

---

# ALM Best Practices Series

**For ALM Practitioners**

## Project Planning and Tracking Best Practices



Document release date: July 2020

# Legal Notices

## Disclaimer

Certain versions of software and/or documents (“Material”) accessible here may contain branding from Hewlett-Packard Company (now HP Inc.) and Hewlett Packard Enterprise Company. As of September 1, 2017, the Material is now offered by Micro Focus, a separately owned and operated company. Any reference to the HP and Hewlett Packard Enterprise/HPE marks is historical in nature, and the HP and Hewlett Packard Enterprise/HPE marks are the property of their respective owners.

## Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

## Restricted Rights Legend

Contains Confidential Information. Except as specifically indicated otherwise, a valid license is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

## Copyright Notice

© Copyright 1997-2020 Micro Focus or one of its affiliates.

## Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
  - The number before the period identifies the major release number.
  - The first number after the period identifies the minor release number.
  - The second number after the period represents the minor-minor release number.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition, visit the following URL:

<https://admhelp.microfocus.com/admhelp/Content/AdmHelpCenter.htm>

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your sales representative for details.

## Support

You can visit the Software support web site at:

<https://softwaresupport.softwaregrp.com/>

This web site provides contact information and details about the products, services, and support that Micro Focus offers.

Micro Focus online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up Micro Focus support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as a Passport user and sign in. Many also require an active support contract. To find more information about support access levels, go to the following URL:

<https://softwaresupport.softwaregrp.com/doc/KM000005>

To register for a Passport ID, go to the following URL:

<https://softwaresupport.softwaregrp.com/>

# Contents

About Project Planning and Tracking .....	9
Audience.....	10
Prerequisites .....	10
Terminology .....	11
Structure .....	12
Feedback .....	12
<b>1 Introduction to Project Planning and Tracking .....</b>	<b>13</b>
Planning a Release.....	13
Designing Organizational Policy into a Release.....	14
Monitoring Release Performance vs. Plan.....	15
<b>2 Using a Release.....</b>	<b>17</b>
Defining the Right Release .....	17
Cycles and Phases.....	18
Tracking and Mitigation .....	19
<b>3 Using Scope Items .....</b>	<b>20</b>
Defining the Right Scope Item .....	20
Setting Scope Item Content .....	21
<b>4 Using Milestones .....</b>	<b>23</b>
Milestone Types.....	24
Defining the Right Milestone.....	25
Milestone Tracking Period.....	25
Milestone Scope.....	26

Milestone KPIs .....	26
Configuring the KPI Set .....	27
Refining Measured Scope per KPI .....	27
Tracking Milestone Status .....	28
Changing Milestone Scope and KPIs .....	28
Rescheduling Milestones .....	29
<b>5 Using KPIs .....</b>	<b>30</b>
Understanding and Configuring KPI Measured Values .....	31
On What Population is a KPI Measured? .....	31
KPI Calculation Definition .....	32
KPIs Value Calculation Process .....	32
Tracking KPIs by Cycle .....	33
Predefined KPIs .....	34
Requirement Management .....	34
Status Interpretation .....	34
KPIs .....	35
Test Authoring Process .....	35
Status Interpretation .....	35
KPIs .....	35
Test Execution .....	36
Status Interpretation .....	36
KPIs .....	36
Defect Tracking .....	38
Status Interpretation .....	38
KPIs .....	39
<b>6 Using Thresholds .....</b>	<b>40</b>
Objectives over Time .....	40
Defining the Right Thresholds .....	41
<b>7 Release Analysis .....</b>	<b>43</b>
Release Scorecard .....	43
KPI Analysis Graphs .....	45

Publishing Scorecards and KPI Graphs.....	46
<b>8 Release Maintenance .....</b>	<b>47</b>
Scope Items and Tracking.....	47
Rescheduling .....	48
Scope Changes.....	51
KPI Threshold Changes .....	51
<b>9 Using and Customizing KPI Types .....</b>	<b>52</b>
KPI Calculation.....	52
Default KPI Goal.....	53
Updating KPI Types.....	54
KPI Type Analysis.....	54
<b>10 Roles and Permissions .....</b>	<b>56</b>
Release Policy Manager.....	56
Release Manager.....	56
Project Manager .....	56
Team Leader .....	57
<b>11 Administrative Aspects.....</b>	<b>58</b>
Scheduling.....	58
Daily Calculation Start Time.....	58
Limiting Calculation Duration.....	58
Calculation Recurrence .....	59
Manual Activation per Project .....	59
Disabling Automatic PPT Calculations for a Project .....	60
Purging.....	60
Disabling Project Planning and Tracking for the Entire Site .....	60
Capacity Planning.....	61
Troubleshooting.....	61

PPT Event Logging ..... 61

PPT Technical Logging ..... 62

Release Calculation Failures – Troubleshooting use case..... 62

12Conclusions .....63





# Welcome To This Guide

Welcome to the Project Planning and Tracking Best Practices guide.

This guide provides concepts, guidelines, and practical examples for the best implementation of managing and tracking releases in various organizations. Following these practices promotes effective releases in your organization.

This guide applies to ALM 11.00 and later.

## About Project Planning and Tracking

For most organizations, application lifecycle management (ALM) “starts with requirements and ends when you chuck it over the wall to operations,” said Gartner analyst Thomas Murphy. “When you think of release management in the enterprise, typically there’s this whole other world out there that developers in general don’t think of, and hasn’t been associated with ALM.”

*Project Planning and Tracking (PPT)* supports the organization’s effort to deliver timely and high quality releases to its customers. The top factors contributing to high release performance are:

- Planning release scope and objectives and their expected progress throughout its duration
- Monitoring release quality, progress, and productivity with respect to the objectives
- Up-to-date release status visibility to stakeholders
- Mitigating risks and adjusting objectives based on status and on changing circumstances.

Application Lifecycle Management addresses these needs in a comprehensive manner. It facilitates all types of application tasks, using a solid foundation for managing complex initiatives and regulatory requirements. ALM is designed to address the needs of organizations that are looking to track and manage projects of all sizes, including large initiatives and enterprise-wide releases.

The purpose of this document is to assist ALM customers to assess their current SDLC practices and successfully build and maintain a quality methodology using advanced features provided by ALM. All aspects of this

process have been researched using best practice data and expertise from various sources including Micro Focus' operating system administrators, Micro Focus' professional services organization, technical documentation, books from industry experts and personal experience of many customer quality organizations. These guidelines will help reduce in initial creation time and achieve maximum value in operating the ALM.

## Audience

This guide is intended for:

- Release Managers - define release objectives and tracks release performance
- QA Managers – set QA objectives, responsible for QA execution
- Development Managers – set development objectives, responsible for development
- Organization Executives – track release progress, may be involved in setting objectives, usually a scope
- Project Managers – manage the overall project including the release

## Prerequisites

To use this book, you should have a good acquaintance with major phases of Software Development Life Cycle (SDLC). You should also be familiar with the business processes in actual IT organizations.

Operational knowledge and administrative privileges of ALM are essential in implementing these best practices. It is strongly recommended to read the *Releases and Cycles* and *Project Planning and Tracking (PPT) Releases* chapters of *Application Lifecycle Management User Guide* to get a general understanding of release, cycles, KPIs and milestones features mentioned in this document. To understand administrative aspects of this feature, it is advised to also read the *PPT Calculations* and *Customizing Project Planning and Tracking (PPT) KPIs* chapters of *Application Lifecycle Management Administrator Guide*.

Note: All features discussed in this document are available only in Application Lifecycle Management. In Quality Center Starter Edition and Quality Center Enterprise Edition these features are disabled.

## Terminology

**Organization** – when used throughout this document refers to an organization that manages a release using *ALM Project Planning and Tracking*. This may be a company's IT department, R&D department, QA department, or manufacturing department etc., or any sub-organization of these.

**Release** – a well-defined deliverable scheduled to be released for general use on a specific date. The activities leading to the readiness of a release for delivery are managed and tracked to ensure on-time and on-quality delivery.

**Scope item** – a tracked portion of the release scope. A scope item may be a business requirement, feature, theme, Change Request (CR) or a backlog item.

**Milestone** – a significant point on a release timeline for which objectives are set, managed, and tracked.

**Key Performance Indicator (KPI)** – a performance measure on a release activity, used to evaluate status based on preset objectives and to analyze bottlenecks.

**Threshold** – a means for specifying release objectives. A threshold specifies an acceptable value for a KPI measurement on a specific date.



# Structure

This guide is organized as follows:

- Introduction to Project Planning and Tracking
- Using a Release
- Using Scope Items
- Using Milestones
- Using KPIs
- Using Thresholds
- Release Analysis
- Release Maintenance
- Using and Customizing KIP Types
- Roles and Permissions
- Administrative Aspects
- Conclusions

# Feedback

If you have questions, comments, or valuable best practice information you want to share, send a message to the following email address:

[\*docteam@microfocus.com\*](mailto:docteam@microfocus.com)

# 1 Introduction to Project Planning and Tracking

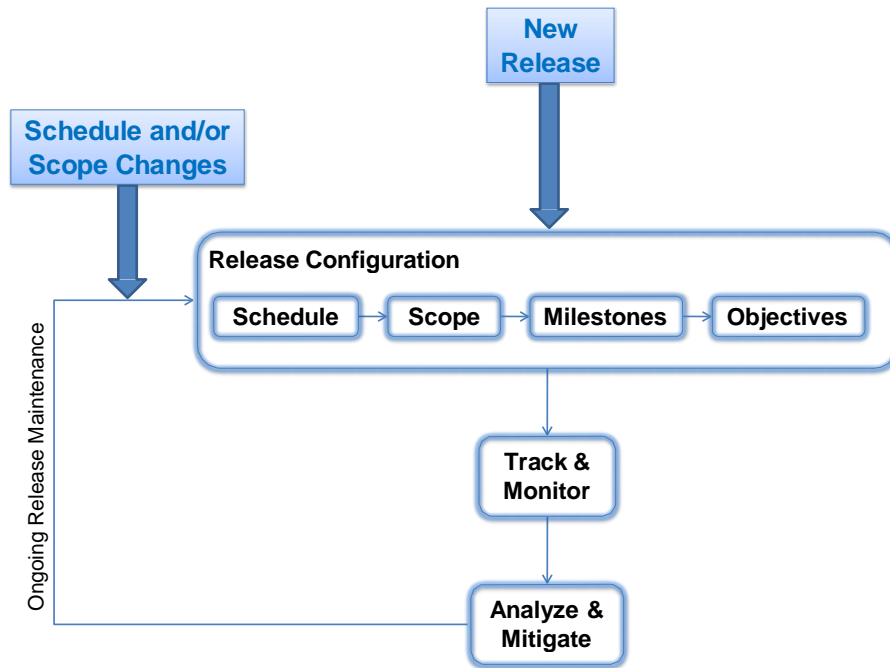
This chapter highlights the processes involved in managing a release with *Project Planning and Tracking (PPT)*. The following provides detailed best practices per the various aspects in managing PPT releases within an ALM project and within your organization. It presents the tasks involved in defining a release, including the definition of release, scope items, milestones, KPIs, and thresholds. It shows how these definitions assist in monitoring and tracking a release and enabling release progress and quality.

## Planning a Release

A well-planned and maintained release is the basis for a successful release delivery. Planning a release entails the following tasks:

- Setting release' start and end dates.
- Defining scope - features/themes/CRs (change requests) and identifying the assets that comprise it (requirements, tests, test instances and defects) by means of scope items.
- Defining milestones and associating them with scope – this determines the release activities.
- Planning objectives per milestone – by setting KPIs and thresholds.

We know that changing circumstances, such as, changes in customer requirements, changes in resources and capacity, or adjusted estimates, may impose changes on the above factors as a release progresses. Therefore, release planning is not restricted to the beginning of the release. It is rather an ongoing process of adjusting to address the changing circumstances. The diagram below illustrates this process.



**Figure 1 – PPT Release Setup and Maintenance Process**

## Designing Organizational Policy into a Release

Project Planning and Tracking allows for planning releases based on methodologies practiced in the organization. Following are release constructs that may be subject to change, according to organizational policy:

- **Duration**

For example, 3 months for a minor release, 6 months for a major release, etc.

- **Milestones**

Milestones and their placement on the release timeline define a lifecycle. For example, per cycle for an agile release, per phase for a waterfall release, a General Availability (GA) milestone to track release health, a Signoff milestone to track the signoff process between the development team and the QA team, etc.



- **Key Performance Indicators**

Key Performance Indicators (KPIs) of interest are specified on the release milestones to track performance of the activities associated with the milestone. For example, in a classic waterfall release, KPIs such as Tests Executed, Tests Passed, and Requirement Coverage would be measured on a System Tests Phase milestone in order to track system tests execution.

- **KPI thresholds**

KPI thresholds reflect practiced objectives. For example, on the GA milestone, the threshold for the Requirement Coverage KPI is set very high (95%) and for the Severe Defects KPI it is set very low (0).

## Monitoring Release Performance vs. Plan

A release is monitored on an ongoing basis throughout its duration to ensure that its quality and progress are as expected. In case of deviation of the actual release performance from the plan, the release manager can analyze the root cause and take mitigating actions as deviation occurs. For instance, the release manager may choose to allocate more resources to compensate for slippage or he/she may advocate for scope reduction, etc.

Monitoring is done using release graphs, the main one being the scorecard. A scorecard displays release KPI measurements – the scores; each score has a color (green, yellow or red) indicating its status; the status is based on the KPI's thresholds.

The figure below shows a sample scorecard. The rows are release features and the columns are KPIs. The KPIs are grouped by the milestones (release stage) on which they are measured. The circled cell, for example, represents the test execution status on the 'Online Recurring Booking Service' feature.

Mercury Tours - Release 10.5 Scorecard									
	Code Freeze					Release to Production			
	Defects Fixed per Day	Passed Requirements	Rejected Defects	Severe Defects	Tests Executed	Defects Fixed per Day	Passed Requirements	Severe Defects	Tests Executed
Booking System - New Booking Options	2	95 %	10 %	1	100 %	0	100 %	1	100 %
Online Recurring Booking Service	2	95 %	6 %	0	100 %	0	100 %	0	93 %
Reservation Extension Service	2	95 %	5 %	0	100 %	0	100 %	0	100 %
Self Service Profile Management	2	95 %	7 %	0	100 %	0	100 %	0	100 %

**Figure 2 - Release Scorecard**

## 2 Using a Release

### Defining the Right Release

A well defined release takes *schedule and scope* into account. To define the right release for your needs, start by setting release start and end dates.

A release's duration should *span over the time* that release activities are carried out and tracked within the organization.

In order to enable proper scoping of a release, all assets related to the release – requirements, tests, test instances, and defects – must reside in the same ALM project as the release.

**ALM assets related to the release - requirements, tests, test instances, and defects - must reside in the same ALM project as the release.**

In addition to timeline and scope, which are release-specific, a release definition is comprised of cycles, milestones, and KPIs. These may be driven or partially driven by organizational policy or common practice. Project Planning and Tracking allows defining these constructs in a template release, defined in a template project, as part of the Cross Project Customization feature. The template project that is the basis for your ALM project should include a release template per type of release practiced in the organization. See the KPI Type Customization chapter for information regarding KPI types and cross project customization.

A newly created release may be based on a release template or created from scratch. When possible, base a new release on a release template, rather than creating an empty release. This will ensure that the release is started with the set of cycles, milestones, and KPIs commonly practiced in the organization for the selected type of release.

**Define a release template in the ALM template project per release type practiced in your organization. Base new releases on release templates, rather than creating them from scratch.**

Changes in plans are a fact of life for release managers and sometimes a release reschedule may be required. Rescheduling a release is done by changing both or one of the start and end dates to take place sooner or later

than expected. Rescheduling a release may cause *automatic rescheduling* of its milestones and their tracking periods to ensure that they are contained within the release timeline. Refer to the Release Maintenance chapter for further information regarding rescheduling a release.

## Cycles and Phases

Cycles and phases are common practice in releases. Phases are common in the context of waterfall and V-model releases, cycles are used in many types of releases – agile and waterfall as well.

Easily map Project Planning and Tracking milestones to phases by defining the milestone at the phase end date and the phase exit criteria as milestone KPIs. You can also define the tracking period of the milestone as the duration of the phase. See the Milestones chapter for further information on the milestones to be defined when working on a phased release.

Cycles are pre-defined time slots in the release. Defect resolution and test execution may be planned by cycle by assigning defects and test instances to cycles. When cycles are defined for a release, use the Assign-to-Release and Assign-to-Cycle features on the ALM entities in order to assign the work to the cycles. This will come in handy when setting objectives. For further information on KPIs and milestones (and setting objectives), see the corresponding chapters in this document.

**Use the Assign-to-Release and Assign-to-Cycle features on the ALM entities in order to assign work to cycles.**

# Tracking and Mitigation

Release status should be monitored on a regular basis by all its stakeholders. Project Planning and Tracking provides several reports for this matter, such as the release scorecard, the KPI progress graph, and others. See the Release Analysis chapter for further information regarding reports available for tracking release progress.

The *release manager* uses the tracking tools to track release progress and identify risks that may jeopardize the delivery. He/she leverages early problem detection to resolve issues and minimize damage cost. *Development and QA managers* may use the tracking tools to track team performance, progress of team activities, and quality of the team deliverables. Various executives may use the tracking tools to obtain a high-level status of the release and its deliverables in terms of quality and progress.

**Release status should be monitored on a regular basis by all its stakeholders.**

Reports should be set from the perspective of the stakeholder for whom they are intended. Statuses and values displayed in the reports are established based on planned objectives and on delivery state. Statuses may be green (OK) yellow (warning) or red (critical). The release manager should take actions to keep occurrences of yellow and red statuses to a minimum; this is a vital factor in effective release management. The first action is by means of prevention - take into account available resources and their capabilities when planning the release. Reactive actions involve analyzing the reason for those statuses and reacting (adding resources, enhancing capacity, reducing scope, adjusting release plan, etc.). Note that false alarms – statuses that are yellow or red, but in real life are not of concern to stakeholders - lower stakeholder guards, thus causing disregard to real problems manifested in real alarms.

**Different stakeholders have different perspectives of the release. Setup reports per stakeholder perspective. Keep occurrences of yellow and red statuses to a minimum.**

For information on planning a release and its expected statuses, adjusting the plan, and on analysis and mitigation of statuses, refer to the KPIs, Milestones, and Release Analysis chapters in this document.

## 3 Using Scope Items

A basic factor in every release is the definition of the release scope. In Project Planning and Tracking releases, the scope is defined by means of scope items. A scope item represents a unit within the release deliverable(s). It may be a business requirement, feature, theme, Change Request (CR) or a backlog item that is to be delivered as part of the release.

Scope items are the units tracked in a Project Planning and Tracking release. Well-defined scope items are the basis for setting up tracking and analysis reports that will provide release stakeholders with the most value.

### Defining the Right Scope Item

The extent of a scope item may vary. One scope item may be defined as a small CR that is developed and made ready for delivery in a matter of days and another one may represent an overall feature that is developed throughout the course of an 18-month release. One may cover a requirement sub-tree containing hundreds of requirements and another may cover a much smaller number of requirements. It is up to the release manager to set up the scope items and to decide on their size.

So, what is the optimal size of a scope item?

- It should represent a unit within the release that is handled as a unit of work by all involved groups (manufacturing/development, QA, delivery, etc.). It is a release deliverable and is tracked as the release progresses.
- A release should contain up to 30 scope items in order for it to be manageable and in order to ease its tracking by the stakeholders. Note that the maximum number of scope items that can be entered in a release is limited by a site configuration parameter and may be modified by the system administrator.
- A scope item should not be too large; setting goals for its related activities, and then tracking activity execution and analyzing and mitigating activity problems, should be manageable tasks. See for information regarding scope items and tracking.

- A scope item should include all delivery aspects of the delivery unit it represents. It **should not** be split based on entity type or based on cycle or team. Properly applying KPIs to the scope items (with and without optional filters) in milestones, allows for proper tracking of its related activities.

**Considerations for establishing a scope item's scope:**

- Represent a tracked unit of delivery
- Not too large: goal setup and tracking of related activities should be manageable
- Not too small: maximum 30 per release to ease release overall tracking

## Setting Scope Item Content

A scope item consists of requirements, tests, test sets, and defects.

- **Requirements**

The basis for a scope item's content is the set of requirements that define it; these specify the actual deliverable. Managing the requirements tree based on features, allows easy definition and maintenance of the scope items.

For example, the 'Mercury Tours – Release 10.5' release in the ALM demo project includes a 'Reservation Extension Service' scope item. The requirements in this scope item are set by selecting the corresponding feature in the requirements tree. Note that in order to include only the requirements that are to be implemented in the 10.5 release, a filter is defined on top of the selection to include only requirements in which the Target Release field is 'Release 10.5'.

**Maintaining a feature-based requirements tree contributes to ease of scope item setup and scope item maintenance.**

- **Tests and Test Instances**

As in the case of requirements, tests and test instances may be included in a scope item by means of selecting sub roots and by means of filters. However, as opposed to requirements, tests and test instances may also be included in a scope item by means of linkage. One may indicate on a scope item that it includes the tests covering the scope item's

requirements. Similarly, one may indicate that a scope item includes the test instances included in the scope item's tests. It is recommended to manage test and test instance linkage in the ALM implementation and to utilize this when setting scope item content.

**It is recommended to maintain test and test instance linkage in your ALM project. Such linkage, if implemented, should be utilized when setting scope item content.**

- **Defects**

Defects are included in a scope item by means of filtering. It is recommended to use the Target Release and Target Cycle fields in your filter settings, to include defects in a scope item.

**Include defects in a scope item using the Target Release and Target Cycle fields in your filter settings.**

*Note that scope item content is usually dynamic.* For example, entities may be added to scope items because they comply with a specified filter or because they are a descendant of a selected root. This requires special care when planning progress of activities related to the scope item. See the Thresholds chapter for more details on this subject.



---

## 4 Using Milestones

Milestones are checkpoints on a release timeline used to verify that the release and its interim deliverables are progressing as planned. Some examples are code freeze, end of system and integration tests phase, beta release, and many more.

Project Planning and Tracking milestones provide the tracking platform for releases. KPIs are tracked and reported against milestones. The combination of KPI readings and statuses provides an overall picture of the release health throughout the release duration. A good setup of milestones is the basis for effective tracking and maintenance of the release. It allows stakeholders to track the relevant KPIs at all times and to respond to problems as they occur.

The figure below illustrates how milestones fit into the greater scheme of a release. Milestones are set along the release timeline, and scope items and KPIs are assigned to the milestones. The scope items specify expected milestone deliverables and the KPIs, along with their thresholds, specify objectives for these deliverables. A KPI measurement is produced for each scope item defined on the milestone and its status (green, yellow or red) is assessed based on the KPI configured thresholds.

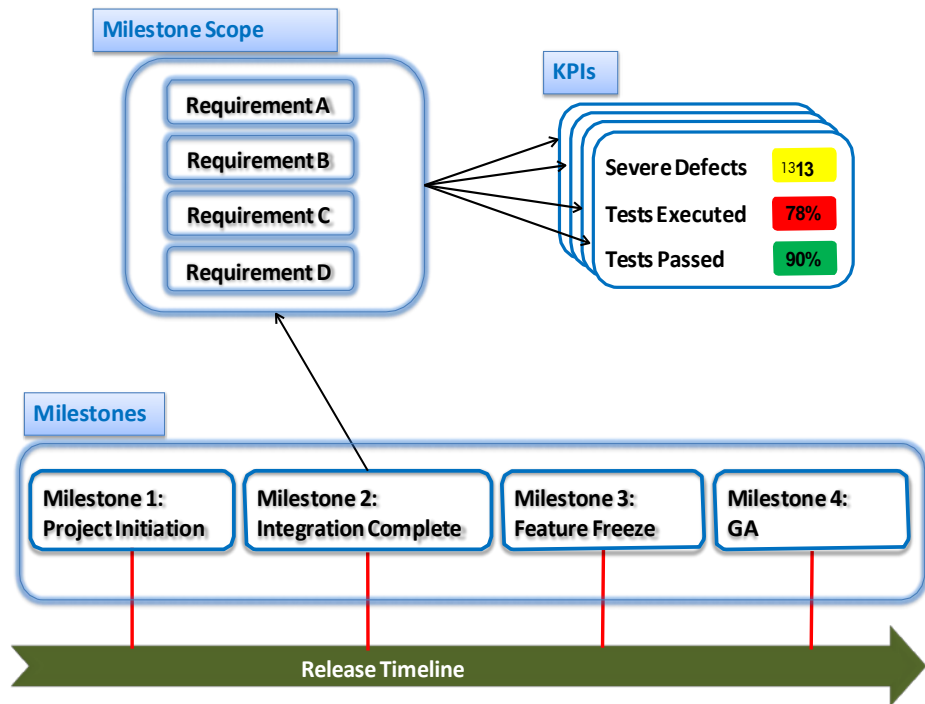


Figure 3 – The Milestone Concept

## Milestone Types

Milestones are set on strategic points on the release timeline such as Alpha, Beta and Code freeze. A milestone should be configured to provide the relevant view of the status based on the practiced methodology.

- **Release readiness milestone**

This milestone reflects the release readiness for delivery. It is tracked starting at a point before the release due date and until the due date. It tracks the status of requirement coverage, test execution, passed tests, passed requirements, and defects. The scope of this milestone includes the entire release scope.

- **Strategic milestone**

This milestone measures factors relevant to strategic events such as a Beta release. In these types of milestones, scope items may not require full completion or full coverage; they may also require different quality rating indexes than at the end of the release. Set strategic milestones along with the relevant KPIs and their thresholds based on the release methodology practiced in your organization.

- **Phase milestone**

This milestone is used to verify exit criteria of a release phase and to ensure entry criteria for the next phase. Entry criteria are a set of KPIs on the milestone that measures the completion, coverage, and quality of the previous phase(s). They are mostly used when there are dependencies between scope items, and quality parameters are critical for the next phase. The notion of phase milestones may be applied to cycles in an iterative release as well as to phases in a waterfall style release.

**Set milestones along with the relevant KPIs and their objectives (thresholds) based on the release methodology practiced in your organization.**

## Defining the Right Milestone

### Milestone Tracking Period

A milestone's due date is the date by which the milestone's scope should meet its objectives as defined by the milestone KPIs and thresholds. A milestone's start tracking date is the date from which KPIs are calculated. KPIs will continue to be calculated until the milestone due date.

A milestone's tracking period should accommodate all activities related to its scope. For example, assume an "Integration Testing" milestone, which has two scope items representing two features, and which measures test execution and test authoring progress; the milestone-tracking period should accommodate both the test authoring and the test execution activities on both features. The tracking period should include extra time to cover unforeseen adjustments that may occur if deliverables are delayed.

**A milestone's tracking period - beginning at the start tracking date and ending at the due date - should accommodate all activities related to its scope. The tracking period should also leave room for adjustments if deliverables are delayed.**

## Milestone Scope

A milestone's scope should be set to reflect its planned deliverables. For example, assume a release containing the following scope items: 'Self Service Enhancements', 'Satellite Provisioning' and 'Billing UI Enhancements'. However, the Beta release will only include the ability to perform satellite provisioning and to bill for it; it need not include self-service related activities. Therefore, we define a 'Beta' milestone that includes the 'Satellite Provisioning' and the 'Billing UI Enhancements' scope items.

**A milestone's scope should be set to reflect its planned deliverables.**

## Milestone KPIs

KPIs are metrics that allow measurement of milestone health over time. As depicted in [Figure 3](#) above, each KPI is calculated for each of the milestone's scope items. A KPI result consists of a value and a status. The value is the actual number calculated for the KPI, and the status is either green (OK), yellow (warning) or red (critical).

A milestone's set of KPIs should indicate the health of the milestone's deliverables. For example, on a Beta release milestone, set KPIs that measure test authoring, requirement coverage, test execution, requirement passage, and defects state to ensure that tests were written to cover all deliverables and that they were executed and passed (in expected volumes) and to ensure that the quality in terms of defects is satisfactory.

See the KPIs chapter in this document for further information regarding KPIs.

**A milestone's KPIs are the indicators of the health of the milestone deliverables.**

## Configuring the KPI Set

As described above, milestone KPIs are associated with milestone scope items. By default a KPI is applied to all milestone scope items, however this may be refined: KPI measurements for specific scope items may be explicitly disabled. Similarly, KPI thresholds may be configured to override the default KPI thresholds for specific scope items on the milestone.

## Refining Measured Scope per KPI

At times, it is necessary to limit the measured scope for a KPI so it applies to certain parameters of the scope item. For example, on a cycle milestone, we may want to measure only the entities related to the specific cycle. To be more specific, assume a billing enhancements feature. Feature testing spans over the May-2011 and June-2011 cycles. In order to get a proper status of the feature on the May-2011 cycle, its KPIs should apply only to requirements and tests assigned to that cycle.

When assigning a KPI to a milestone, you may specify a filter called the *Optional Filter* (using the KPI's Details window). Use the optional filter to refine KPI status to be per cycle, per work group, or per other categorization of the entities measured by the KPI.

For example, assume a release consisting of 12 cycles that contains a self-service feature. The feature is developed and tested throughout the April and May cycles. Following are the respective configuration steps:

- Define a 'Self Service Enhancements' scope item containing the self-service feature requirements.
- Define a milestone per cycle and assign the 'Self Service Enhancements' scope item to the 'April 2011' and 'May 2011' milestones.
- Add a *Test Instances Executed* KPI to the above milestones. This will be used to track execution of the test instances in these cycles.
- Set the optional filter in each KPI to include only test instances of the right cycle. The filter is defined based on the *Target Cycle* field of Test Instance. For example, the filter in the KPI on the April milestone reads:

```
Filter: Target Cycle[^Releases\Telephony\Phoenix\April
2011^].
```

Refer to the KPIs chapter for further information regarding KPIs and their usage in Project Planning and Tracking.

**Use the KPI optional filter to refine KPI status to be per cycle, per work group, or per other categorization of the entities measured by the KPI.**

## Tracking Milestone Status

Milestone status can be tracked as part of the release scorecard or using custom reports such as a scorecard focused on the milestone KPIs and/or specific KPI graphs. For further information on scorecards and reports, refer to the Release Analysis chapter in this document.

For example, assume that we have a ‘Feature Freeze’ milestone. We expect that by its due date all functional requirements are defined and reviewed and all tests are authored and mostly executed. The status of such a milestone will be the collection of these three measures.

**Stakeholders track milestone status throughout its duration to assess deliverables health and to initiate mitigating actions when required.**

## Changing Milestone Scope and KPIs

A milestone’s scope may change over time and scope may be added and/or removed from the milestone. For example:

- A preferred customer requests to add a feature to the Beta release or a certain feature is postponed to a later milestone to accommodate for resource balancing.
- Specific requests may come from stakeholders, such as development managers who wish to track their teams, or from a new executive that wants to see new types of information.

When assigning a new scope item or KPI to an **active** milestone (after its start tracking date and before it is due), calculated KPI values are produced. However, you should note that KPI historic information will be missing – these are the KPI calculations for the new KPI/scope item up to this date. This affects reports such as the KPI progress graph.

When scope items or KPIs are removed from a milestone, the calculated KPI values are removed as well. Removing a scope item or KPI and then adding it again is similar to adding a new scope item or KPI (i.e. calculated KPI values are available only from the date on which they were added).

## Rescheduling Milestones

Rescheduling of milestones may be required due to changes in the release timelines or due to changes in release focus, etc.

Rescheduling a milestone in a Project Planning and Tracking release does not maintain the thresholds curve of the milestone KPIs. When rescheduling milestones, one must *revisit* the thresholds and *adjust* as required to suit the release plan.

Note that a milestone may be *automatically rescheduled* due to a rescheduling action on a release. Refer to the Maintenance chapter in this document for further information regarding rescheduling effects.

## 5 Using KPIs

KPIs – Key Performance Indicators – are the building blocks of release tracking. Project Planning and Tracking KPIs measure release activities by means of the related ALM assets. They provide the actual measure, status evaluation over time, and breakdown measurements. Together, these enable ongoing tracking of release status and early detection of bottlenecks and problems as well as analysis and resolution means.

A good setup of KPIs will provide release stakeholders a clear picture of what is important, what the current state is, and what they need to do or change in order to reach the release goals.

KPIs may be divided into three categories based on the performance aspect they measure: progress, quality, and productivity.

- Progress KPIs

Measure how the release is doing in terms of timeline and resources (for example, number of authored tests or percentage of tests executed)

- Quality KPIs

Measure how it is doing in terms of quality (for example, percentage of passed tests or number of severe defects)

- Productivity KPIs

Measure the productivity of the team (for example, percentage of automated tests or percentage of rejected defects)

To keep all stakeholders aligned and committed to the predefined release goals, it is important to limit the number of KPIs you use to those essential for the release reaching its predefined success criteria.

**For release status clarity – use KPI selectively. Limit KPIs to those essential for the release reaching its predefined success criteria.**



# Understanding and Configuring KPI Measured Values

In order to better understand how to setup your KPIs most effectively, you need to understand how a KPI value is calculated.

KPIs are measurements of specific types of ALM entities, such as Requirements, Tests, Defects, etc. They are calculated in the context of milestones. Each milestone KPI yields a KPI value for each of the milestone's scope items. The entity type and the specific calculation are defined within the KPI Type in the ALM customization area. The population of entities to which the calculation is applied is determined by the scope item and by the configuration of the KPI on the milestone.

It is important to know, that KPI calculations include the option of storing breakdown information. For example, in a KPI counting tests one can add a breakdown by test-status or by test-designer. This breakdown will provide the number of tests per test-status or per test-designer within the KPI's population. This information enables analysis of the KPI value in order to get a better perspective on the KPI value. KPI type and breakdown definitions are part of the ALM customization area (see the 'Using and Customizing KPI Types' chapter).

## On What Population is a KPI Measured?

A KPI measurement is performed on the entities included in the scope item. The measurement considers only the entities of the type set by the KPI. For example, calculation of a 'Severe Defects' KPI will consider the defects included in the scope item.

The population may be narrowed down by means of the Optional Filter applied to a KPI on a milestone. For example, we might want to track a KPI – 'Severe Defects for the A Team' - in a Beta milestone. This may be implemented by adding a Severe Defects KPI to the milestone and by setting its Optional Filter as follows: `Filter: Responsible Team[A Team]`. Note that it assumes that the Responsible Group field exists in the Defect entity and that it is properly populated.

When you need to fine-tune a KPI's calculation (for example, use a different set of statuses, etc.), modify the KPI Type in the ALM customization area, or create a new one. This will maintain consistency of the KPI meaning when used throughout the project releases. We recommend not using the optional filter for this purpose.

Note that a KPI's optional filter applies to all milestone scope items.

## KPI Calculation Definition

A KPI's calculation is defined by the KPI Type in the ALM customization area. In general, the KPI Type specifies one or two groups of entities that are applied to the KPI population described above. These groups are used to calculate values that are either entity counts, or the sum of values in an entity field or entity percentage.

KPI Types are part of project customization and are shared between project releases; they provide for a consistent usage of KPIs within the project releases. For example, when stakeholders look at a Passed Requirements KPI value across milestones in an individual release and then across releases, it will have the same meaning in all occurrences and its value can be clearly understood in all cases.

Refer to the KPI Type Customization chapter for further information on KPI Types.

**The KPI calculation formula is defined in the KPI type. KPI types are part of organizational methodology, and shared across releases and stakeholders.**

## KPIs Value Calculation Process

The KPIs are calculated on a recurring basis as defined in Site Administration (for more information, see *Application Lifecycle Management Administrator Guide*). The calculation of each KPI involves scanning and filtering of all the ALM entities relevant to the milestones and their scope items. To make sure this process will have as little effect as possible on system performance, follow these guidelines:

- **Do not include irrelevant entities in release scope items.** In addition to impairing the calculated KPI values and blurring the statuses, this also degrades system performance.
- While it is advised that a release setup include all KPIs needed by the stakeholders, we also suggest **disabling KPIs** that are **not used** by the stakeholders. Such KPIs burden the release setup views and reports as well as system performance.

- **Remove unused KPIs from milestones and releases** – they are a burden to performance. For example, if you have started defining a release or a milestone and abandoned it, make sure that they do not include KPIs, or that the KPIs are disabled.

## Tracking KPIs by Cycle

You may need to track a KPI on a more detailed level than a scope item. For example, assume a Verification phase of a waterfall/v-model type release. To track the verification phase, we add a milestone to the release due at Verification end date and tracked throughout Verification duration, and we add the relevant scope items and KPIs to the milestone. The phase (and the milestone) may extend over multiple cycles. In order to obtain better control and enable early problem detection, we may want to track the progress of the test instance execution on a cycle basis.

When tracking milestones that extend across multiple cycles, it is recommended to track progress on a cycle basis by using the optional filter and by assigning entities to cycles using Assign-to-Release and Assign-to-Cycle features. In our example, we add a Test Instances Executed KPI per each cycle in the milestone's duration (Tests Instances Executed – January, Tests Instances Executed – February, etc.) and apply the following optional filter to each of them: `Filter: Target Cycle[^Releases\MyCar Model 2.3\January 2010 Cycle^]` (each with its corresponding cycle ID).

**When tracking milestones that extend multiple cycles, it is recommended to track progress on a cycle basis by using the optional filter and by assigning entities to cycles using Assign-to-Release and Assign-to-Cycle features.**

# Predefined KPIs

KPIs evaluate the release by setting quantifiable measurements on release activities such as:

- Requirement management
- Test authoring and approval
- Test execution
- Defect tracking

Project Planning and Tracking comes with an out-of-the-box set of predefined KPIs; these may be used (and adjusted) to measure the above activities and others. Listed below are the predefined KPIs grouped by the activities they measure; for each predefined KPI, its category is noted in parenthesis by its name.

Note that these out-of-the-box KPI types are built upon the out-of-the-box ALM. Customizing ALM entities and the fields used in these KPIs, requires the KPI types to be changed accordingly. See the KPI Types chapter for more information regarding defining and changing KPI types.

## Requirement Management

Requirement management KPIs deal with the verification process of the implementation of release requirements. The process includes two activities monitored by two respective KPIs: the signoff of the requirements by the QA organization and the development of the tests that verify the requirements.

### Status Interpretation

If measured progress of requirement review is falling beneath planned thresholds, it could mean:

- Missing or inefficient resources
- Requirements are not ready for review or not defined yet

If measured quality is falling beneath planned thresholds, it could mean:

- Test execution is lagging behind and therefore deliverable quality cannot be determined
- Deliverable is of low quality or not yet ready for testing

- Requirements are not clear

## KPIs

- **Reviewed Requirements (Progress KPI)**

This KPI measures the percentage of business or functional requirements that have been reviewed.

- **Covered Requirements (Quality KPI)**

This KPI measures the percentage of requirements covered by at least one test. Note that only functional, testing, undefined, and business-model requirement types are counted.

## Test Authoring Process

Test authoring KPIs measure the test development process. One KPI relates to the progress of the test authoring process and the other relates to the potential efficiency of the testing process itself.

### Status Interpretation

If measured progress test authoring is falling beneath planned thresholds, it could mean:

- Missing or inefficient resources for developing tests

If measured number of automated tests is falling beneath planned thresholds, it could mean:

- Test authors are not aware that they should provide for automated tests
- Test automation is not feasible and thresholds should be adjusted

## KPIs

- **Authored Tests (Progress KPI)**

This KPI measures the number of ready tests (with status Ready).

- **Automated Tests (Productivity KPI)**

This KPI measures the percentage of tests that are automated from the tests population.

## Test Execution

Test execution KPIs measure two main attributes of the release:

- Progress

Is the release converging towards executing all tests planned for execution in the tracked period (release/cycle/phase)?

- Quality

What the quality status is as reflected by the tests executed in the current release/cycle and how is it with respect to the plan?

## Status Interpretation

If measured progress is falling beneath planned thresholds, it could mean:

- Missing resources
- Inefficient resources
- Release is not ready for execution (unimplemented areas)
- Release ill-readiness for execution (implemented areas could not be tested)
- Tests are not maintained within ALM (may be performed informally)

If measured quality is falling beneath planned thresholds, it could mean:

- Deliverable is of low quality or not yet ready for testing
- Conflicts between tests and product

## KPIs

- **Tests Executed (Progress KPI)**

Percentage of executed tests (whether passed or failed) of the measured population of tests.

This KPI is most suitable when measuring the progress of the entire release, as it measures executed tests, regardless of the number of times they were executed (each test is counted once).

Notes:

- Use the KPI's optional filter in order to measure test execution on a specific cycle. Use the cross filter with Test Runs of the cycle in subject and with a Failed or Passed status. For example: Cross

```
Filter: <Cross Filter Test-Run>Status[Passed Or  
Failed];Target Cycle[^Releases\MyCar Model 2.3\January  
Cycle^].
```

- If the tests in your scope are used over multiple consecutive releases, you can still use this KPI to measure their progress. Use the same technique described above, only, rather than listing a specific cycle for the Test Runs, list all cycles included in the current release.

- **Test Instances Executed (Progress KPI)**

Percentage of executed test instances (whether passed or failed) of the measured population of test instances in the test sets.

This KPI is most suitable when measuring progress within a cycle or a short phase, since each execution of the tests (each test instance) will be considered in the KPI, thus yielding a measure that reflects the actual work done.

Notes:

- The KPI measurement is regardless of the test associated with the test instances. Therefore, when performing a test twice, it is counted twice in the KPI measurement in order to measure the progress of test instance execution with respect to the plan.
- Use the KPI's optional filter in order to measure test instance execution on a specific cycle. Simply set the target cycle on the filter. For example: Filter: Target Cycle[^Releases\MyCar Model 2.3\January Cycle^]

- **Passed Tests (Quality KPI)**

This KPI is the percentage of tests in which the last run passed, of the measured population of tests.

It provides an indication as to the quality of the release or the measured scope. Note that this measure alone is not a sufficient indicator. We may see a high percentage of successful tests also in a situation where few tests were executed. Thus, a solid track of test execution planning and test execution is a pre-requisite for a reliable quality measure.

- **Passed Requirements (Quality KPI)**

This KPI is the percentage of requirements in which all covering tests passed, of the measured population of requirements. Only functional, testing, business model, and undefined requirements are considered.

This KPI provides an indication as to the quality of the release or the measured scope as well as to the degree of completeness of the scope. As in the case of the Passed Tests KPI, this measure alone is not a sufficient

indication; a solid track of requirement coverage is a pre-requisite for a reliable quality measure.

Note that the considered requirement types are those for which test coverage is supported. If there are more such requirement types in your local ALM implementation, you should add them to the KPI type definition (both to the numerator and to the denominator). See the KPI Types chapter for best practices for updating KPI types.

## Defect Tracking

Defect tracking KPIs measure two attributes of the release:

- Quality  
Is the release or the scope of the tracked period converging towards its expected quality?
- Productivity  
Is the effort of defect detection and resolution as efficient as could be expected?

## Status Interpretation

If measured quality is falling beneath planned thresholds, it could mean:

- Deliverable is of low quality or not yet ready for testing
- Conflicts between tests and product or unclear requirements

If measured defect fixing productivity is falling beneath planned thresholds, it could mean:

- Lacking defect fixing resources or there are fewer defects than expected
- Defect fixing resources are inefficient

If measured defect rejection is higher than planned thresholds, it could mean:

- Requirements are not clear
- The QA team does not understand the product and open defects needlessly
- Defects are ill-phrased and misunderstood by their assignees



## KPIs

- **Severe Defects (Quality KPI)**

This KPI counts active defects (status is new/open/reopen) with a severity level of 5-Urgent or 4-Very High, within the measured population of defects.

It provides a good indication as to the readiness of the release or the measured scope. Note however, that in case test execution is lagging, this measure will not give a reliable indication.

- **Defects Fixed per Day (Productivity KPI)**

This KPI provides a measure of the defect-fixing rate. It counts the number of defects in which the status changed from 'open' or 'reopen' to 'fixed' in the last 24 hours..

Looking at the KPI Progress Over Time graph gives an indication of the capacity for fixing defects and may be used when planning work and when planning the expected number of remaining open defects (for example when defining the thresholds for the Severe Defects KPI).

You may measure this KPI per team by means of the optional filter, allowing for planning on the team level.

It is suggested to use this KPI in productivity-focused scorecards and exclude it from release scorecards for two reasons:

- It might display false alarms at times. Why does this happen? Since this is a daily count KPI on work done, the measured values may not be continuous. For example, calculations on non-working days will yield zero values while calculations on working days will yield perfectly healthy values. In order for the KPI status to be green on Mondays, the threshold planning needs to consider this effect.
- It does not contribute to the general understanding of release health.

- **Rejected Defects (Productivity KPI)**

This KPI is the Percentage of rejected defects of the measured population of defects.

## 6 Using Thresholds

Thresholds are the means for setting objectives on Project Planning and Tracking releases and for planning and tracking release progress towards these objectives throughout its duration.

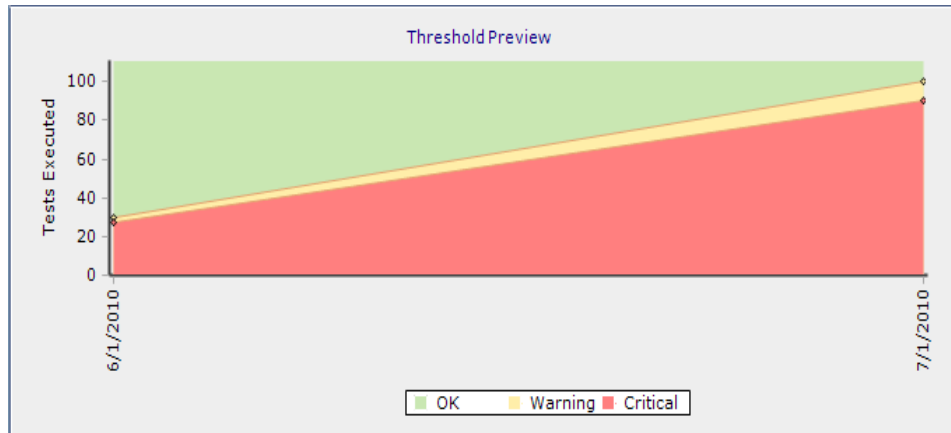
As we learned so far, a release is tracked using KPIs and their statuses. The statuses indicate how close or far the release situation is from its planned goals. A KPI's threshold defines how its measured value is evaluated to a status. It splits the range of possible measured values of a KPI into three sub ranges, thus defining the three statuses—green (OK), yellow (Warning) and red (Critical) – and the KPI's status is established based on the sub range into which its measured value falls.

For example, assume a 'Tests Instances Executed' KPI assigned to an 'Alpha' milestone. Our objective is to complete 95% of the testing by the milestone due date. Quality so far in the release has been very good and there are still a couple of cycles ahead. Therefore, we can afford slippage of 10% in the objective assuming we will catch up in the next cycle. So in our case – the 'Test Instances Executed' KPI on the 'November 2010' milestone – we set the threshold to 95% with a 10% tolerance.

### Objectives over Time

The importance of tracking over time is driven by the fact that it provides the ability to detect problems and bottlenecks early enough to allow for effective mitigation. Therefore, objectives are set throughout a milestone's duration (from its 'start-tracking-date' until its due date). These form a curve on the milestone timeline.

The figure below shows a threshold curve for a 'Tests Executed' KPI. We can see that test execution is expected to be 30% done by the time milestone tracking starts and that it should grow to 100% up to the due date. The tolerance is set to 10%. Note the colored zones on the graph. The green zone is the area above the KPI expected values, the yellow zone represents the tolerance (up to 10% below the KPI expected values) and the red zone is anything worse than (below) the tolerated values.



**Figure 4 - Thresholds Curve Example**

## Defining the Right Thresholds

A KPI's threshold curve should reflect the KPI's objective for the milestone. It should also be set so that once the milestone becomes active the KPI status is green at all times.

Several factors guide the planning of a good threshold curve:

- Due date threshold should follow practiced policy for the KPI**  
 The KPI objective is reflected in the threshold assigned to the milestone's due date. KPI objectives are usually based on practiced policies in the organization (test execution should be 100% or 95%, number of severe defects should be zero or three, etc.). See the KPIs and KPI customization chapters for best practices for KPI objectives.
- Threshold curve should be aligned with resource plan**  
 Adding and removing resources from the activity implied by a KPI affects KPI results. Intermediate thresholds set for the KPI should take into account the planned resources and their expected productivity.
- Threshold curve should consider content changes in dynamic scope items**  
 Scope item content is usually dynamic, i.e., it may change over the duration of the milestone. For example, defects might be added during test execution, thus effecting the content of defects measured by a 'Severe Defects' KPI; a requirement may be added to a scope item's requirements sub root, thus effecting the content of a 'Requirements Reviewed' KPI,

etc. When setting thresholds, one needs to consider expected changes in the KPI's content with respect to the available resources. Similarly, affected thresholds need to be adjusted when unplanned changes are applied to scope item content.

- **Threshold trend for percentage KPIs might vary**

For example, a 'Tests Executed' KPI when measuring dynamic scope items. KPI results grow as long as the content does not, but once the content grows, for example, new tests were added to the scope item, the measured execution percentage drops. Thresholds should be set to a lower percentage at the times when the content is expected to grow in order to avoid alerts in status tracking.

- **Consider activities toward the milestone**

When designing a threshold, you will need to take into account possible changes due to the phase of activities being done at the same time. For example, when first starting to design the requirements, none of the requirements will have been reviewed. As time goes by, the percentage of reviewed requirements reaches a steady state and toward the end of the release (or the milestone), 100% of the authored requirement should be reviewed. The threshold curve should reflect these dynamics.

**A KPI's threshold curve should reflect the KPI's objective for the milestone.**

## 7 Release Analysis

Tracking a release is essential for achieving release goals – it allows for early problem detection and resolution. Project Planning and Tracking provides several tracking tools. The scorecard provides release status at a glance; it displays the results for the release KPIs along with their statuses. The KPI graphs analyze the results of a single KPI.

### Release Scorecard

The scorecard view provides the release status at a glance; it displays release KPI results and statuses in a table format. We also refer to the results as scores. The scorecard enables ongoing tracking of the release status and early detection of bottlenecks and problems. It also provides analysis and resolution means using drilldown capabilities on the KPIs. See [below](#) for further information on KPI analysis graphs.

The figure below shows a sample scorecard. Each row is a release feature and each column is a KPI. The KPIs are grouped by the milestones on which they are measured. The circled cell, for example, represents the test execution status on the 'Online Recurring Booking Service' feature.

Mercury Tours - Release 10.5 Scorecard									
	Code Freeze					Release to Production			
	Defects Fixed per Day	Passed Requirements	Rejected Defects	Severe Defects	Tests Executed	Defects Fixed per Day	Passed Requirements	Severe Defects	Tests Executed
Booking System - New Booking Options	2	95 %	10 %	1	100 %	0	100 %	1	100 %
Online Recurring Booking Service	2	95 %	6 %	0	100 %	0	100 %	0	93 %
Reservation Extension Service	2	95 %	5 %	0	100 %	0	100 %	0	100 %
Self Service Profile Management	2	95 %	7 %	0	100 %	0	100 %	0	100 %

**Figure 5 - Release Scorecard**

Project Planning and Tracking measures KPIs in the context of milestones and their scope. Upon every round of KPI calculations (which is done once or twice a day), one KPI measurement is produced per milestone scope item. A scorecard cell represents a single KPI measurement. In addition to the KPI measurement, a scorecard cell displays the KPI status. The status is

determined based on the KPI objective and on its measurement. Statuses may be green (OK), yellow (warning) or red (critical).

For example, if the status of the ‘Severe Defects’ KPI for the ‘Self Service Enhancements’ feature in the ‘Beta’ milestone is critical (red), this means that the amount of critical defects is higher than the objective defined for defects at this time. This means that the release manager should take action in order to get this milestone back to an OK status (green). He/she can do this by allocating more resources for bug fixing, by postponing the milestone due date and adjusting objectives, by reducing milestone scope, etc.

When a KPI cell is yellow (warning) the release manager should evaluate whether this indicates a problem or whether this is an expected deviation. For example, deviations are expected when the content of a scope item is still growing and goals cannot be accurately predicted. The KPI over Time graph may assist in determining whether a warning state is tending towards a critical state or whether it is pretty stable and tending towards a resolution.

Project Planning and Tracking provides a scorecard view in the *Scorecard* tab of the *Management* module. This scorecard provides the most recent status of the release KPIs and is common to all *Release* module users.

ALM *Analysis* module allows defining release scorecards as analysis items, configuring them, and making them widely available. These scorecards may be shared more widely and may be configured to focus on specific KPIs and/or on the status as per a specific date. See [Publishing Scorecards and KPI Graphs](#) section for further information regarding the usage of scorecards and KPI analysis graphs as analysis items.

# KPI Analysis Graphs

Project Planning and Tracking provides KPI analysis graphs used to analyze the root cause of a problem manifested in a KPI status.

Project Planning and Tracking provides the following graph types for tracking and analyzing specific KPIs:

- **KPI Over Time graph**

Provides a view of a KPI's trend by displaying the historic results and statuses for the KPI. The user can see if there is improvement/deterioration over time.

- **KPI Breakdown graph**

Shows breakdowns of the KPI results. The precise breakdown types available for a KPI type are defined in the Project Planning and Tracking customization and are calculated along with the KPI. Breakdowns are used to better analyze KPI results.

For example, the 'Passed Tests' KPI for the 'Self Service Enhancements' feature in the 'Beta' milestone shows a high value and is colored in red. Its breakdown graph - 'Tests by Execution Status' – may indicate whether our problem is with the progress in test execution or with the quality of the tested deliverable. In the first case, the release manager might choose to add more testing resources.

- **KPI Breakdown Over Time graph**

Shows the history of the KPI breakdown results. Looking at the previous example, in case the breakdown by execution status indicates that the issue is in the product quality, the release manager may check the breakdown over time to see if this issue persists and then escalate the issue to higher management, or if it is local and then escalate only to the development manager.

These graphs are available as drilldowns from a scorecard cell and may be defined as analysis items in the Analysis Module as well.

KPI graphs as analysis items example: The 'Passed Tests' KPI for the 'Self Service Enhancements' feature in the 'Beta' milestone is in critical status (red). We want to monitor this KPI until the KPI status turns back to OK (green). In order to do so we create three analysis items configured to view this KPI – a KPI over Time graph, a KPI Breakdown graph, and KPI Breakdown Over Time graph. We also create a dashboard page, 'Beta Passed Tests KPI' and add the new analysis items to it. We now use the new dashboard page to track the KPI status and its behavior and to determine if

action is required, until it returns to a green state indicating that the testing activity is back on track.

KPI breakdowns and their calculation are defined in the ALM customization area.

## Publishing Scorecards and KPI Graphs

Publish release scorecards and KPI graphs in order to share them with release stakeholders; compose an analysis item for each item to be shared. As any ALM analysis item, these can now be published either by referring the stakeholders to the analysis item in the *Analysis* module, or by embedding one or more analysis items in a dashboard page and referring the stakeholders to the page in the *Dashboard* module. Another way to publish an analysis item is via a URL reference to it; the URL is obtained using the analysis item's *Share Analysis Item* option. The URL may then be sent via email or embedded into a web portal page.

Following are some examples of published scorecards and KPI graphs:

- 'Freeze Scores' scorecard

This is a scorecard presenting the release scores as per the code-freeze date. This analysis item may be published in the Web portal with comments regarding the scores and it may be used for future references.

- Team scorecard

This is a scorecard presenting KPIs for the scope items relevant for a specific team. This is done by configuring the scorecard's scope item filter to include the relevant scope items.

**Publish release scorecards and KPI graphs in order to share them with release stakeholders.**



## 8 Release Maintenance

A Release is a *live entity, subject to ongoing change* - the scope of its deliverables may change, schedules may change and objectives may change. Changes may be the result of external constraints such as customer demands for extended deliverables or competition forcing changes in release due dates, and of internal constraints such as unexpected changes in available resources or changes in effort estimates.

Project Planning and Tracking provides means to perform the continuous task of release maintenance throughout release lifetime.

### Scope Items and Tracking

*Special care* should be taken in planning activity progress when scope item content is implied by selected *sub trees* or *by related entities*. In this case, the scope item content is dynamic; it changes as requirements are added to the selected requirement sub trees, and as tests are added to cover these requirements or as they are added to selected test sub trees etc.

The actual release activities are implied by adding KPIs and assigning scope items to milestones. This means that adding content to a scope item affects the extent of activities based on that content. For example, we assigned the 'Reservation Extension Service' scope item to the 'Web Portal Integration' milestone and added the Test Executed KPI to the milestone. This KPI provides the percentage of tests already executed from the scope item's tests population. The KPI result will vary as tests are added to the scope item content, regardless of the work being done. Therefore the plan for the KPI goal progress should take into account the expected growth in content population throughout the milestone duration, as well as the expected rate of test execution. Alternatively, a release manager may update the plan as new tests are added.

The figure below shows a scorecard in which the scope items are tracked using KPIs on milestones. A detailed explanation of scorecards and other reports is provided in the Tracking Tools chapter.

Mercury Tours - Release 10.5 Scorecard							
	Web Portal Integration			Release to Production			
	Active Defects	Passed Tests	Tests Executed	Defects Fixed per Day	Passed Requirements	Severe Defects	Tests Executed
Booking System - New Booking Options				0	100 %	1	100 %
Online Recurring Booking Service	9	95 %	100 %	0	100 %	0	93 %
Reservation Extension Service	9	98 %	100 %	0	100 %	0	100 %
Self Service Profile Management	9	100 %	100 %	0	100 %	0	100 %

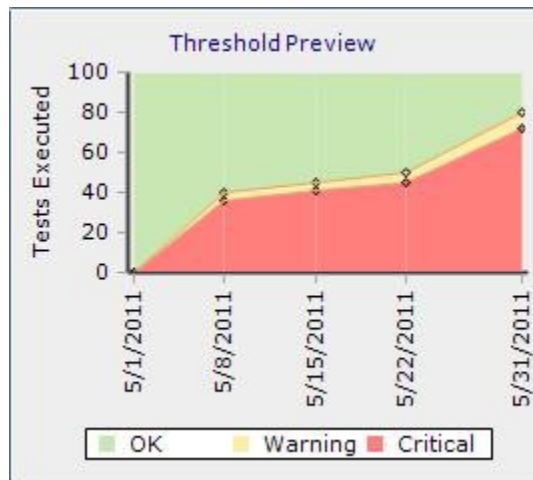
**Figure 6 - Release Scorecard**

## Rescheduling

Various circumstances may trigger rescheduling in a release. Rescheduling can be on the release as a whole and on individual milestones.

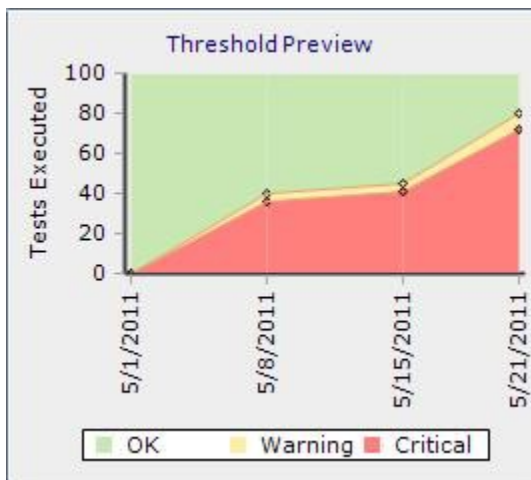
Rescheduling a release maintains the release milestones though it may shift them back or forth on the timeline to fit the new schedule; it may also shorten them to fit the new release duration in case it was shortened.

Rescheduling a milestone, whether manually or automatically (as a result of release rescheduling), affects the threshold curves of its KPIs. In general, the curves are shifted to suit the changed schedules, however if the duration was shortened, then later thresholds on the curve might be lost. Note that whether milestone duration is shortened, extended or unchanged, its KPI goals are maintained, i.e., the threshold value set for its due date remains as is. Similarly, the threshold value defined for its start date is maintained. The pictures below demonstrate changes in a threshold curve as a result of shortening and extending milestone duration. Next figure displays the original curve.

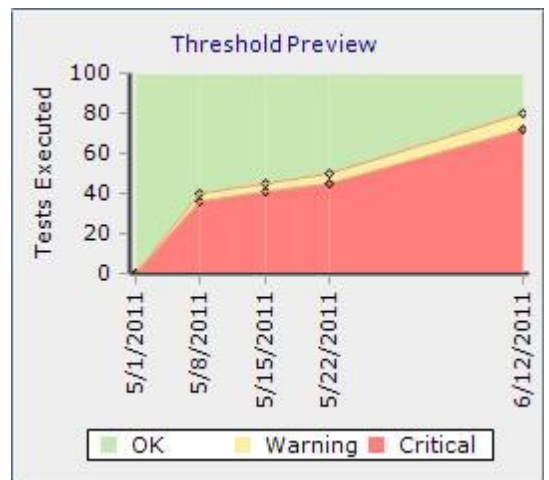


**Figure 7 - Original Threshold Curve**

The next two graphs show the curve after shortening the milestone duration, and the curve after extending the duration.



**Figure 8 - Shortened Threshold Curve**



**Figure 9 - Extended Threshold Curve**

Note that threshold value changes may yield KPI status changes. I.e., a KPI value for a specific date may have had a red status before the shift and a green status after the shift, although the KPI value itself remains unchanged.

Also, note that KPI calculation occurs only for active milestones, i.e., milestones for which today's date is between their start tracking date and

due date. Therefore, in case an active milestone became inactive or vice versa, KPI calculation will be effected.

**After rescheduling a release, you should verify and adjust milestone due dates and tracking periods to match the intended plan for the release. You should also verify the threshold curves of KPIs on milestones shortened to accommodate the new schedules.**

It should be noted that release changes are **not** reversible. For example, when you shorten a release you may lose threshold values you defined. Making the release longer again will not bring those threshold values back.

## Scope Changes

A scope change may be done by adding or removing a feature/ theme/ CR/ etc. to or from release deliverables. It may also be done by modifying the content of a delivered item. Such changes may be applied to the main release deliverables and they may be applied to interim deliverables.

Project Planning and Tracking release scope is changed by adding or removing scope items from the release, by adding or detaching scope item assignments from milestones, or by modifying the content within a scope item.

**Shift scope between cycles by updating scope item content filters or by updating KPI optional filters to refer to the new cycle.**

Scope changes affect KPI calculation and reporting. Removing a scope item from a milestone eliminates the KPI; it will no longer be calculated and it will no longer be reported in scorecards or be available for selection as the subject of a KPI graph. Changing scope item content will create a change in KPI readings, and might create a 'bend' in the KPI Over-Time graphs.

It may be advisable to make a note in the scope item description when you make major changes to it. This will help understand sudden changes in the KPIs related to this scope item.

## KPI Threshold Changes

Changes in thresholds affect the release reports (scorecard or KPI graphs) regardless of the time at which the KPI calculation occurred. This is because the KPI status is evaluated upon generation of the reports and not as part of the daily KPI calculation process.

**Release stakeholders should be notified to expect status changes in their reports when such thresholds changes are performed.**

## 9 Using and Customizing KPI Types

KPI types specify the measurement aspects of Project Planning and Tracking KPIs. The set of KPI types determines the KPIs used to track and manage releases in the organization and the means for calculating them. KPI types also define the basic KPI analysis tools. These are available to release stakeholders along with the KPI results as part of the release tracking tools.

KPI types are managed in *ALM Project Customization* in the Project Planning and Tracking category and are part of the *ALM Shared Customization* feature. A well-defined set of KPI types, along with other customization aspects such as *Project Entities* and *User Defined Fields*, allow for standardization of release configuration and tracking across the organization. They also allow release stakeholders to be aligned and committed to predefined release goals.

### KPI Calculation

Project Planning and Tracking KPIs measure release activities by means of the related ALM assets. A KPI calculation is applied to a group of entities, determined by the scope item assigned to the release milestone, to which the KPI was added. A KPI may measure the number (count) of a specific subset of entities within this group, or the percentage of one subset of another subset. KPI types specify how KPI results are produced.

A KPI type specifies the following measurement elements:

- Entity type  
Includes requirements, tests, test instances or defects. This provides a first cut on the scope item's population.
- Measurement type  
Usually count or percentage.
- The subset of measured entities  
In case of a percentage measurement, it specifies two subsets of measured entities. The measured entities subsets are specified by means of a filter, and if the filter is not specified, then the subset will consist of all entities of the designated type in the scope item. When

defining a percentage KPI type, be sure to specify the *numerator entity set*, so that it is a subset of the denominator entity set.

A KPI's calculation may be further refined in two ways:

- Rather than merely counting the entities in a subset, the calculation may sum up the values of one of the entity fields. For example, we may want to define a 'Severe Defects Effort Estimate' KPI type. This will be defined similarly to the 'Severe Defects' KPI type, only that rather than counting the specified defects it will sum up the 'Estimated Fix Time' fields in them. Note that only numeric fields may be used for this summing.
- Rather than considering all entities specified in a subset, it may consider only those in which a certain field has changed from one specified value to another. For example, the 'Defects Fixed per Day' KPI type defines that only the defects in which the status changed from 'Open' or 'Reopen' to 'Fixed', within the last day, will be counted. Note that such a refinement may be applied only to entity fields that support history and that use a lookup list.

## Default KPI Goal

The KPI type establishes the default for the KPI goal. When adding a KPI to a milestone, this value is set as the KPI's threshold for the milestone's due date (and start tracking date). The release manager configuring a release may choose to override the goal and to set it to vary according to the duration of the milestone.

**Set the Default Threshold and Warning Range of a KPI Type to the goal and tolerance for this KPI type.**

The KPI Type also establishes the expected trend for the KPI behavior – whether the situation is better as the KPI grows, or vice versa. For example, in case of the 'Severe Defects' KPI, the lower the KPI result, the better the status; the 'Tests Executed' KPI is the other way around – the higher the result, the better the state is.

## Updating KPI Types

Changes in KPI types affect all on-going releases in the project; in case of KPI changes in a template project, all stake holders of all releases in all derived projects are affected. Changing KPI types change the manner in which their related KPIs should be considered by release stakeholders. For example, if the 'Severe Defects' KPI type has been changed to count 'medium' defects as well as 'high' and 'very high' defects, then the stakeholders can expect an increase in the resulting KPI values. They might be seeing red statuses in their scorecards, and will probably need to adjust thresholds accordingly. Release stakeholders might also see sudden changes in the trend of the KPI-Over-Time graphs.

**KPI types should be updated only to reflect true policy changes that should apply immediately. Changes in KPI types should be communicated in advance to affected release stakeholders along with a detailed description of implications and guidelines as to the required adjustments to release configuration.**

When exploring the options, one should consider whether the need is for a new, similar KPI type, or whether the existing KPI type must be changed. For example, the need for changing a certain KPI type could come from users of an ALM project implemented with additional entity fields etc. thus requiring different KPI type filters. In such a case, it may be advised to hold distinct KPI types for the different ALM projects.

## KPI Type Analysis

In addition to the KPI type specification, *Project Planning and Tracking* provides a graph based analysis tool – KPI breakdown - for a better understanding of the KPI results. It provides additional data relevant to the calculated KPI values. KPI breakdowns are calculated along with their KPI, and are available for tracking by means of drilling down from a scorecard cell and as independent analysis items.

A breakdown filter is based on the same entity type as its owner KPI Type. Note that the filter of this breakdown is independent from the one defined in the KPI Type.

For example, the *Tests Instances Executed* KPI type has a *Not Run Test Instances by Responsible Tester* breakdown. It shows all tests instances that were not executed yet, grouped by the responsible tester. In case the KPI



result is red, we can check whether one or more testers are creating an apparent bottleneck.

Calculated breakdown results are stored over time and breakdown trends may be viewed using KPI breakdown Over Time graphs. These help the release/team manager determine whether issues are local or common behavior.

Be cautious when defining breakdowns. They are calculated along with their owner KPIs, and degrade overall system performance. Unused breakdowns should be disabled.

---

# 10 Roles and Permissions

Project Planning and Tracking addresses the needs of various role players in the lifecycle of a release. ALM permission mechanism provides the tools to define the tasks that may be performed by each role.

## Release Policy Manager

This person is responsible for setting release policies to be implemented in the organization's releases. His/her interaction with the system is mainly within the customization area of ALM template projects. He/she defines release templates and KPI types used throughout the organization.

The release policy manager needs permissions to manage all Project Planning and Tracking entities and to customize them.

## Release Manager

This person defines the release dates, major milestones, and deliverables, and sets desired goals for the deliverables. Deliverables are defined by setting scope items for the release. Goals are defined by configuring the KPIs on the milestones.

The release manager needs permissions to manage all Project Planning and Tracking entities.

## Project Manager

This person may add specific milestones, assign their scope (from the release general scope), and set desired goals for deliverables. He/she may refine threshold curves to suit resource availability.

The project manager needs permissions to manage milestones, milestone scope items, KPIs, and thresholds.

## Team Leader

This person is responsible for setting up a plan that will allow his team to meet the goals set by the project manager. The plan is reflected in the threshold curves throughout the milestone duration. For example, let's look at a *Tests Executed* KPI. The threshold curve for this KPI should reflect the expected progress of test execution by the team.

The team leader needs permissions to manage thresholds.

# 11 Administrative Aspects

Project Planning and Tracking is a site-wide activity with several management aspects.

The topics covered in this chapter are:

- Scheduling
- Purging
- Disabling Project Planning and Tracking for the Entire Site
- Capacity Planning
- Troubleshooting

## Scheduling

Project Planning and Tracking calculations are executed on a daily basis according to the site scheduling definitions. Let's review them all.

### Daily Calculation Start Time

The calculations are executed during the site's off-peak hours, in order to minimize its effect on online users. The selected start time does not guarantee the actual start time of the calculations. The actual calculation execution depends on available system resources and PPT configurations.

If the application server is unavailable at the scheduled start time, PPT calculations are performed after the server has started, but before the time limit has passed.

### Limiting Calculation Duration

PPT calculation time can be limited by an abort time.

Limiting the calculation duration does not guarantee that the entire site calculations are completed by the selected end time. At the abort time, the

system completes the active release calculations. Release calculations that have yet to start are considered failed.

The limit should be applied only if PPT proves to deteriorate online user activity. By limiting the calculation duration, you ensure that online activity is not affected (possibly at the expense of completing the release calculations for the entire site).

## Calculation Recurrence

PPT calculations can be performed every 24 or 12 hours.

When deciding on the calculation recurrence, the following factors should be considered:

- Business demands

Calculations that are scheduled to take place twice a day provide more up to date reports.

- Database strength and load

Performing PPT calculations add to the database workload, which in turn might affect online activity according to the database specifications and usage.

- Number of active milestones and KPIs

This is the actual amount of work that is performed in a calculation cycle. The more active milestones and KPIs, the more time it takes for the calculation to complete.

## Manual Activation per Project

PPT calculations can be triggered manually on a specific project by clicking on the “Run Now” button in the project planning and tracking section under the project details tab.

Manual activation of PPT calculations for a project can be used to refresh its releases’ results without waiting for the site scheduling, and should be used according to business demands.

Clicking the “Run Now” button does not guarantee that a project is calculated immediately. The project is added to the existing list of PPT release calculations that are pending execution and is calculated according to system resources availability and PPT configuration settings.

## Disabling Automatic PPT Calculations for a Project

Automatic PPT calculations can be disabled for specific projects by unchecking “Automatic Calculation State” in the project planning and tracking section under the project details tab.

Disabling PPT calculations for a project is considered if PPT calculations for the site do not complete successfully and the analysis of the problem shows that the problem is related to a specific project. For example, if the calculations for a specific project *always* end after exceeding the PPT abort time, then there might be a problem with that project.

## Purging

PPT calculations accumulate a lot of data – KPI calculation instances, KPI results, and breakdown results. In order to control the database size, historic PPT results can be purged by configuring the purging scheme.

Data is purged from PPT tables in the site schema and from PPT tables in ALM projects. **Purged data cannot be restored** and cannot be presented in release scorecards.

There is a built-in exception to the purging scheme – the calculation results of the last 5 days of a milestone (the last 5 days up to its due date) are never purged automatically, in order to retain this important data for analysis purposes, such as release post mortems.

The purging scheme is decided according to the organization’s business needs, records retention policy, and database size constraints.

## Disabling Project Planning and Tracking for the Entire Site

Disabling PPT on the site stops all PPT activities in all application server nodes immediately, except for active database requests, such as SQLs, running when PPT is disabled.

This option should be used only in cases where you suspect that PPT is the source of the performance related problems in the system. This way you can confirm or rule out your suspicion.

## Capacity Planning

PPT database activity can be fine-tuned with the PPT advanced settings in the PPT tab.

- “Number of engines” controls PPT concurrency level
- Engines throttle controls the amount of work each engine performs in a unit of time

Monitoring and analysis of ALM’s database activity can provide information on the workload incurred by PPT calculations. If PPT calculations have low impact on the database, PPT configuration can be changed to perform with a higher level of concurrency, resulting in faster calculations and allowing more PPT calculation cycles per day. On the other hand, PPT can be configured to a lower level of concurrency to reduce its impact on the ALM database.

## Troubleshooting

There are several ways to troubleshoot the potential problem in PPT execution.

### PPT Event Logging

PPT events are written to an event log file on each application server. The log file is configured in the servers tab in the site administration.

The events written to the log file describe major PPT events in a release context, mainly:

- The release calculation start time and number of KPIs to be calculated
- The release calculation end time, calculation duration, and the number of KPIs calculated successfully from the entire release KPIs.

The events log file should be used to track PPT calculations completeness. The event log file can be used for performance tuning and capacity planning by monitoring the number of KPIs calculated for specific releases and the time it takes to calculate them.

## PPT Technical Logging

The PPT technical log is integrated with the site administration technical log. This type of log provides the execution flow of the PPT calculations and exceptions thrown during calculations. It can help with identifying system wide errors that prevent successful completion of PPT calculations, e.g. database connectivity problems when it is down.

## Release Calculation Failures – Troubleshooting use case

When a release fails to calculate, it is indicated in the PPT event log.

A failed release calculation is usually retried up to 10 times, unless the PPT calculation time limit is exceeded. Each of the failed calculations appears in the event log. The site administrator cross-references the PPT event log with the site administration technical log according to time stamps. The exceptional condition that caused the release calculation to fail appears there. If the error condition was a system-wide error, such as the database was down, then recalculating is performed according to business needs (wait for next calculation cycle / manually calculate specific projects / change site scheduling so entire site gets calculated). If the error condition is specific to a release, e.g. erroneous KPI definition, then the site administrator informs the release stakeholders with regards to the error details.



---

## 12 Conclusions

Software plays a dominant role in today's business, regardless of the vertical market or core competency. Every organization must be able to guarantee high quality, working software to properly position and deliver its products to the market. Software does not run the business, working software runs the business. Now more than ever, software is a critical component of winning the competition. To meet the expectations of customers and shareholders alike, businesses must be able to deliver high-quality, working software. As software transforms the world of business, all companies creating software must transform their own ability to create and manage software.

A well-planned and maintained release is the basis for a successful software delivery. Changing circumstances, such as customer requirements modifications, changes in resources and capacity, or adjusted estimates, affect release progress. Therefore, release planning is not restricted to the beginning of the release. It is rather an ongoing process of adjusting to address the changing circumstances. *Project Planning and Tracking (PPT)* provides the means to increase time-to-market and beef up quality of products. Capabilities include defining and monitoring project milestones, dynamic health assessment, and automatic project updates with respect to milestones. With PPT in place, you ensure your software is delivered on time and on quality while monitoring the progress at any point of time.