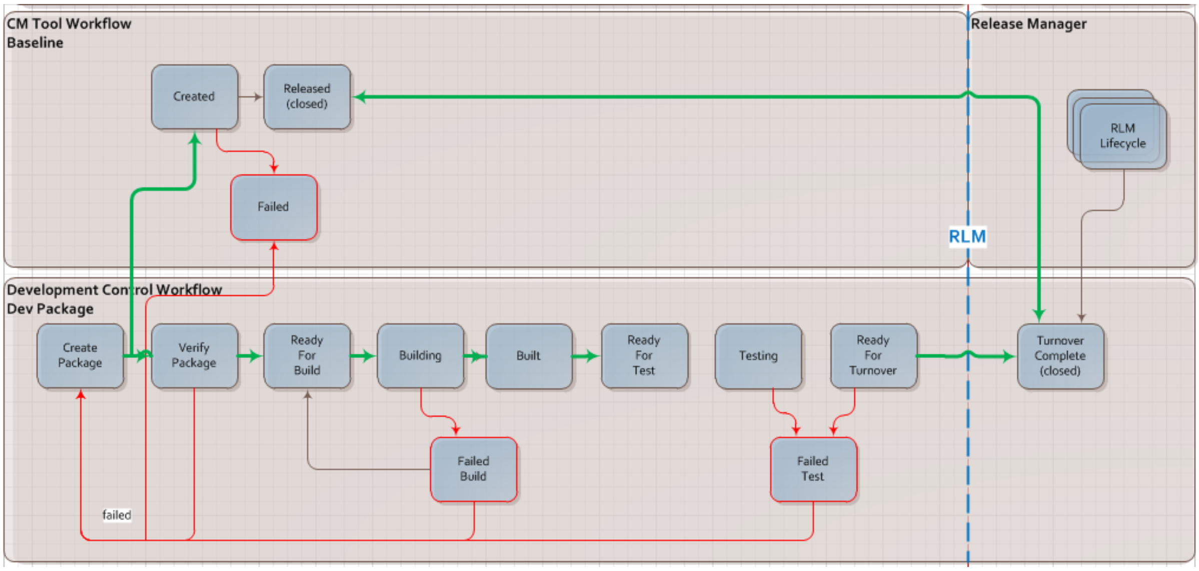
4. **Ready for Test** and **Testing**: QA staff test the deployed build. Information about the packages is synchronized to requirements in the test management tool. Test cases for validating those requirements are associated with the requirements. QA run their tests and update with pass / fail information, which is then synchronized back to the packages in Dev Packages. If a test fails, then either the build, baseline, or source code will be need to be updated to correct the defect. The team must review the results and decide who will fix any issues that the testing reveals. If the source code is updated, the build manager must re-generate the baseline, and re-run the build and deploy it. When testing is complete and the

build is approved for turnover for release, QA transitions the package to the **Ready for Turnover** state.

5. **Ready for Turnover**: At this state, the release management process takes over. For example, you may use Serena Release Manager to coordinate deployment.

6. **Turnover Complete**: If you are using the integration between Serena Release Manager and Development Manager, a turnover in the form of a Dimensions CM baseline is published to Serena Release Vault. This can be input to a release and deployment process in Serena Release Manager.

The following diagram illustrates the integration points between the Dev Package workflow and Dimensions CM. This is based on the default 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.



Products and Components

The product and component states (and corresponding SBM states) are:

- Draft (inactive)
- In use (active)
- Deprecated (active)
- Out of Use (inactive)

Supporting Your Development Process

Serena Development Manager is a powerful, highly customizable lifecycle management and reporting application. To make the best use of it, spend some time evaluating your own development processes. It is important to decide what processes and issues Serena Development Manager can help you solve, to set expectations appropriately and begin to plan roll-out and adoption. Consider creating a diagram of your development lifecycle if you do not already have one, as this can be an valuable resource when mapping your workflows in Serena Business Manager.

What Types of Development Projects Will You Manage?

In Serena Development Manager, you can define project types that correspond to specific development management needs in your organization. For example, you can define different project types for different applications, or for different departments. Consider the following when deciding what types of project you will support:

- Will Serena Development Manager support projects for different departments? Do the departments have their own lifecycles?
- What applications does your organization use, and which of these will you integrate with Serena Development Manager? For example, does your development team use Serena Dimensions CM for source code management, or another system? What about Requirements and Test Case Management?

Best Practices for Development Project Management

- Use the Dev Change Request workflow to manage defect reports alongside other change requests. If you also use HP Quality Center for defect management, you can synchronize your defects between the two systems.

Is Your Development Process Agile?

Determine what type of development practice your project types follow. Do they follow iterative Agile methodologies? Is their process closer to traditional "waterfall" methodologies, with heavier upfront analysis and planning?

What Are Your Workflow States?

Identify the states in your development workflow. You can replicate your existing states, or refine your current practices when configuring the lifecycle in Serena Business Manager. Questions you might hear include:

- Who submits new requests for different types of work (such as features, defects, etc.) to your development team, and in what form?
- What is involved with design and analysis of new features, before developers actually begin work? Who is responsible?
- How is testing managed? Is it part of the development effort, completed afterwards, or both? Who manages the test states?
- What steps are required in order to certify new work for turnover for release? Consider such things as stabilization, defect burndown sprints, and deploying to staging. Who manages these steps?

Best Practices for Defining Your Workflow

- You can customize the workflows in Serena Development Manager as needed. Study and work with the default workflows, itemize the changes that you need to make, and update them using Serena Business Manager Composer.
- Integrate testing into your development lifecycle. In the default implementation, the Serena Development Manager workflow includes a separate state for testing that occurs after development is complete. However, we strongly advocate for testing in context of development. Developers should test their own work, as well as peer review each other’s work.

How Do You Manage Test Cases?

Development Manager provides a Web Services integration to HP Quality Center, enabling you to synchronize information from your development requests in Serena Business Manager to defects in Quality Center. Please see the *Serena Business Manager Connector for Quality Center Implementation Guide* for implementation details.

What Reports Does Your Organization Need?

Serena provides as an option, Serena Dashboard, a powerful dashboard reporting solution that allows you to display rich graphical data on key performance indicators (KPIs). This enables your stakeholders, including team members, managers, and executives, to quickly review project status and make critical project decisions.

However you can define additional reports as needed. Consider what reports you need to define. Who needs to review project information? Which systems are involved? For more information on the default metrics included with Serena Dashboard, see Displaying Project Metrics with the

Serena® Dashboard [page 43]. To learn more about configuration Dashboard views, please see the *Serena Development Manager Installation and Configuration Guide*.