

---

# OpenText™ ALM

## Business Process Models Best Practices

**For ALM Practitioners**

## Legal Notices

© Copyright 2023 Open Text.

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

### 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.

## Contents

Preface .....	6
Analyzing the model information .....	7
Model entities as requirements in the ALM Requirements module .....	8
Creating coverage for models—various use cases .....	9
Re-Import Models— How does it work? .....	16
Status view display—presenting accurate results .....	20
Frequently asked questions .....	21
Glossary .....	22

## Preface

The purpose of this guide is to help you build relationships between ALM test requirements and business process model entities.

Examine the following example of a business process model. It was created using MetaStorm Provision:

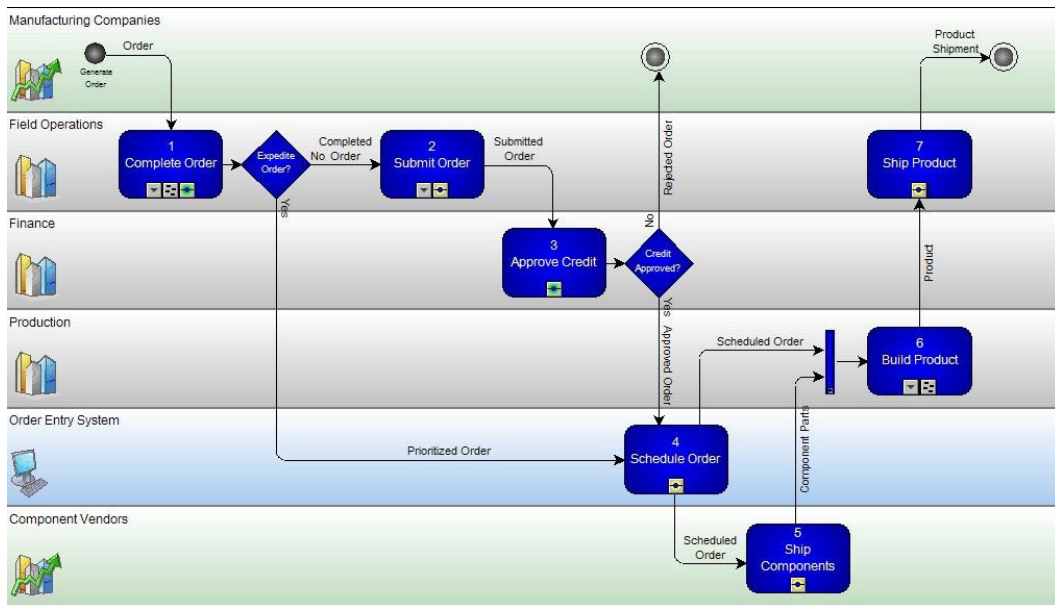


Figure: Sample business process model. The diagram was captured from the Provision tool.

This document will use this example to explain the recommended relationships to apply when importing models into Application Lifecycle Management (ALM) and connecting them to requirements, which will later form the basis for development and test planning.

## Analyzing the model information

The provided model includes additional information that extends beyond testing. When focusing on testing, the following information is useful:

1. **'Activity'** – Once this entity is imported into ALM, it can serve as the base upon which additional functional specifications are added and the functional test is built.
2. **'Path'** – this entity is not a formal entity in the modeling tool, but is introduced in ALM. ALM automatically analyzes the model and offers all the possible end-to-end paths. These paths can then serve as high level 'logical business processes' which need to be tested end-to-end.
3. **'Model'** – this can serve as a base for additional functional requirements, non-functional requirements and general test requirements (such as performance, security, usability, scalability...)

In this sample there are 7 activities, and 5 possible paths. User should add only the paths which are relevant for testing or have *high* business criticality.

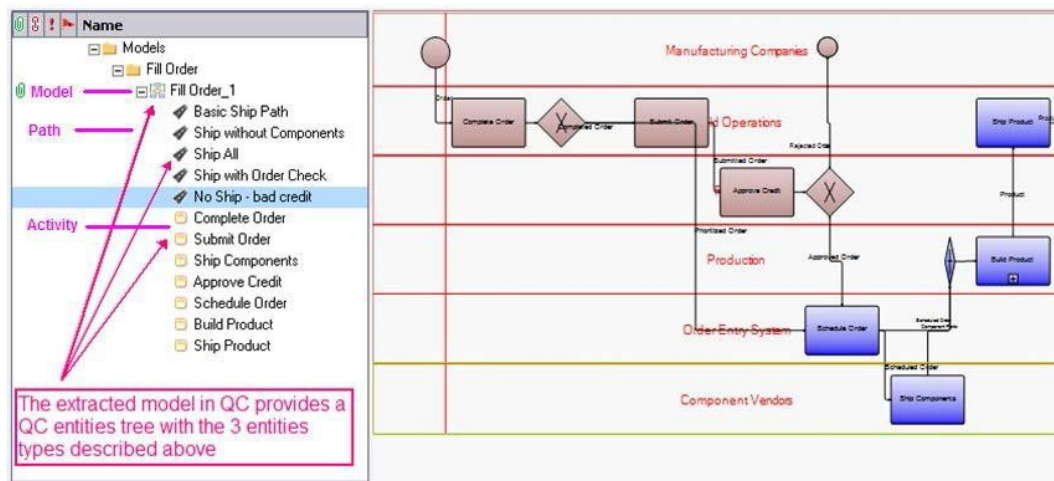


Figure: Screenshot from ALM Business Models module. The left pane displays the business process model tree and the entities structure under the model tree entity.

## Model entities as requirements in the ALM Requirements module

Each imported model automatically creates matching requirements in the ALM Requirements module. For clarity we will name these requirements ***representative requirements*** as they represent a model entity.

The '**Business Models**' folder in the requirements tree contains by default all the newly created representative requirements. These requirements have a special type and icon.

How is the tree structured?

- The folder contains a requirement for each model (For example, 'fill Order\_1').
  - Below each model is a system group node that holds requirements for each 'path' entity.
- Activities are added under the **Activities** folder and are reusable. If the same activity is included in three different imported models, it will be represented only once in the requirements tree.

This shared activity representation approach supports the concept that an activity should be associated to functional test requirements, which in most cases are shared by the activity regardless of the different models in which the activity is included. To specify unique test requirements for a specific model, customers are advised to use the 'path' entity.

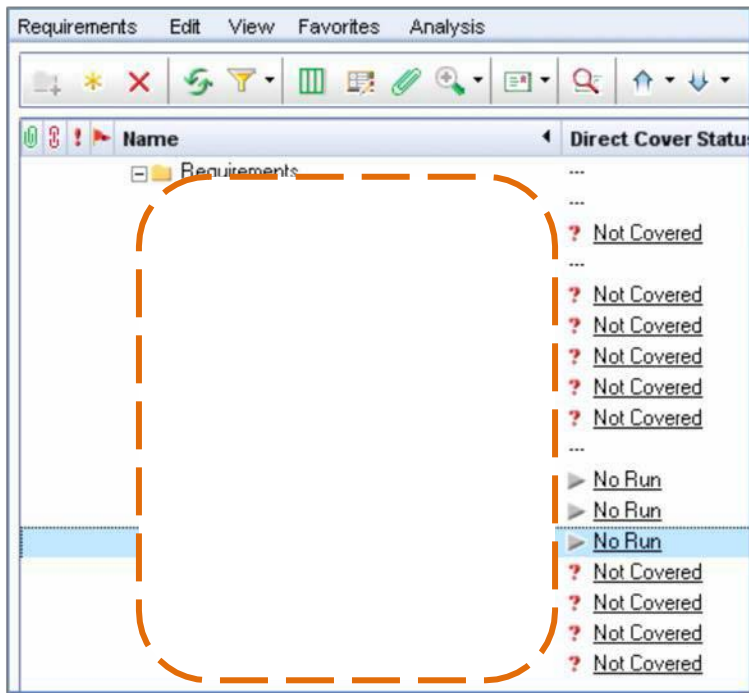


Figure: Screenshot of ALM Requirements module. The requirements tree includes the **Business Models** system folder, which contains all the newly created representative requirements.

## Creating coverage for models – various use cases

The representative requirement is simply a placeholder (an anchor) for model entity coverage. The main work in the Requirements module is to link other requirements and tests to create coverage.

This section describes four possible use-cases for connecting the model to requirements and tests. As different customers engage with ALM using different methodologies and construct the requirements hierarchy differently, the best practices for creating test coverage vary:

**Use Case #1:**

Customer does not use the ALM Requirements module at all. Only the Test Plan module is used.

**Best practice for creating coverage:**

In the Business Models module, create linkage between tests and model entities (activity, path...). Manually link relevant tests to activity, path, and model entities. In the background the tests are actually linked to the representative requirement of the model entity. This guarantees a single flow in the ALM system: Model > Requirement > Test. This operation has the identical result to the user adding test coverage to the entity's representative requirement in the Requirements module.

**Advantages:**

- Requirements creation and usage is transparent to these customers. No additional maintenance is required of them on the requirements side in order to view coverage status and create test sets based on the model scope.
- Enables customers to smoothly adopt the Requirements module once their organization becomes more mature and starts using requirements to create a powerful ALM process.



## Use Case #2:

Customer has an existing requirement tree structure, and tests are already linked to the requirements. For the purpose of this use case we can assume that the requirements tree consists of two folders: one contains the E2E Business Processes, and another details the functional requirements and specifications and associated test requirements. Due to this pre-defined requirements structure the user would **not want to** use the automatically generated representative requirements structure provided by the integration, but he would like to keep working with only the existing tree. Instead, he creates links between the existing requirements and the model entities.



Figure: General example of an existing requirements structure that the customer would like to maintain.

### Best practice for creating coverage:

In the Business Models module, create the link between existing requirements and the corresponding model entities. Manually link relevant requirements to activity, path, and model entities. By linking the model entities to requirements, ALM automatically identifies the tests linked to those requirements and uses them to define the coverage and quality analysis of the model entities.

### Limitation:

The Model Requirements system folder and the auto created representative requirements play an important behind-the-scenes role in maintaining the connections described above. This means that deleting this system folder and requirements (i.e. the 'representative requirements') will remove the coverage. This folder *must* remain untouched in the tree, even if it is not being used.

### Advantages:

- User uses his existing requirements tree structure and benefits from the existing coverage links between requirements and tests. Therefore, there is no need to create specific linkage between tests and model entities.
- The requirements tree can maintain a very flexible structure (frees the user from the parent-child relations to support coverage analysis) and ALM will still be able to provide aggregated quality and coverage status for each model activity and path, by scanning the linked requirements in the requirements tree to collect the pass/fail statuses. This provides for a much more flexible analysis than the child-parent relationship that the Requirements module enforces in order to view standard requirements coverage and quality analysis.

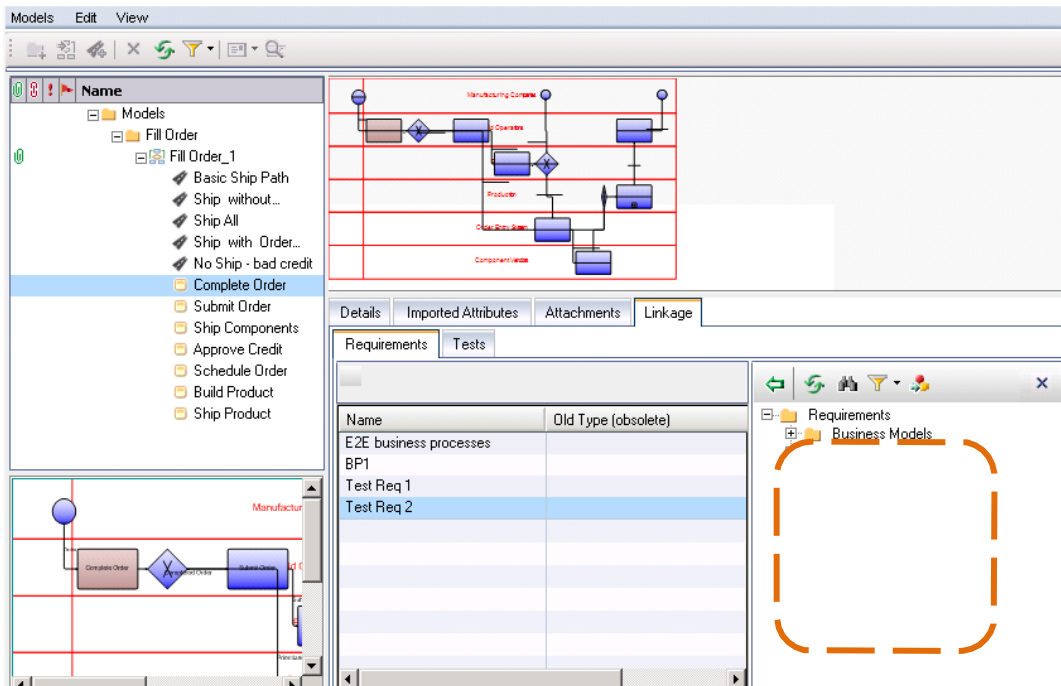


Figure: Screenshot from ALM Business Model Module. An activity entity is selected. In the bottom-right pane, the Linkage > Requirements tab enables linking existing requirements of other types to model entities. In this use-case the “Business Models” folder will be added to the customer tree –

but it is a minor addition that should not affect the process.

### **Use Case #3:**

Customer has an existing requirements tree structure, and only some of the tests are linked to the requirements. The customer wants these requirements tree to represent the model structure. In addition, the customer would like to use the automatically created representative requirements as a high-level structure of business requirements.

#### Best practice for creating coverage:

In the Requirements module, physically drag and drop existing requirements from their current locations in the requirements tree, to under the designated representative requirement.

As for additional tests which are not linked to requirements: these can either be linked to existing requirements of other types (as the customer did in the past) or linked in the Business Models module directly to activity, path, and model entities. The second method (of direct linkage in the business model module) will automatically connect the test to the high-level representative requirement. In practice, this results in creating test coverage for the representative requirement.

#### Advantages:

- The fastest way to create coverage for model entities. When asked to generate a coverage and quality status view, the Business Models module will automatically drill down to the children of the representative requirements in order to create an analysis on coverage and quality.

Name	Direct Cover Status	A
Requirements	---	
Business Models	---	
Fill Order_1	? Not Covered	
Paths	---	
No Ship - bad credit	? Not Covered	
BP1	---	ac
Sub BP1	---	ac
Ship with Order Check	? Not Covered	
BP2	---	ac
Ship All	? Not Covered	
Ship without Components	? Not Covered	
Basic Ship Path	? Not Covered	
Activities	---	
Ship Product	► No Run	
Functional 1	? Not Covered	ac
Test Req 1	? Not Covered	ac
Test Req 2	? Not Covered	ac
Test Req 3	? Not Covered	ac
Build Product	► No Run	
Functional 2	? Not Covered	ac
Schedule Order	► No Run	
Functional 3	? Not Covered	ac
Approve Credit	? Not Covered	
Ship Components	? Not Covered	

requirements tree after  
Dragging and Dropping  
operation

Figure: Screenshot from ALM Requirements Module. Existing requirements were moved to their new location in the requirements tree under the high-level representative requirement.

#### Use Case #4:

Customer wants to partially adopt the automatic representative requirements structure offered by the integration. The customer sees great value in the requirements that represent the model's 'paths', as these constitute end-to-end business processes in the organization that require testing. In addition, since this information is synchronized with the modeling tool, the QA manager receives an updated list of processes which is based on a single 'book of records'.

On the other hand, regarding functional requirements – the customer would like to keep the existing requirements tree intact. The tree was built carefully according to the application structure.

Customers are advised to adapt their requirements tree structure in accordance with this use-case.

#### Best practice for creating coverage:

Use a combination of the recommendations from use cases 2 and 3.

Activity entities: In the Business Models module, drag relevant requirements to the activities from anywhere in the requirements tree.

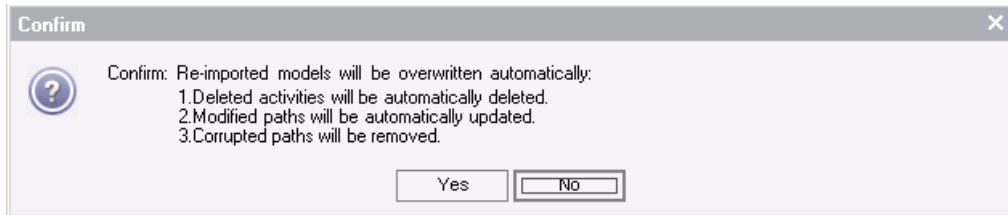
Path entities: In the Requirements module, either create designated requirements of various types as children of the path's representative requirement; or drag existing requirements which, until now, represented business process requirements, to be child requirements of the path's representative requirement. Tests that are created later can be linked directly to the path entity in the Business Models module (this is subject to the specific development of child requirements under the high-level path node, since the user might want to link the test to one of the children).

#### Advantages:

- This method gets the best of both worlds: A fast connection to create coverage for paths; and the ability to maintain a flexible requirements tree structure for functional requirements based on application folders structure, and not requirements per logical activity in the business process model.

## Re-Import Models – How does it work?

When Re-importing a model to ALM, the following confirmation message appears:



This chapter explains the different use-cases around re-import and the consequences of each change in the model. Assume this is the original model that was imported to ALM:

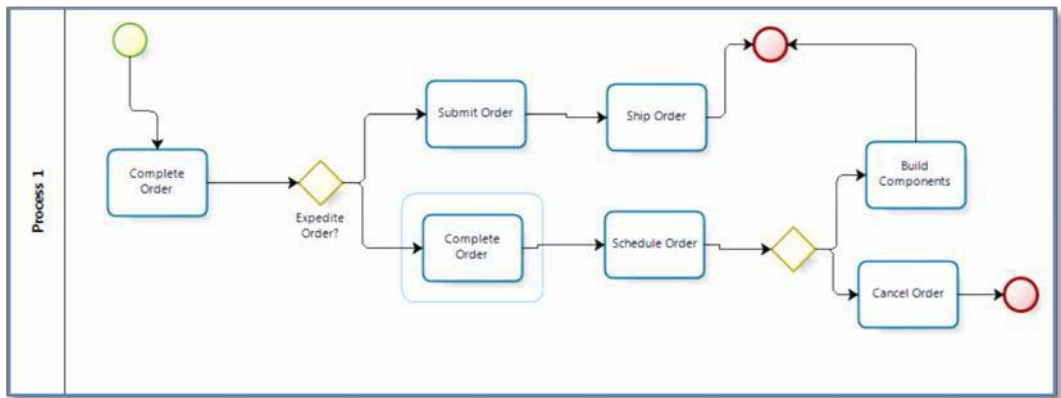


Figure: Sample business process diagram

The model has 7 activities and 3 paths. As discussed in the above use-cases, the Requirements module will contain 7 representative requirements for the activities, one representative requirement for the model, and 3 representative requirements for the 3 paths that will be stored under the model tree entry.

It is recommended that you create a baseline of the model prior to the re-importing models. After re-import, changes can be compared using the compare component of ALM's libraries baseline feature.

## Consider the possible changes:

### An activity was added

#### Scenario:

Model was re-imported and a new activity was added to the diagram

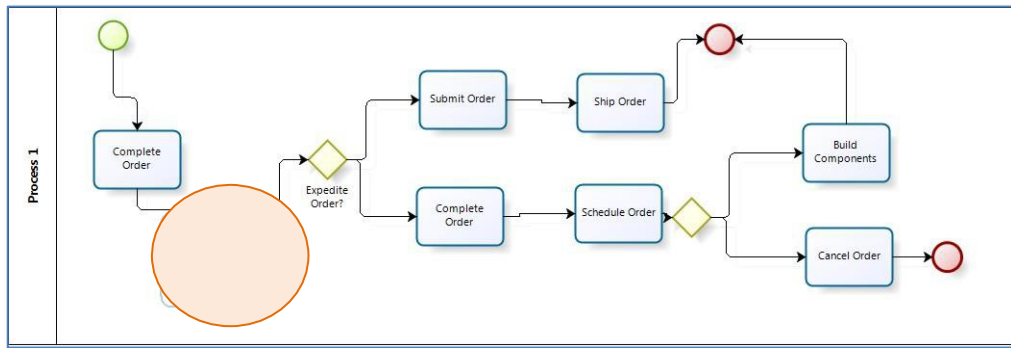


Figure: Sample business process diagram with a change

#### Effect:

1. “Validate Order Credit Card” activity was added to the model tree in the Business Models module.
2. “Validate Order Credit Card” – node was added to the Requirements module, ONLY IF this activity was not already imported to ALM from a different model:
  - a. If the activity was already imported to ALM by a different model, the activity is associated automatically with the existing activity representative requirement – and automatically gains all the coverage defined for this activity.
  - b. If this activity is new in ALM, a new representative requirement will be created for it, and the coverage will be blank. It is the user’s responsibility to add coverage after the re-import process is complete
3. All the paths will be affected by this change: they will all be automatically updated to include this activity in the end-to-end flow. An internal algorithm will

identify that each path needs updating and the path description will be modified to include the “Validate Order Credit Card” activity in the path content. Direct path coverage will not be affected, but when you drill-down to view path coverage, at the level of included activities, the coverage will automatically be extended.

### An activity was removed

#### Scenario:

Model was re-imported and an activity was removed from the diagram.

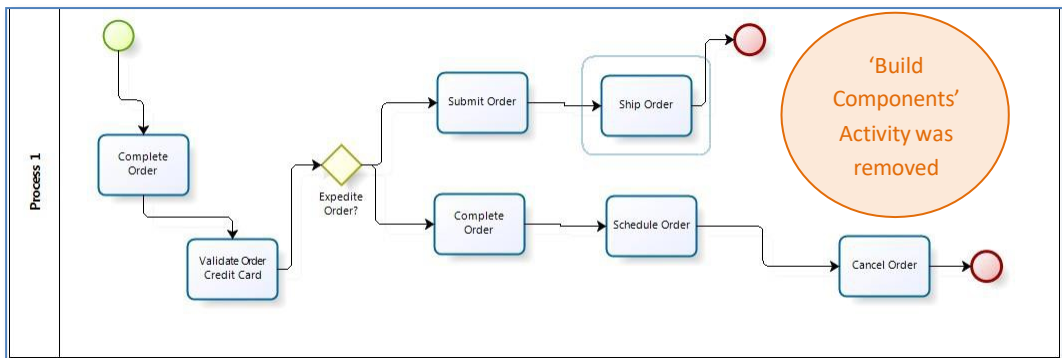


Figure: Sample business process diagram with a change

#### Effect:

1. “Build Components” activity was removed from the model tree in the business models module.
2. “Build Components” –requirement node was NOT deleted from the requirements module!

The connection between the activity and the representative requirement was removed.

If the “Build Components” activity is still valid in other models – there will be no effect on them or on the coverage of those activities.



NOTE: if the “Build Components” activity does not appear in any other imported models, the deletion will mean partial coverage loss:

- The representative requirement of “Build Components” will remain, but any requirements which were dragged and dropped to create coverage in the Linkage > Requirements sub-tab will be lost until a future import which will contain the “Build Components” activity. ALM will not be able to restore a link to an orphan representative requirement. Instead, a new representative requirement named “Build Components\_1” will be generated. User will then need to manually divert the coverage of tests and child requirements from “Build Components” to “Build Components\_1”.
3. Removing activities may affect paths in 3 different ways:
- a. No impact – the removed activity is not part of the path.
  - b. Recoverable change – the removed activity does not harm the uniqueness of the path and the auto-recovery algorithm can amend the path flow automatically.
  - c. Irrecoverable change – the removed activity harms the existing path in a way that cannot be restored. In this case, the path will be deleted.

Looking at our example above, one can see that the removal of ‘Build Components’ causes an ‘irrecoverable change’ for the third path. After the change, the path simply no longer exists and the newly imported model contains only 2 paths. This means that the path entity in the Business Models module will be deleted along with its direct requirement coverage. Similarly to deleted activities, the representative requirement of the path will remain, but will be orphaned.

## **An activity was modified**

### Scenario:

Model was reimported and activity was modified (Name, properties, additional links...)

### Effect:

No effect in ALM. The coverage and linkage remain intact. The properties of the entity will be modified. (If the entity is checked-out by another user, an alert will be added to the model entity.)

## **Status view display – presenting accurate results**

The Status view provides full flexibility in presenting different cross-sections of coverage.

### **“Include requirement children” checkbox**

Depending on the use-case the customer selected to construct his representative requirements, aggregated results from requirements children can be added to the status calculation. For example: if a customer added child requirements to the representative requirements and those are covered by tests – then the child requirement does not need to be manually linked to the model entity. Select this checkbox for the analysis to aggregate also tests coming from children of associated requirements.

### **“Include Representative requirements” checkbox**

For customers who do not want to use the automatically generated representative requirements structure, this checkbox should not be selected. Thus, the coverage from the representative requirements will be ignored.

## Frequently asked questions

- **Q:** What happens if I delete a representative requirement from the Requirements module?
- **A:** The coverage of this entity will be lost. Any operation that creates coverage for the related model entity will automatically re-create a new representative requirement and link the new assets to it. The new requirement will be located under the default location of the “Business Models” node.
- **Q:** How can I extract change impact testing information from re-imported models?
- **A:** For this first release, this capability is not implemented. The user will have to manually trace changes after re-import and create the test set from it. For the next release a built-in feature will be developed to easily extract this information.

## Glossary

BPM	Business Process Model
Representative requirement	Entry in the ALM Requirements module. Representative requirements are generated automatically for each imported business model entity. They serve as the 'connector' between a business process model entity and other ALM entities (requirements and tests).
Orphan representative requirement	If an entity was removed from a business process model, re-importing the model does not delete the representative requirement of the removed entity, as it may hold critical information. It is up to the user to delete those nodes after any necessary information was moved to other places in the requirements tree.