SAPIENS
Partnering for Success

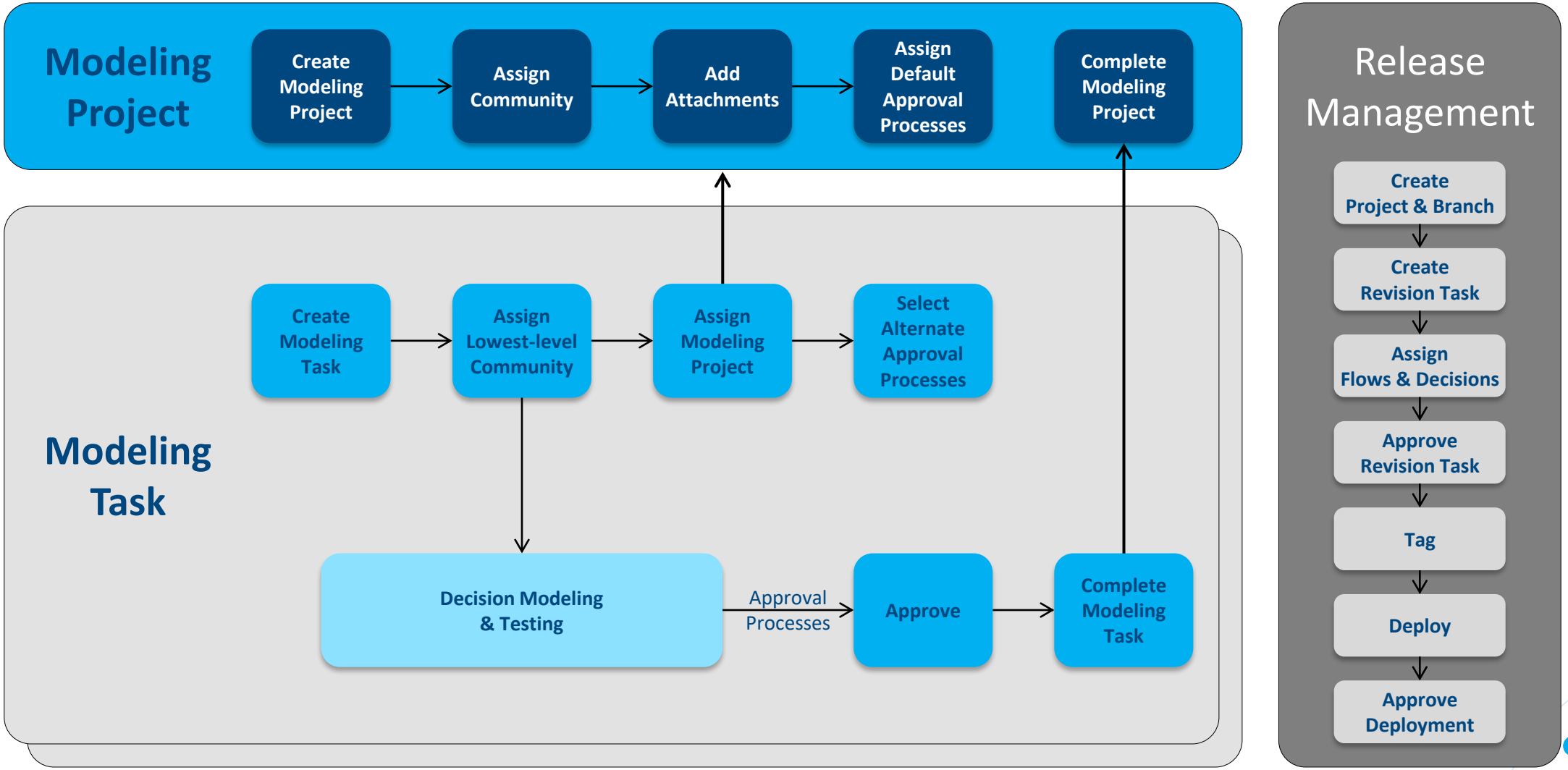
Decision Training for Release Managers

Agenda

- **Introduction**
- Release Management
- Approval Strategy

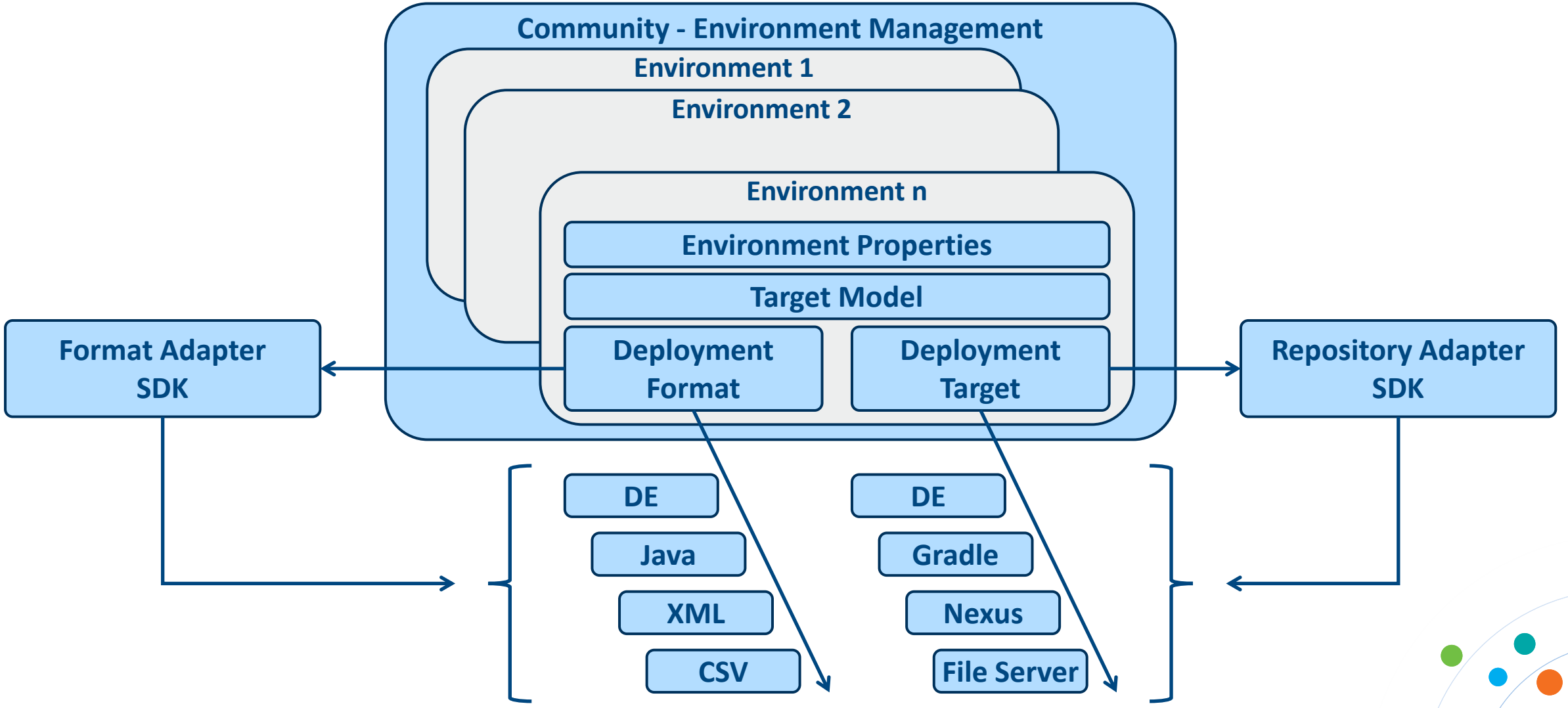
SAPIENS

The Modeling Process

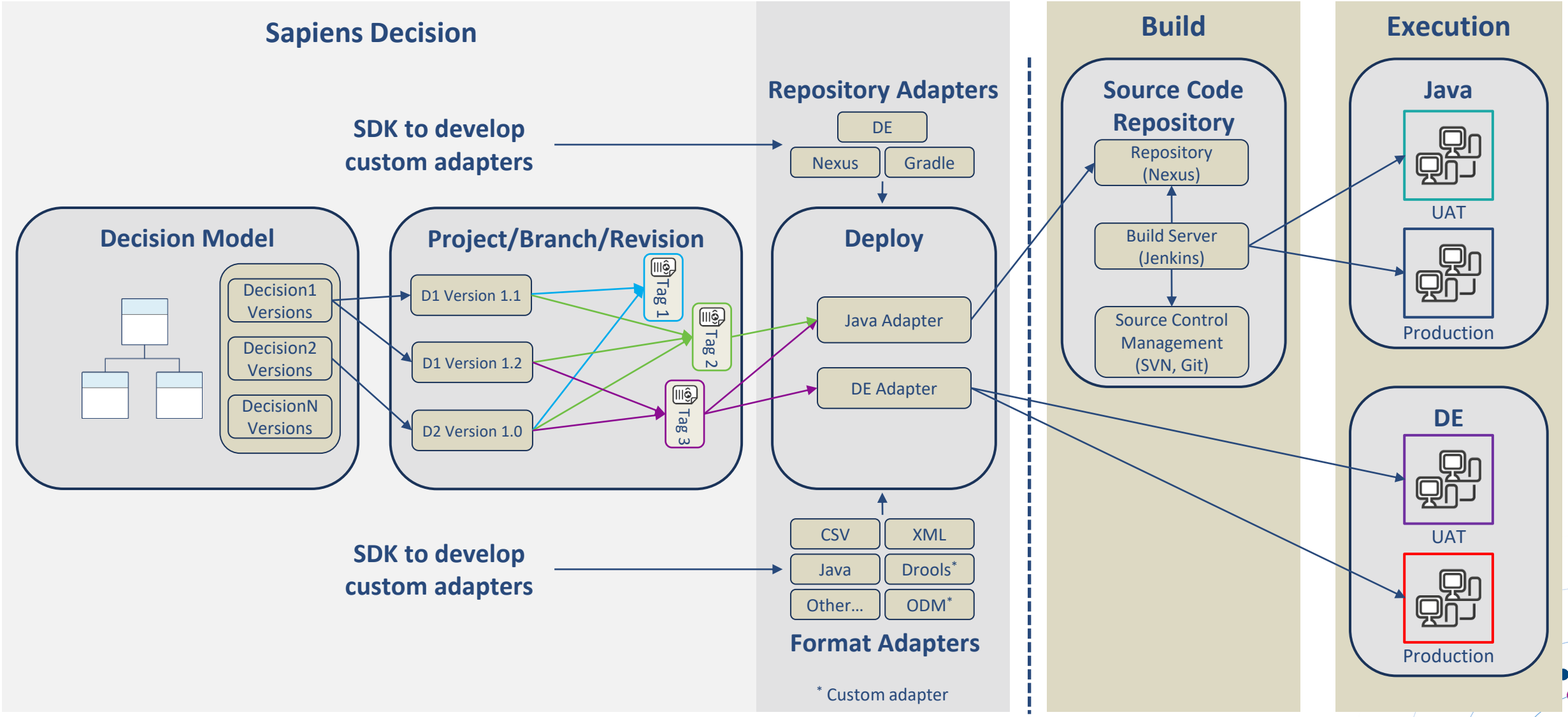


SAPIENS

Deployment Environment Management



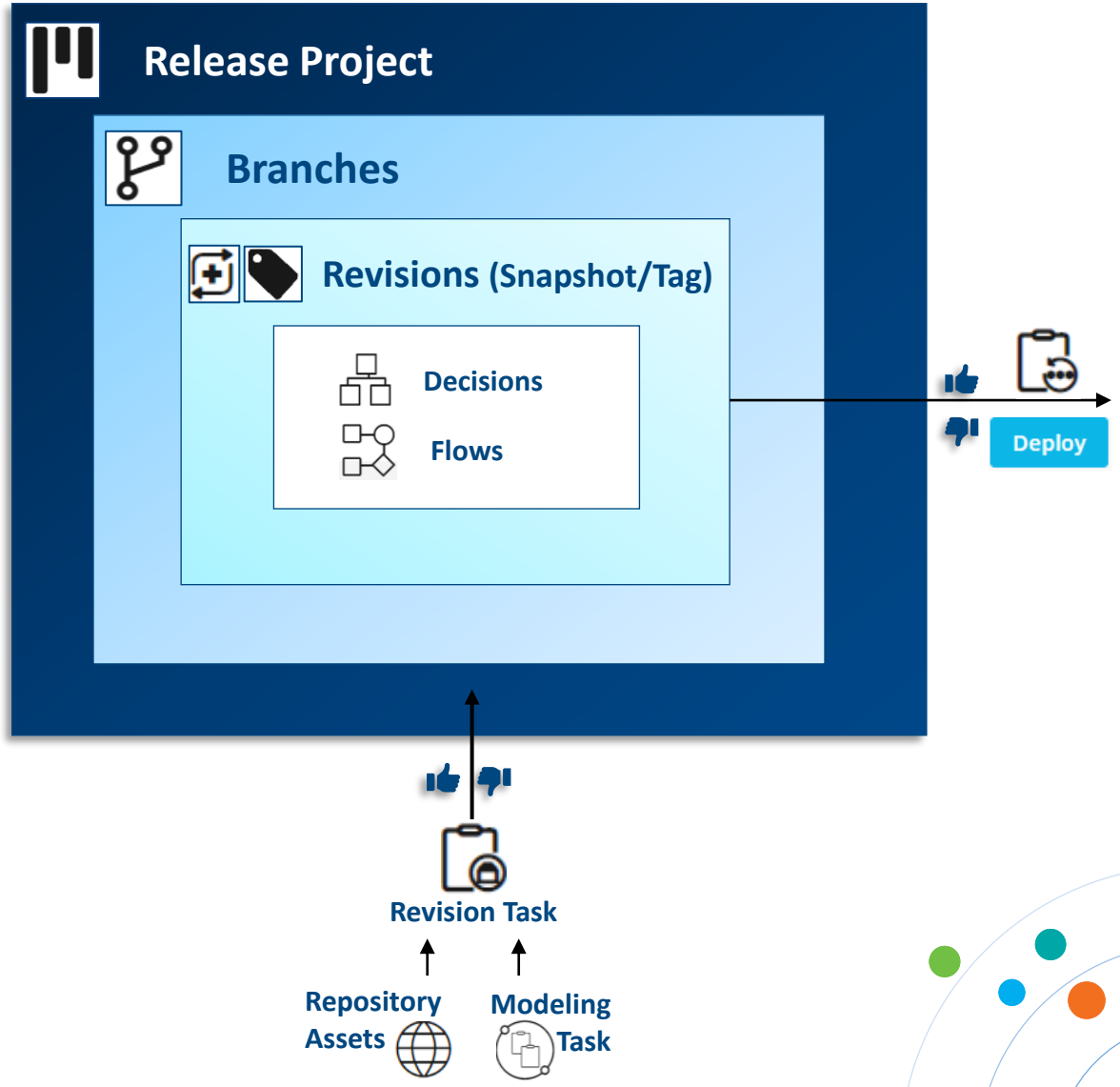
Deploying Executable Logic with Sapiens Decision



SAPIENS

Release Management

Release Management facilitates fast and easy release building and deploying

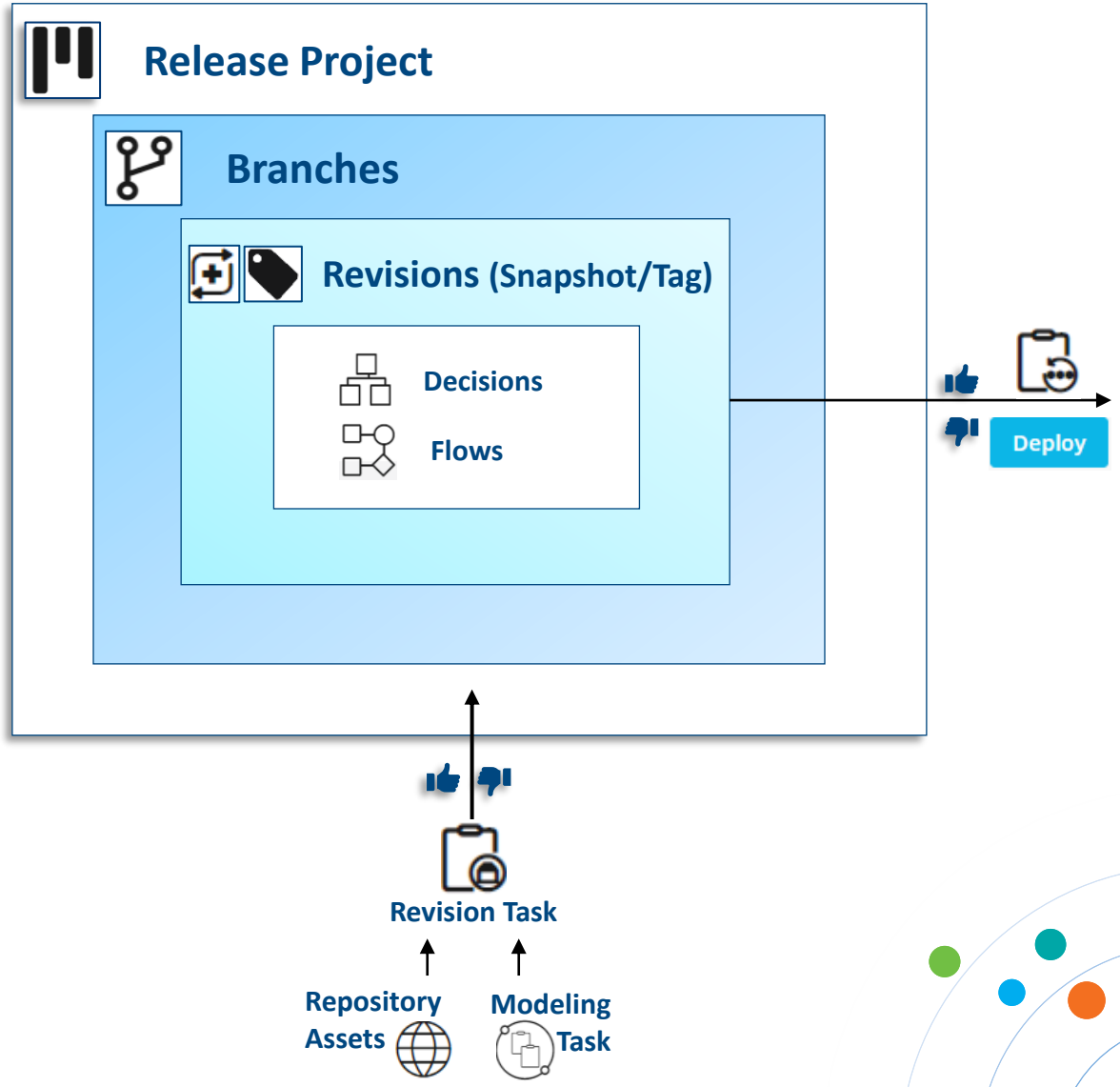


SAPIENS

Release Project

A **Release Project** is a workspace for collaborative management of packaging and deploying Decision assets

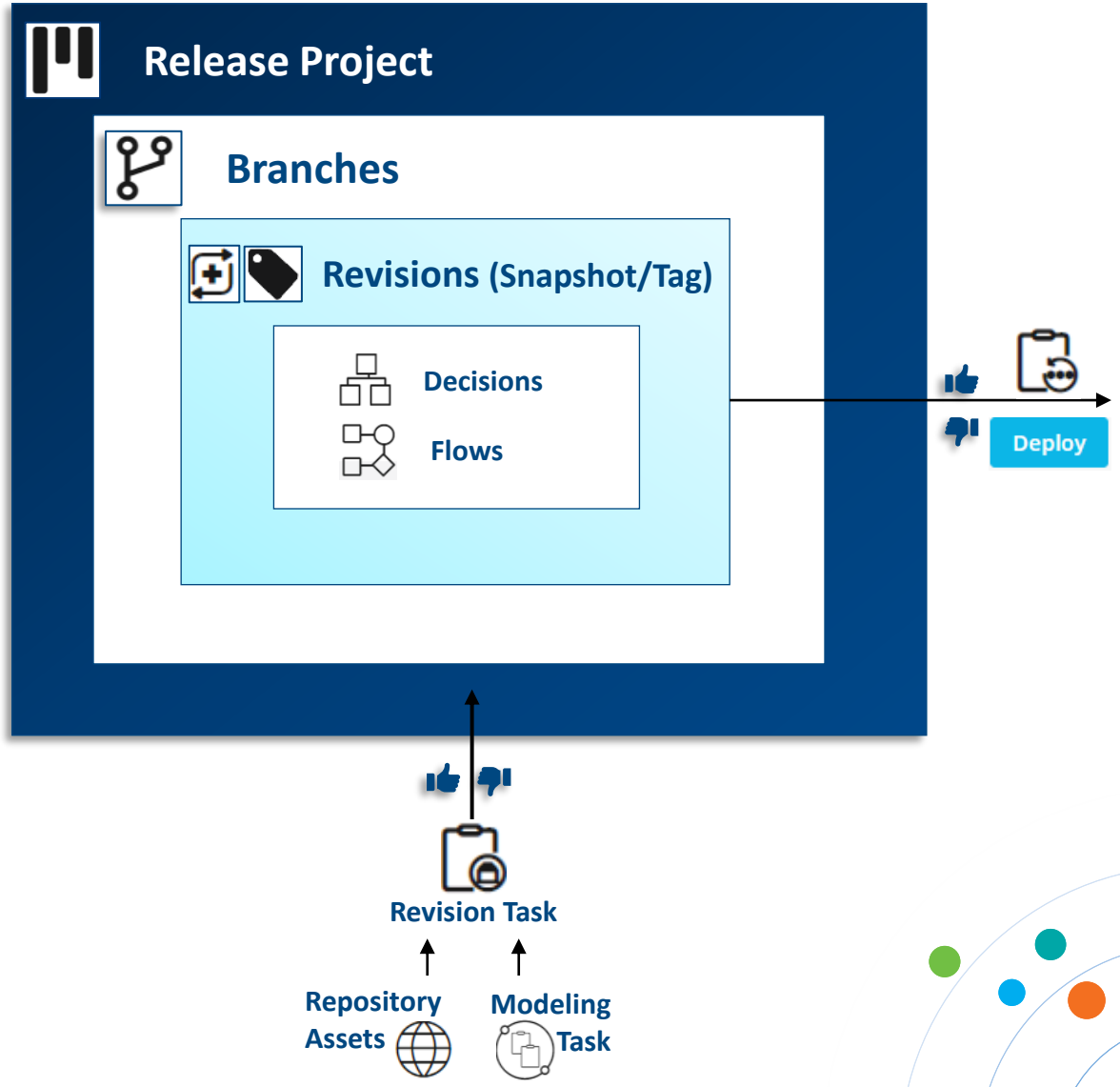
- Release Project is defined at Community level
- A Release Project contains Branches
- Branches contains Revisions
- Revisions contain assets
- A Release Project has owners, responsible for approving changes to its Branches
- A Release Project has Custom Properties
- A Release Project can be retired



SAPIENS

Branch

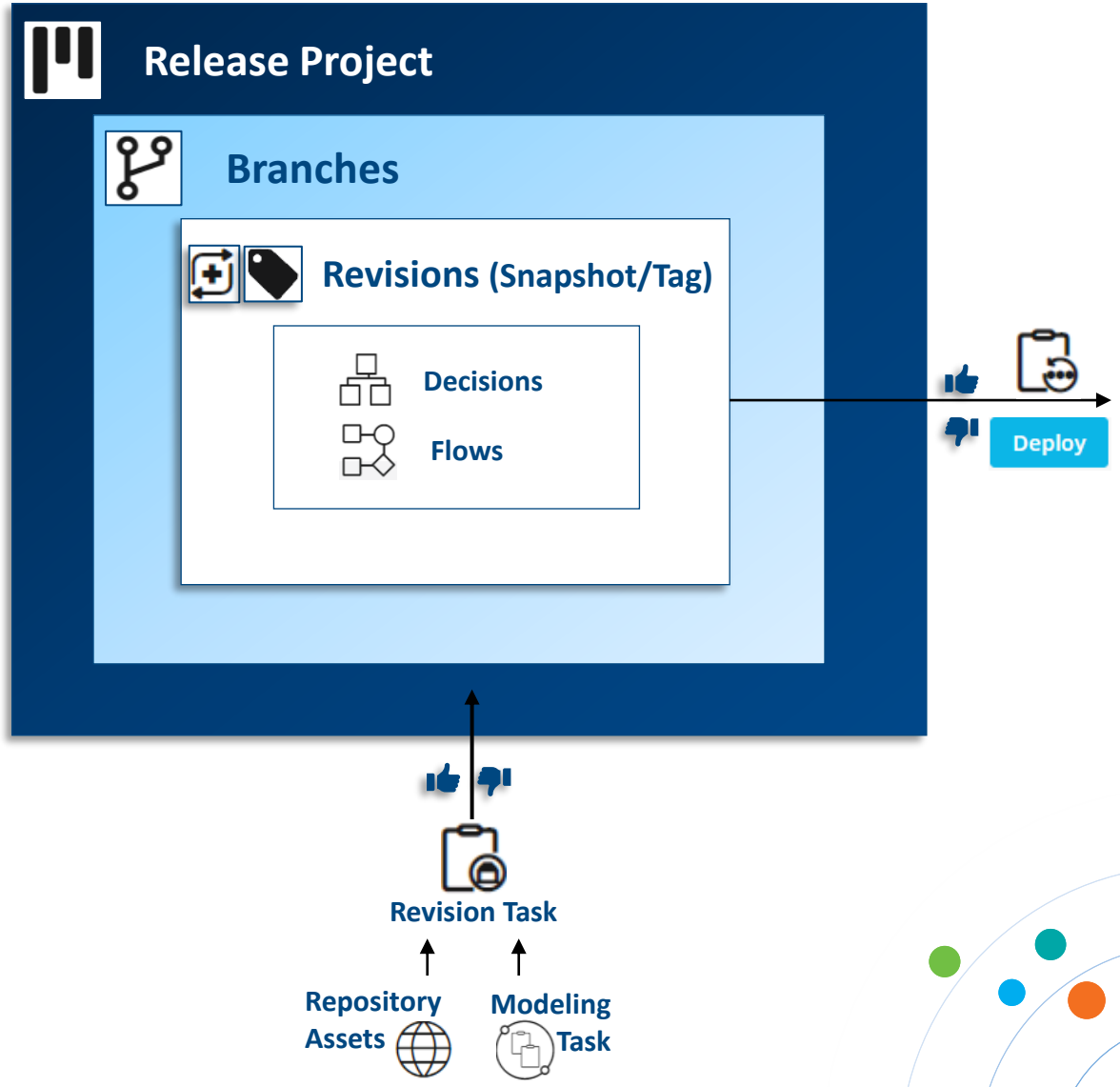
- A **Branch** is a group of Revisions created one after the other
- A Branch stores data as a series of snapshots (Revisions)
- The 'Main' Branch is the first Branch of a Project
- A Branch name can be changed at any time
- Point in time Branch content equals the Revision content at that point in time
  - Latest Revision holds the content of the Branch at its current end point
- A new Branch can be created from any Revision along another Branch
- Branch content (from any Revision) can be merged into the latest Revision of another Branch
- A Branch can be retired



SAPIENS

Revision & Tag

- A **Revision** is a self-contained snapshot of the Branch collection of approved/candidate assets and its Deployment Descriptors
  - Revision ID is an auto-generated sequential number and cannot be changed
  - Revisions are created in sequential order, a new Revision is always created after last Revision
- A **Tag** is a Revision with a customizable label (usually meaningful for the organization)
  - Tag Revisions are shown on the screen with tag icon and label instead of Revision ID
- The list of assets and its Deployment Descriptors included in a Revision (Snapshot/Tag) cannot be changed
- Custom Properties for Revision (Snapshot/Tag) can be changed at any time and are implemented immediately
- A Deployment is performed on a Revision (Snapshot/Tag)

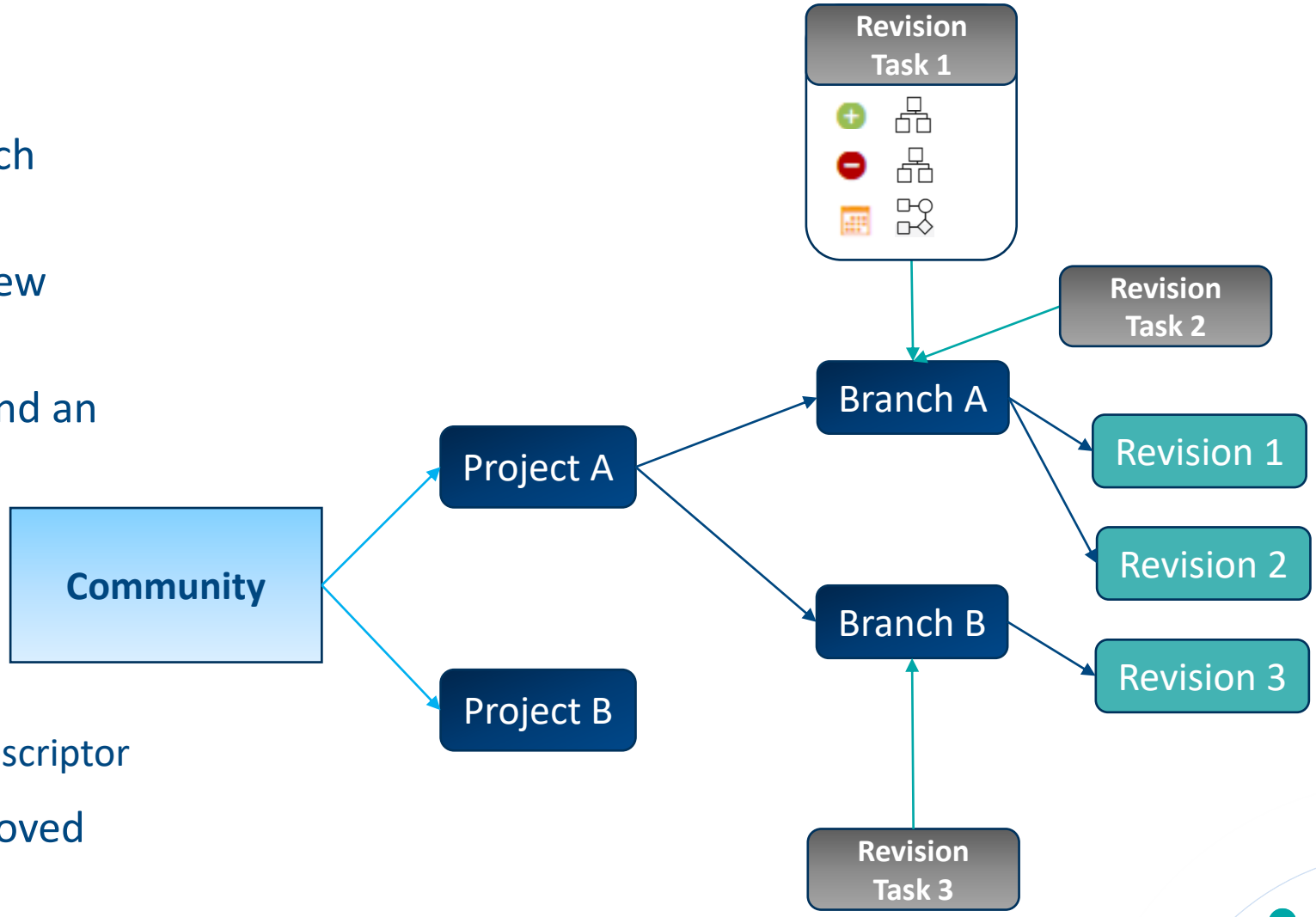


SAPIENS

Revision Task

A **Revision Task** is the means by which Branch content is changed

- Each Revision Task creates one new Revision in the Branch
- A Revision Task contains assets and an action per asset
- Actions are:
  - Add asset to Branch
  - Remove asset from Branch
  - Change asset Deployment Descriptor
- A Revision Task needs to be approved before it is applied



Demo

- Create Release Project (in a Community that has no DD Custom Properties)
  - Use Approval Strategy None
- Create Revision Task 1
  - Add asset A V1.0
- Create Revision Task 2
  - Add assets B, C
- Show Revision Graph
- Create Revision Task 3
  - Add asset A version V1.1
    - Show auto adjustment of dates
  - Remove asset B
  - Don’t submit Revision Task 3 for approval
- Tag Revision 2 as Tag1

SAPIENS

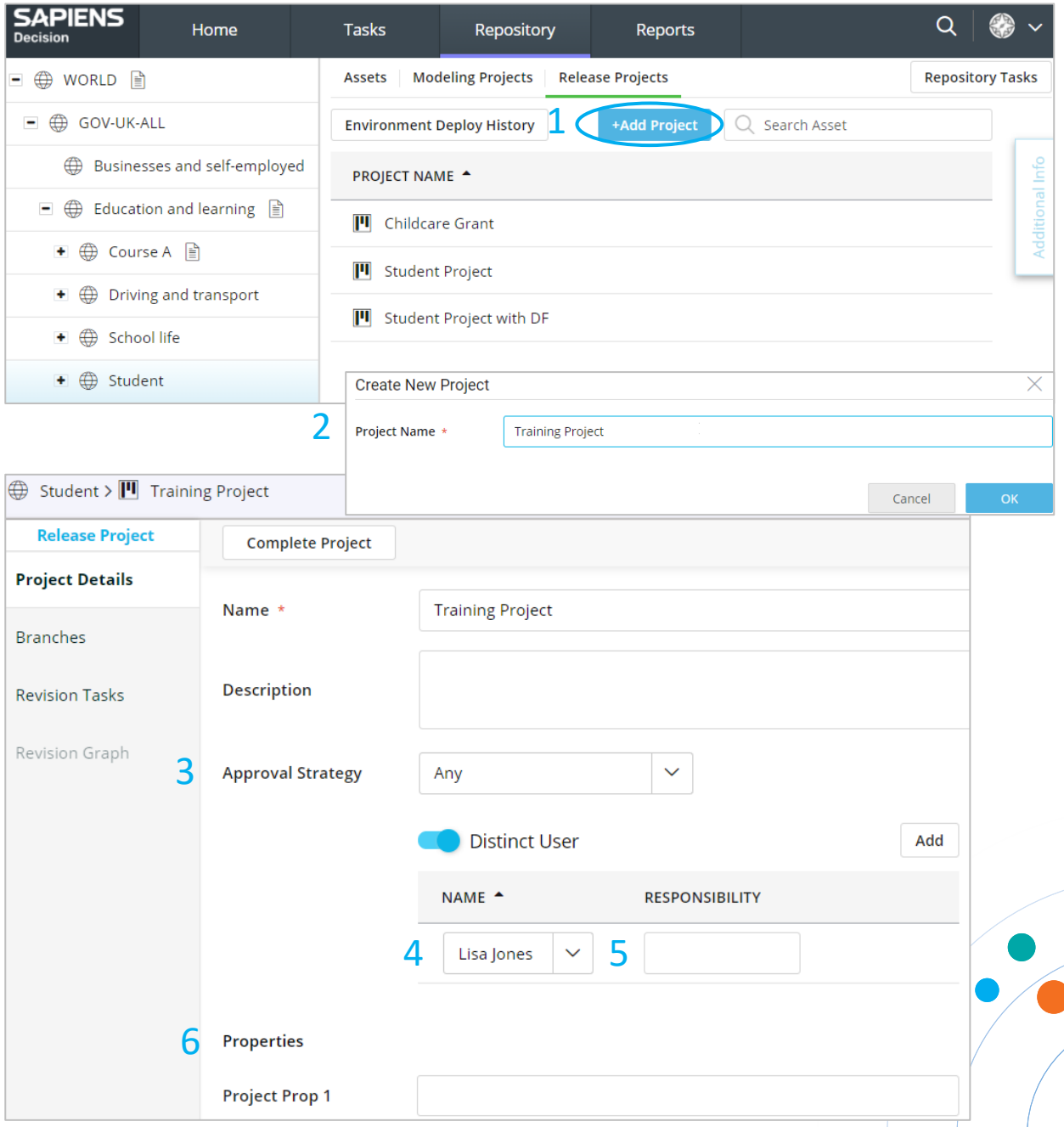
Agenda

- Introduction
- **Release Management**
- Approval Strategy

SAPIENS

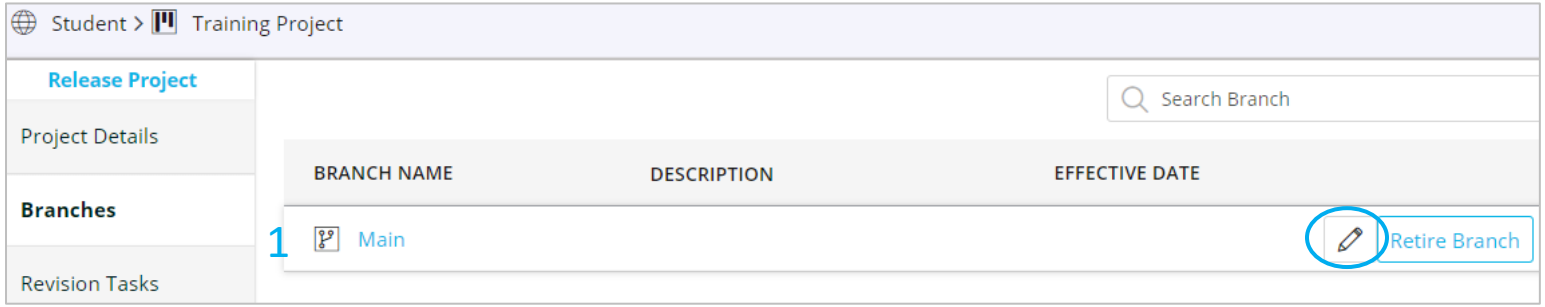
Create a Release Project

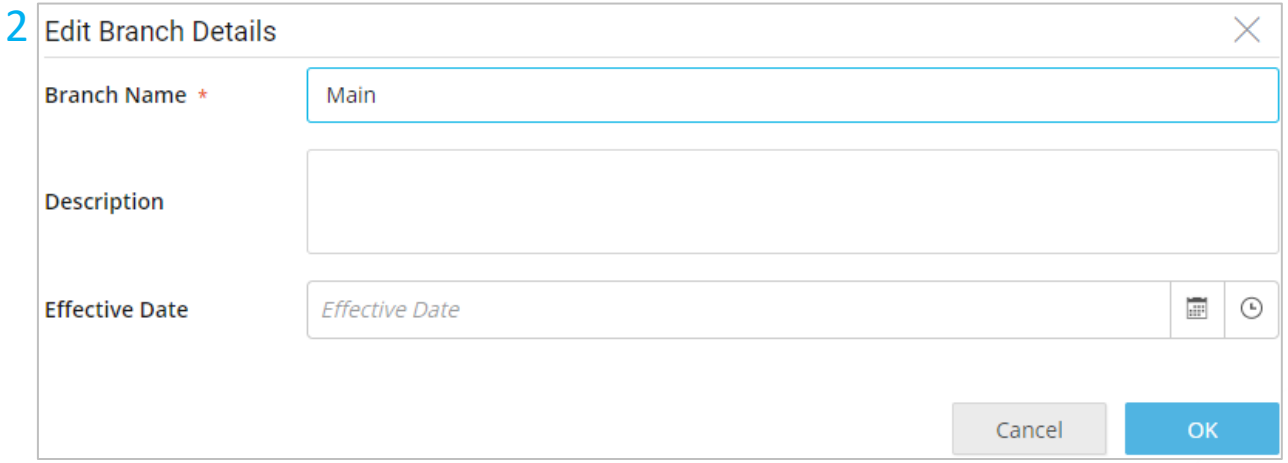
1. Select the Repository tab and choose a Community in the navigation tree and select an existing Project or click +Add Project to add new
2. Insert a unique Project Name and optionally a description
3. Set the Approval Strategy
4. Assign owners to Approval Strategy for the Project (All/Any/None)
5. Optionally, add a Responsibility description for each approver (owner)
6. Optionally, a Project can contain Custom Properties



Review Project Definition

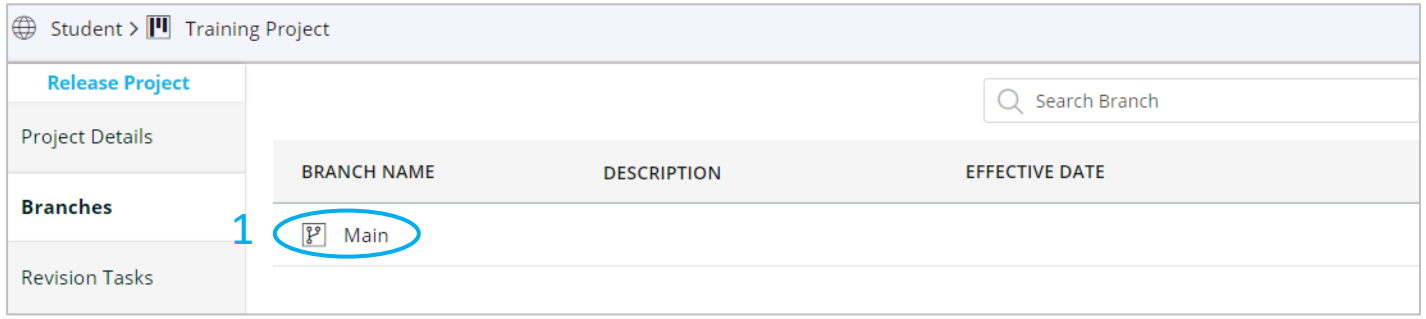
1. On Project creation a default Main Branch is created
2. The icon that appears (when hovering over the Branch row) enables the ability to edit the Branch Name and Description and set the default Effective Date used when adding an asset to the Revision Task



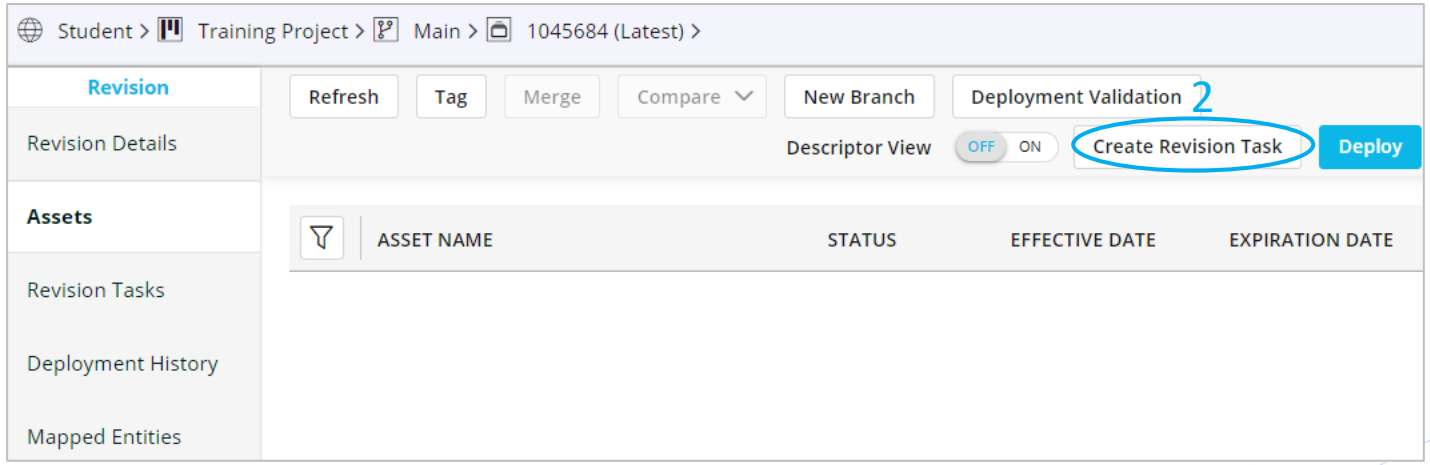


Create Revision Task

1. From the Project screen select the Branch

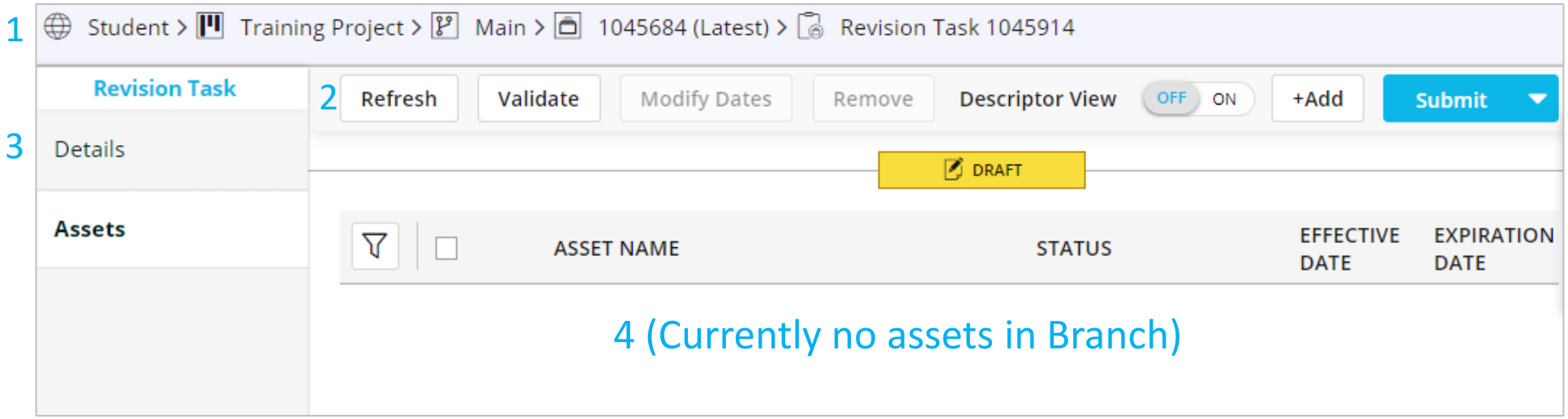


2. From the Branch screen click Create Revision Task button



Revision Task

Community > Project > Branch > Revision > Revision Task

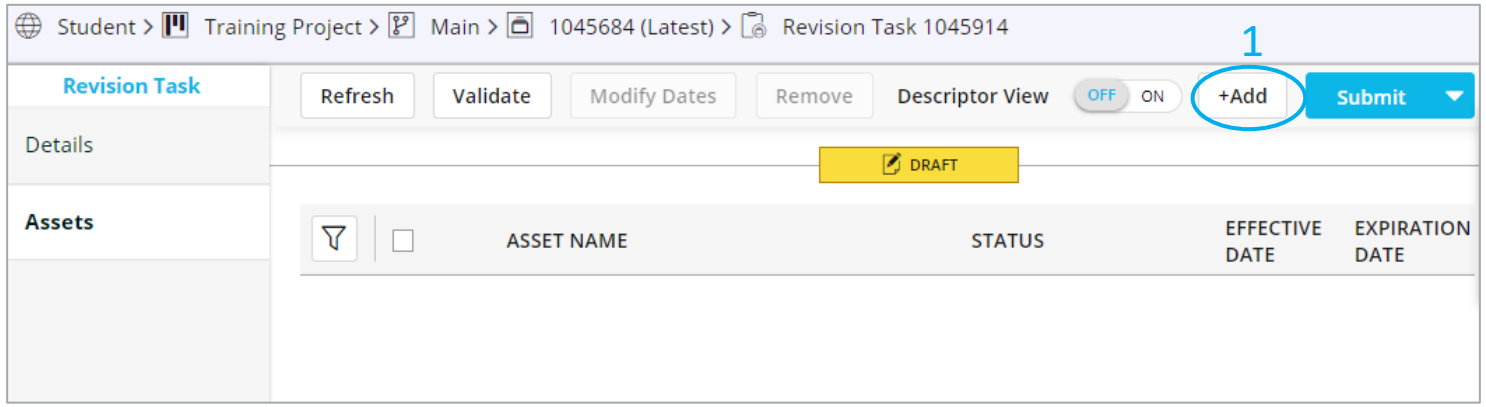


The Revision Task applies to the latest Branch-Revision content

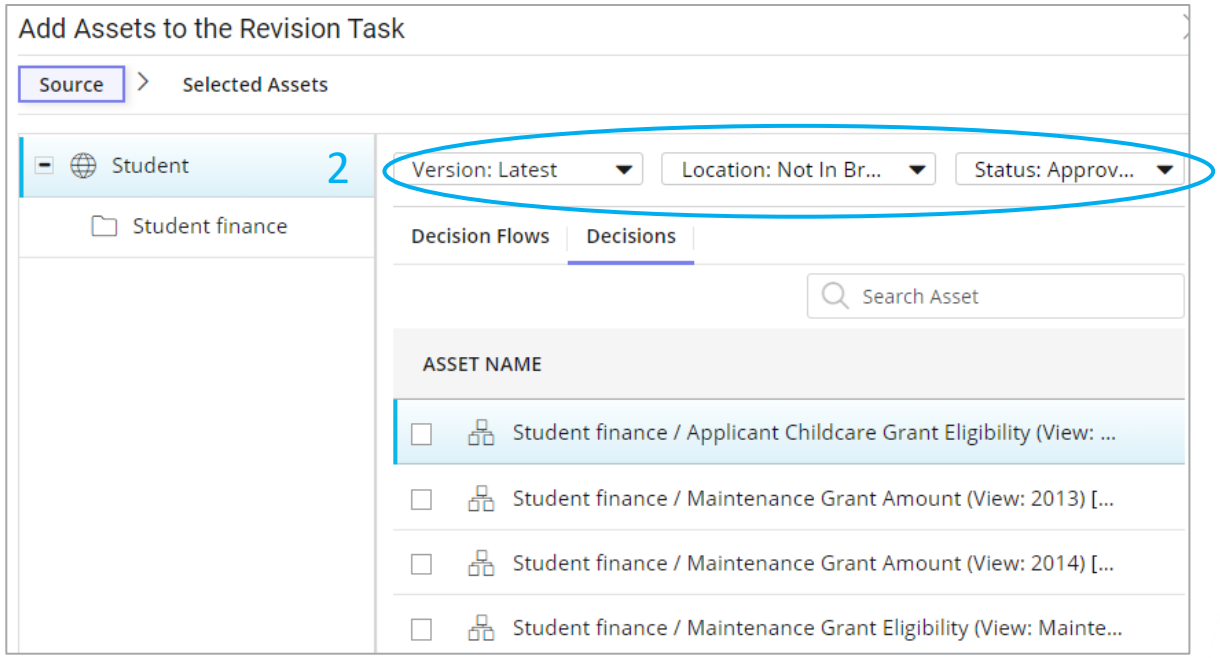
1. Bread crumb enables easy positioning and navigation
2. Use Vertical Tabs to adjust the Revision Request content
3. The Details Tab on the left contains Revision Request ID, Description, Modeling Project, Project Name, Modeling Task Name, Creator, and Creation Date
4. Display shows current Revision Task content and impact of changes on the latest Revision

Add Assets to Revision Task

1. To add assets to the Revision Task, click the +Add button

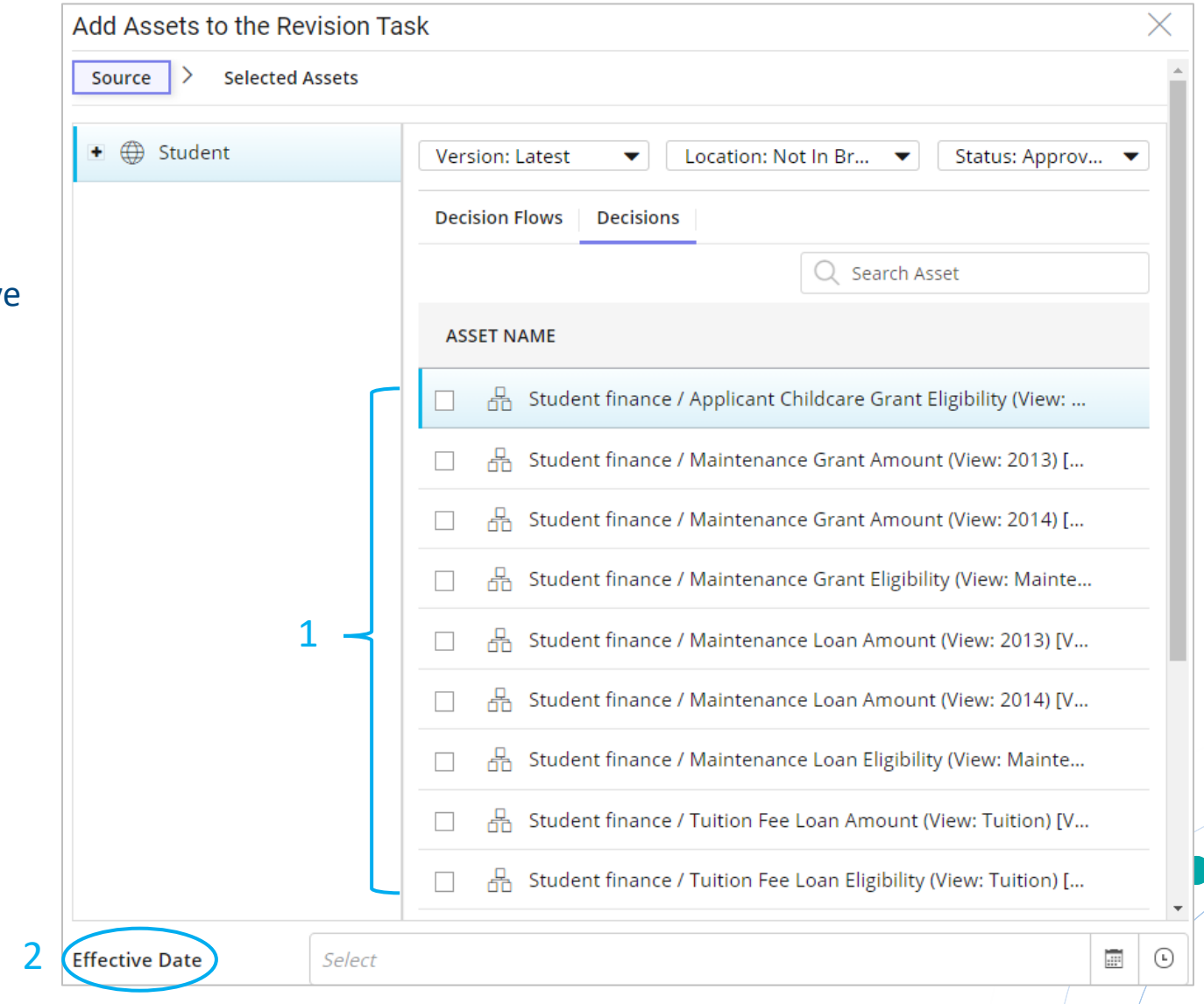


2. Use filter options to locate assets



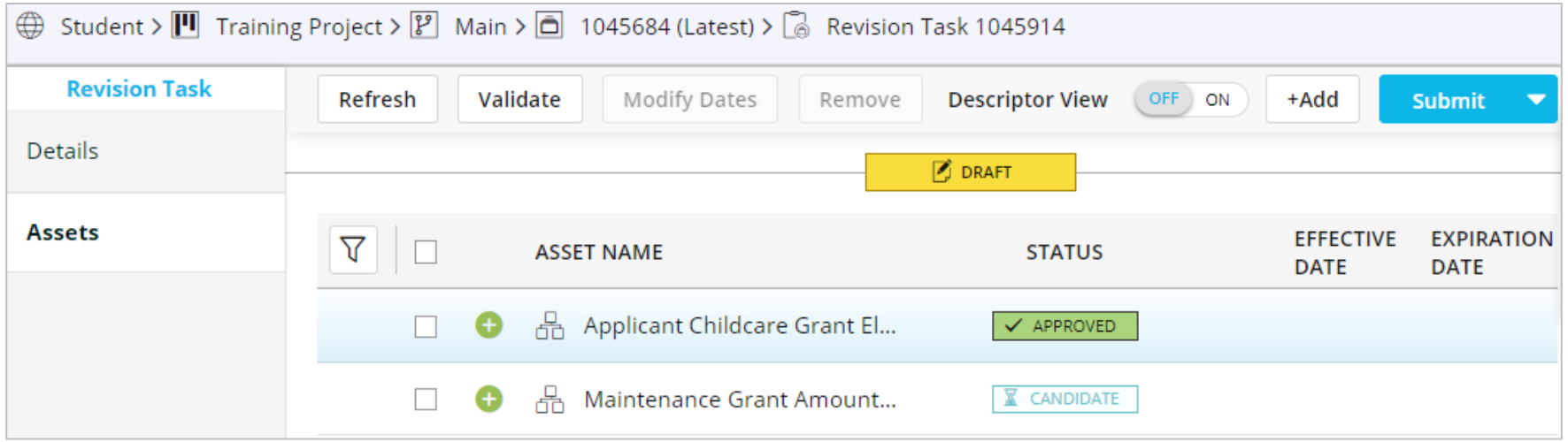
Add Assets to Revision Task

1. Multiple selection is available
2. The default Effective Date can be changed for the selected assets
   - If provided, the Effective Date will be set for all selected assets
   - It is not mandatory to add/modify Effective Date now, it can be done later



SAPIENS

Add Assets to Revision Task

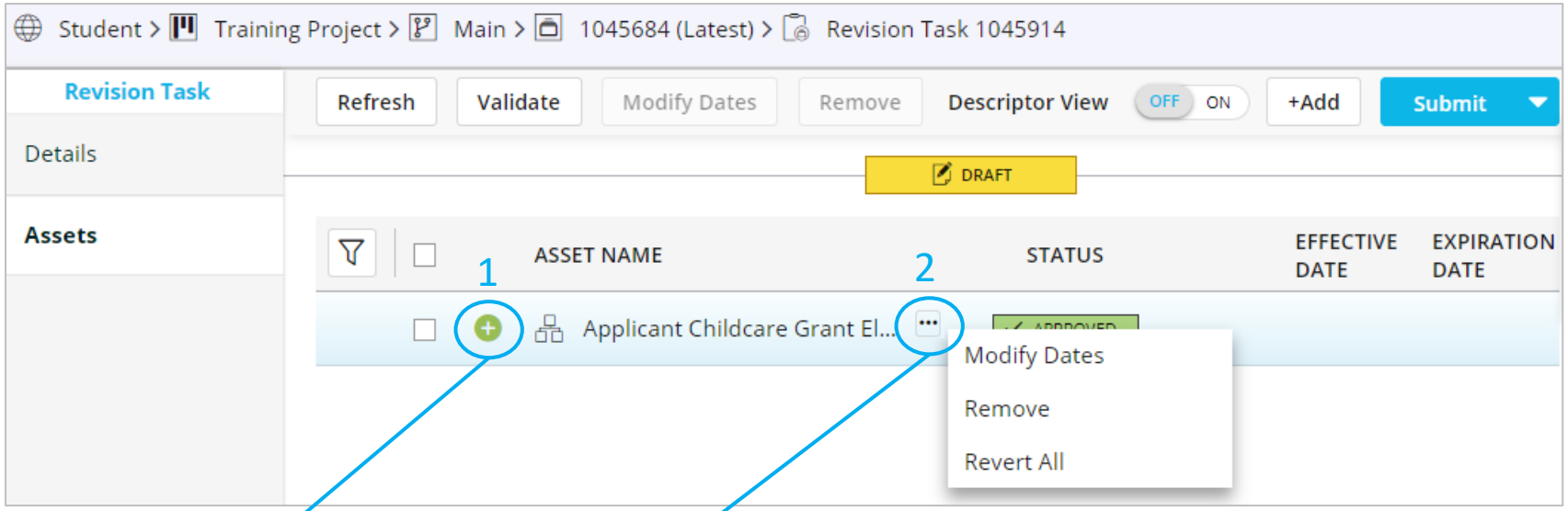


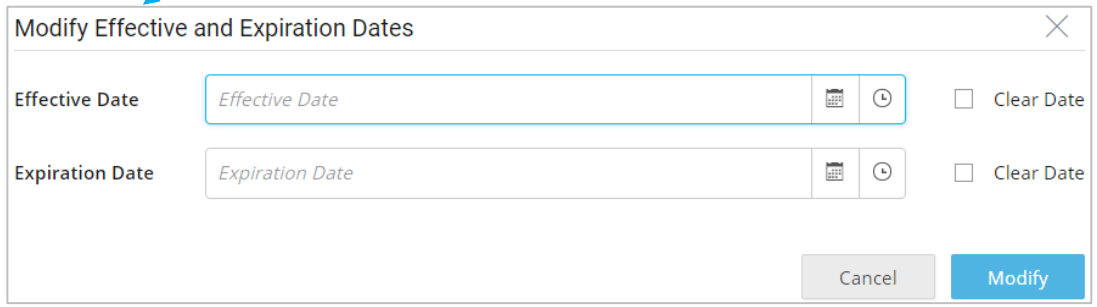
- It is possible to submit Approved assets and/or Candidate assets
- In order to allow Candidate assets in Revision and in deployment, setup actions must be completed

Assets in Revision Task

Through **Revision Task**, assets can be added or removed

1. An icon will be displayed based on the request made on the asset
2. By hovering over the right of the asset name and clicking ••• the Asset may be removed, a change may be reverted, or Effective Date or Expiration Date may be selected

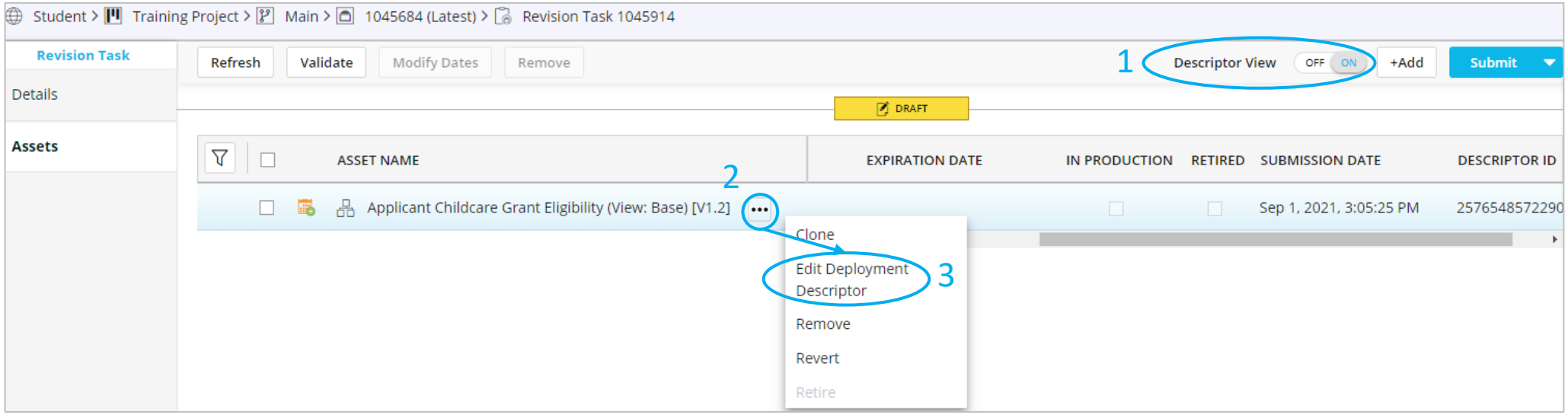




Asset Icon Legend

- Asset Added
- Asset Removed
- Deployment Descriptor Added
- Deployment Descriptor Changed
- Deployment Descriptor Removed

Descriptor View

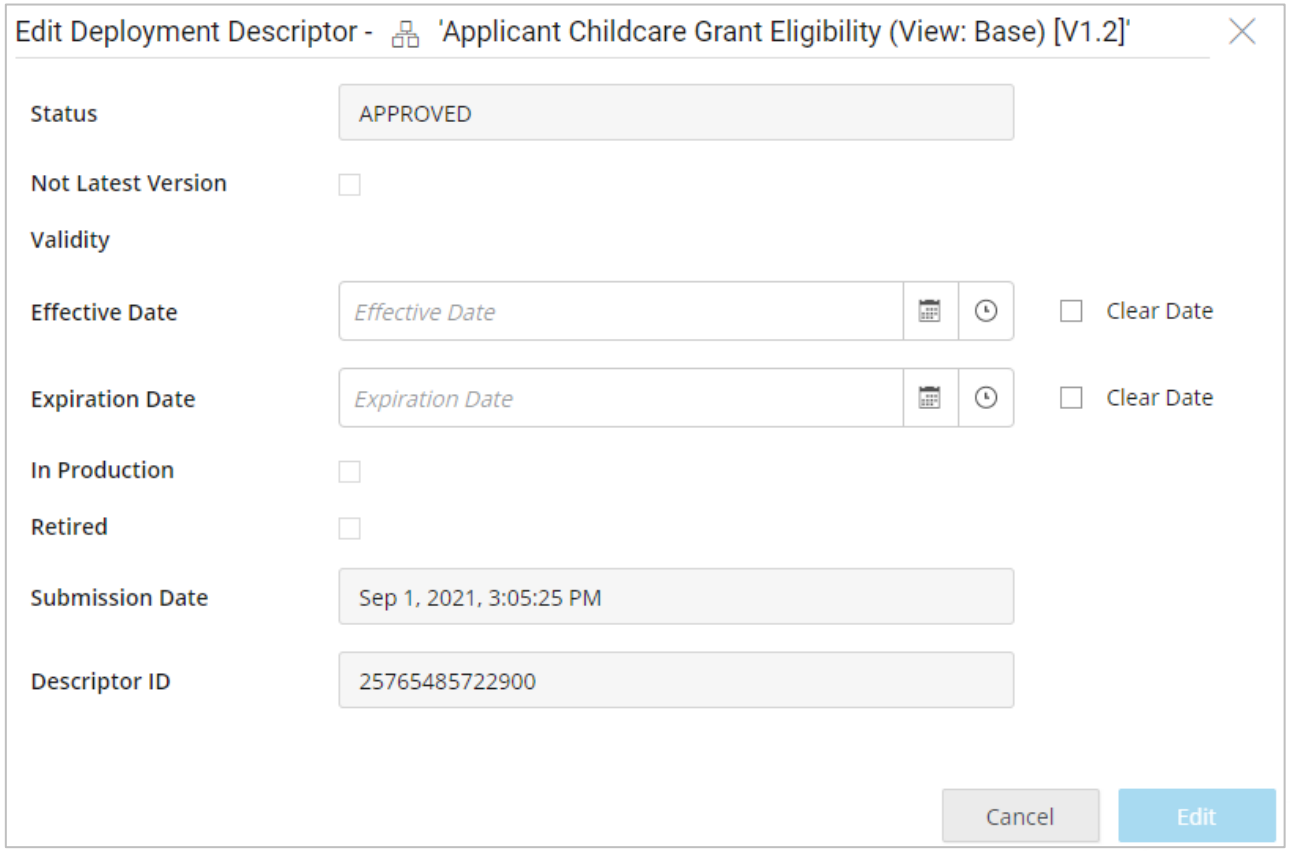


1. By turning on **Descriptor View** OFF ON it can be viewed whether the asset is in production or retired in addition to the Submission Date and Descriptor ID
2. The **Edit Deployment Descriptor** option can also be accessed by hovering to the right of the asset name and clicking •••
3. Custom Deployment Descriptor criteria may be added by an administrator

Effective and Expiration Dates

Effective & Expiration Dates for each asset can be changed using **Edit Deployment Descriptor**

- If a default Effective Date is added to the Branch when adding a new asset to the Revision Task, it is not necessary to specify an Effective Date
- If a default Effective Date is not added to the Branch and there is only one version of the asset, the Effective Date will be “Always Effective”
- If there are multiple versions of the asset and no effective and expiration dates, an overlapping error will occur

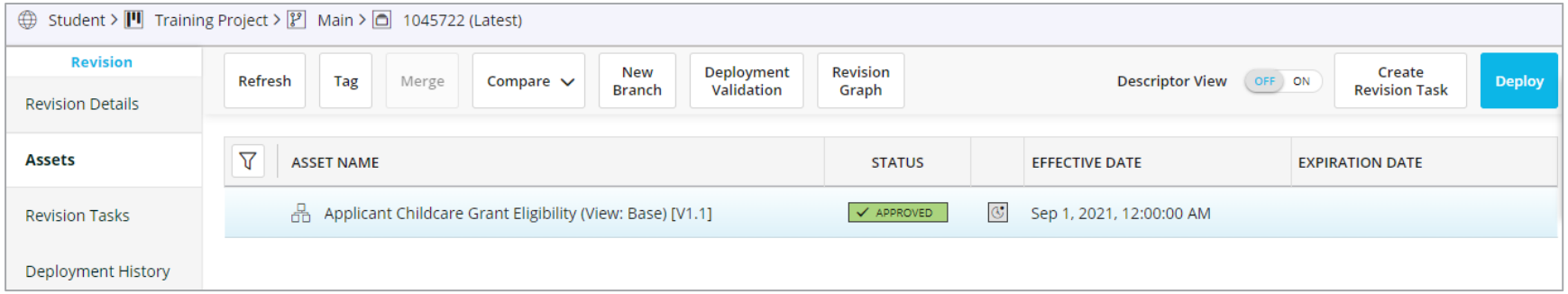


Effective and Expiration Dates

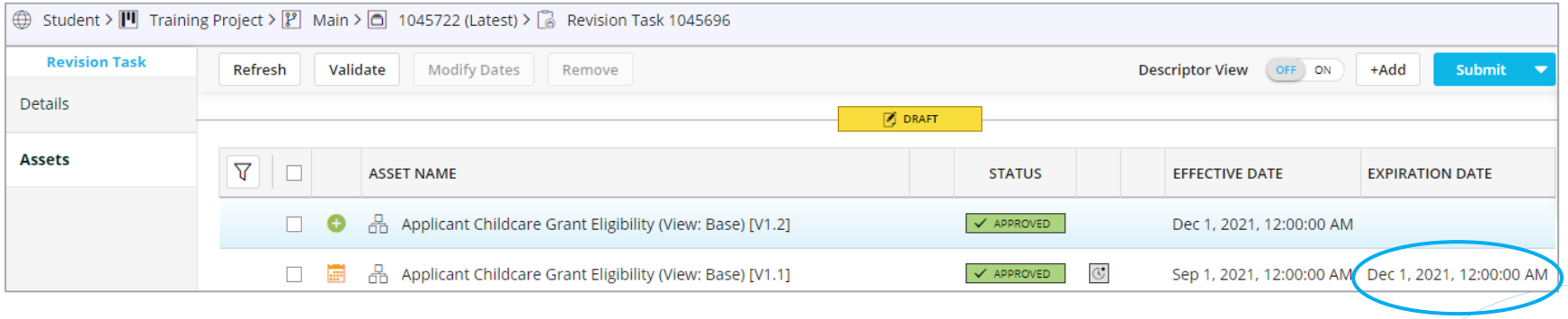
When adding a new Version of an asset to a Revision Task and Effective Date is inserted, the Expiration Date of the previous version of the asset will be changed automatically according to the Effective Date of the new version.

If necessary, the user can change the Effective Date manually.

Asset 1, Version 1.1, effective from Sep 1st with no Expiration Date



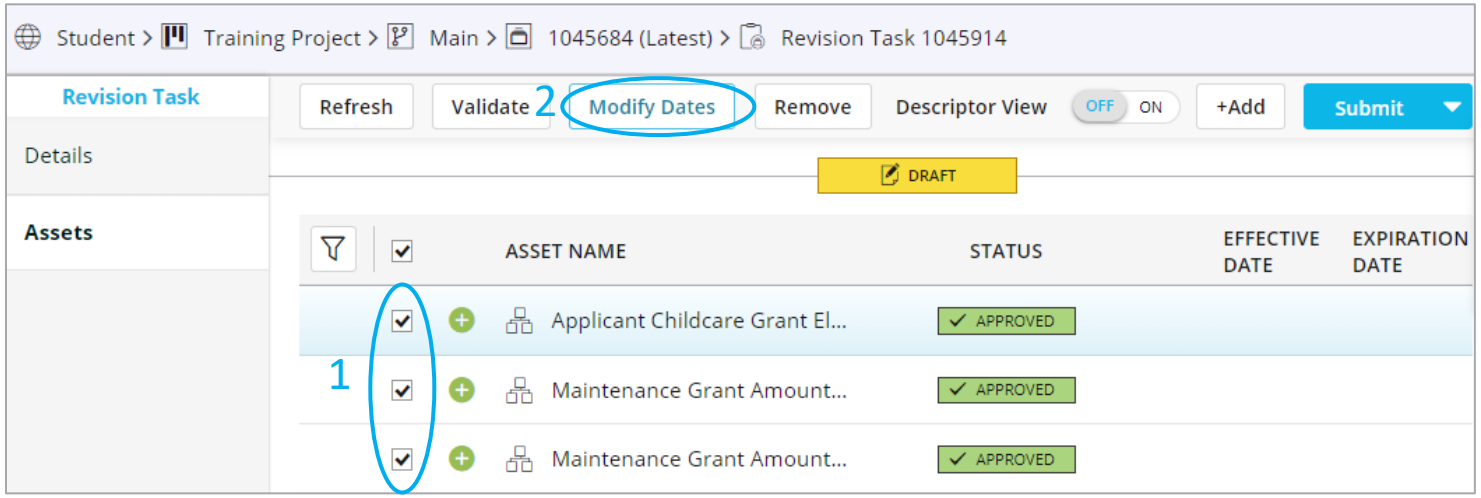
Asset 1, Version 1.2, effective from Dec 1st with no Expiration Date, was added in this Revision Task
The Timeline presents the continuity of the effective versions

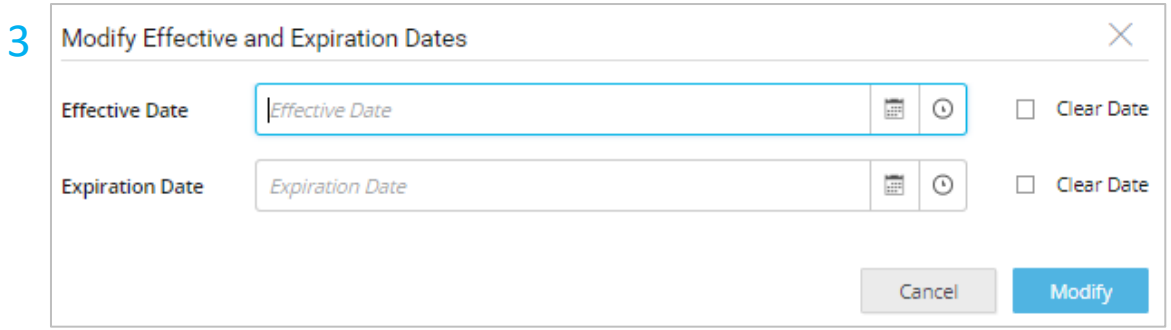


Effective and Expiration Dates Bulk Change

If a bulk change of Effective or Expiration Dates is required:

1. Select the assets that need to be changed
2. Click Modify Dates
3. Select Effective Date and/or Expiration Date and insert the relevant date

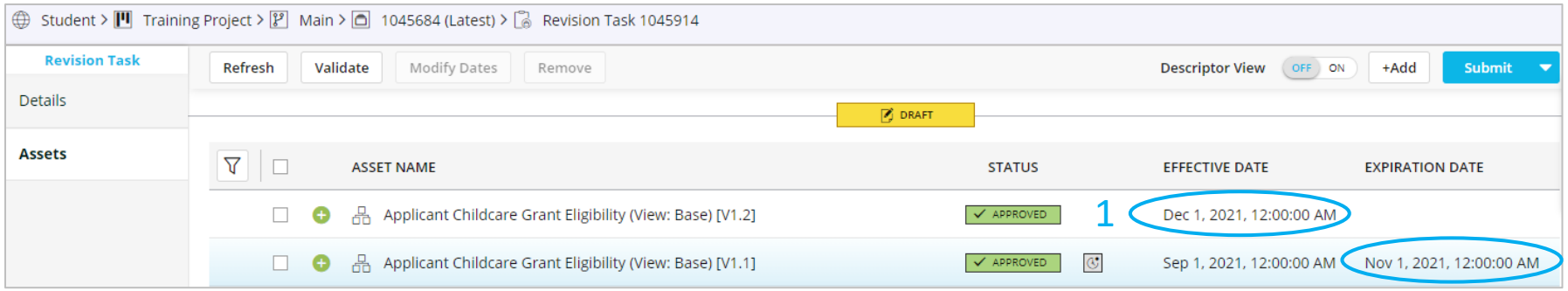


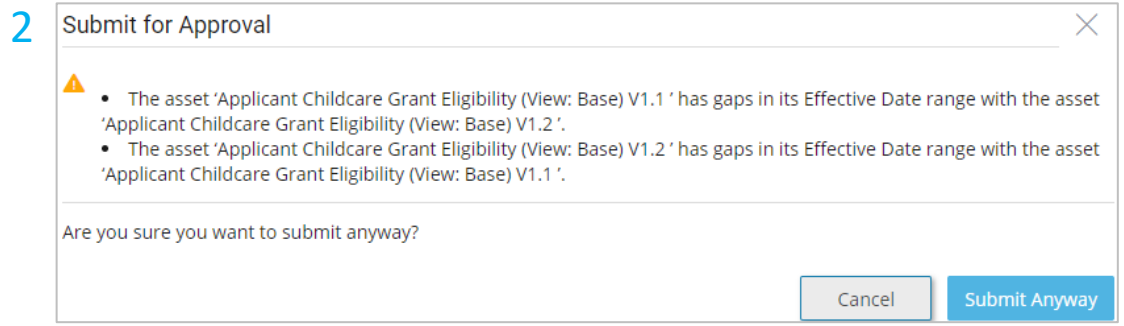


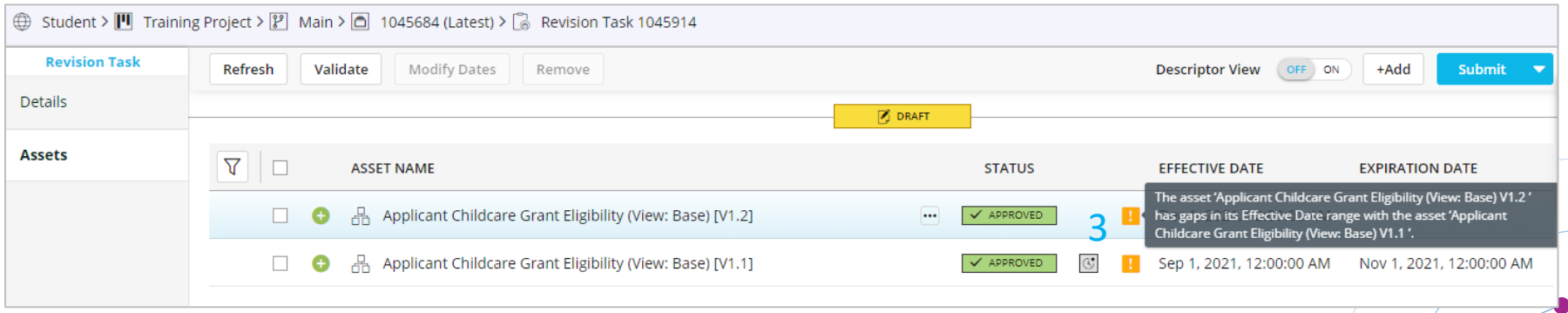
SAPIENS

Effective Date Gaps

1. In case the Effective Date of the next version is after the Expiration Date of the previous version, a gap in the Timeline is created
2. Validation warning message will pop up when submitting the Revision Task
3. Validation warning icon will be displayed

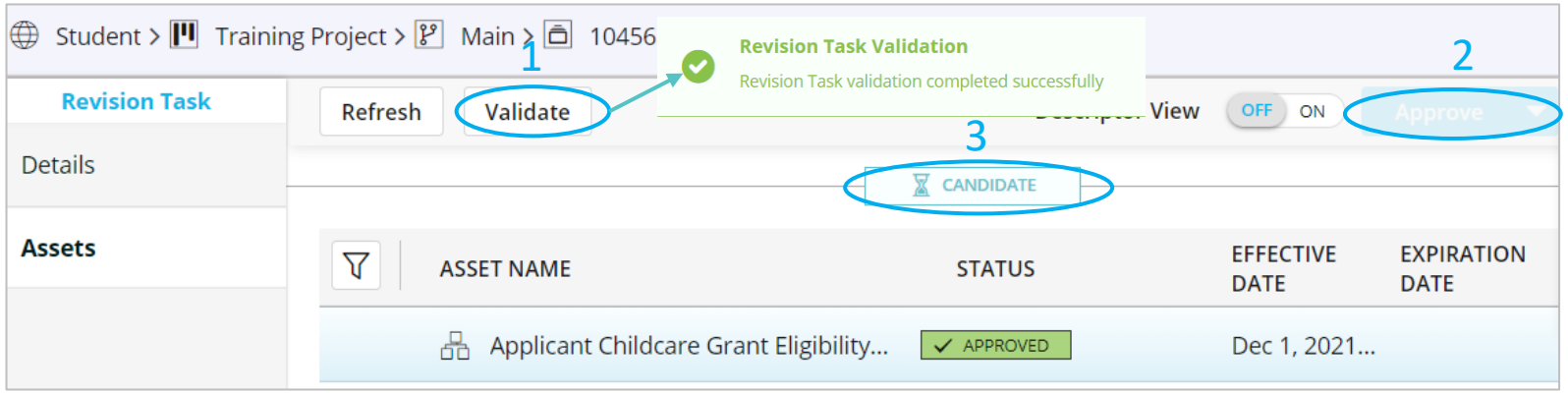


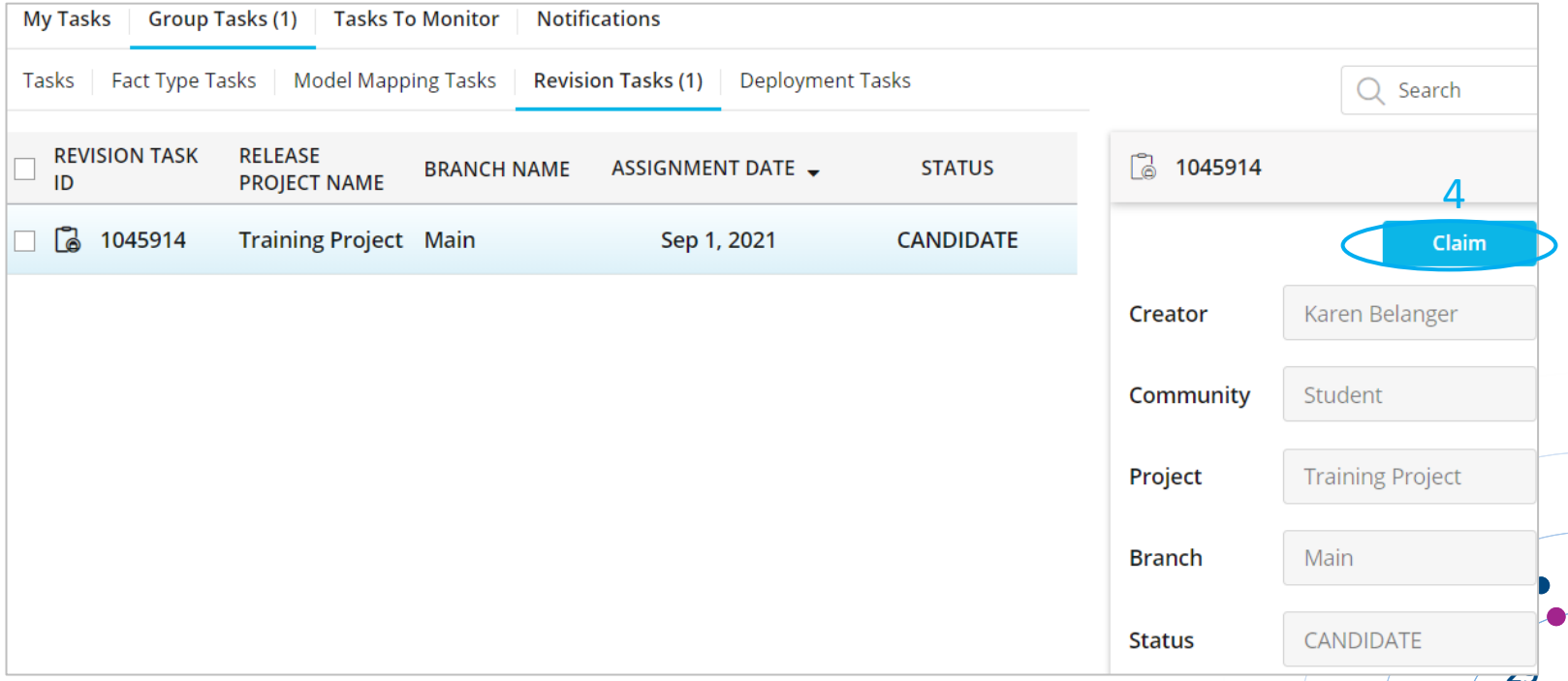




Submit the Revision Task

1. Revision Task can be validated prior to submitting for approval
2. Submitting the Revision Task will prepare it for deployment if Approval Strategy is chosen as ‘None’
3. Submitting the Revision Task will turn it to Candidate Status if Approval Strategy is not chosen as ‘None’
4. The approver must claim and view their assigned task under the Tasks tab > Revision Tasks tab and selecting the Revision Task ID

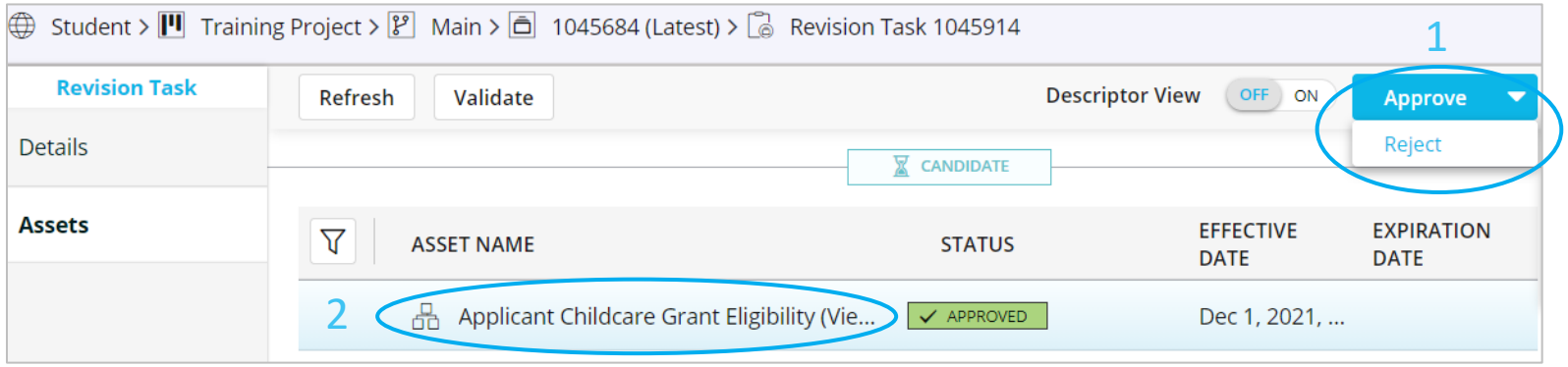


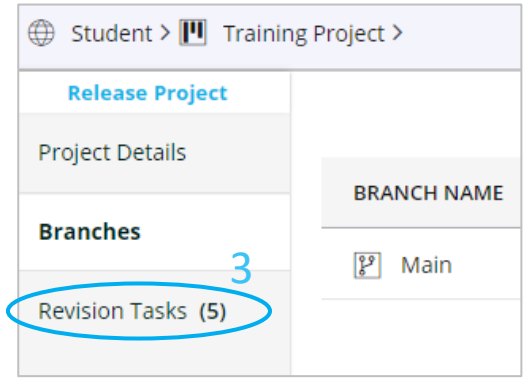


SAPIENS

Approve the Revision Task

1. The submitted Revision Task can be directly Approved or Rejected from the Revision Task by the Owner/s defined under the Project Approval Strategy
   - Requests in Candidate status are not editable
2. The Revision Task content is usually viewed before an Approve/Reject action is taken
   - The included assets can be viewed by clicking on the Asset Name
3. After being claimed, the owner can also access Revision Tasks for review using the Release Project vertical tab on the left





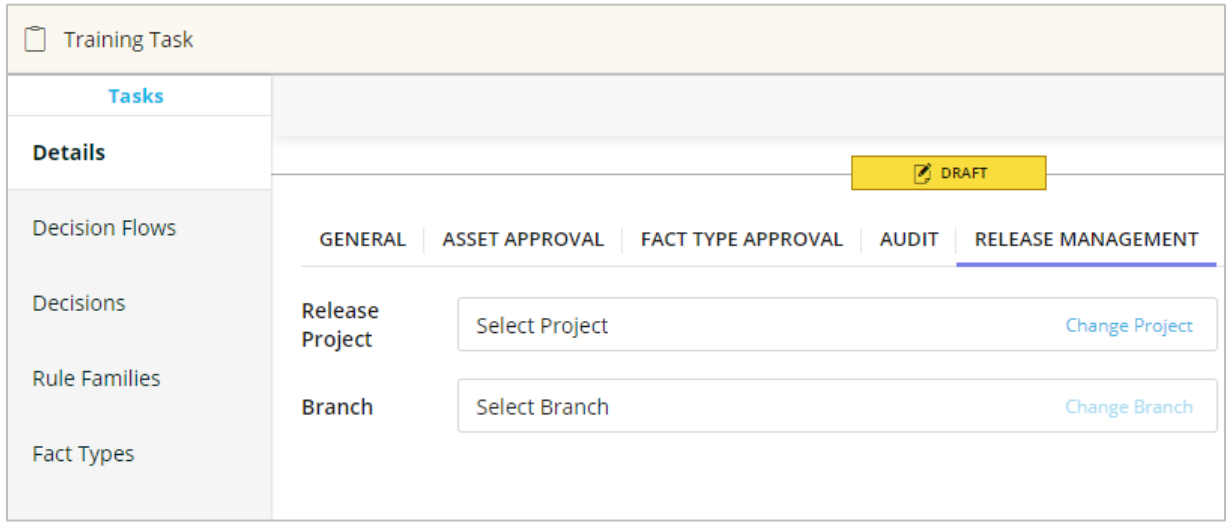
Demo

- Set the Approval Strategy of the Release Project to All and set yourself as owner
- Submit Revision Task 3
- Look at the Task List for the owner task
- Approve the Task

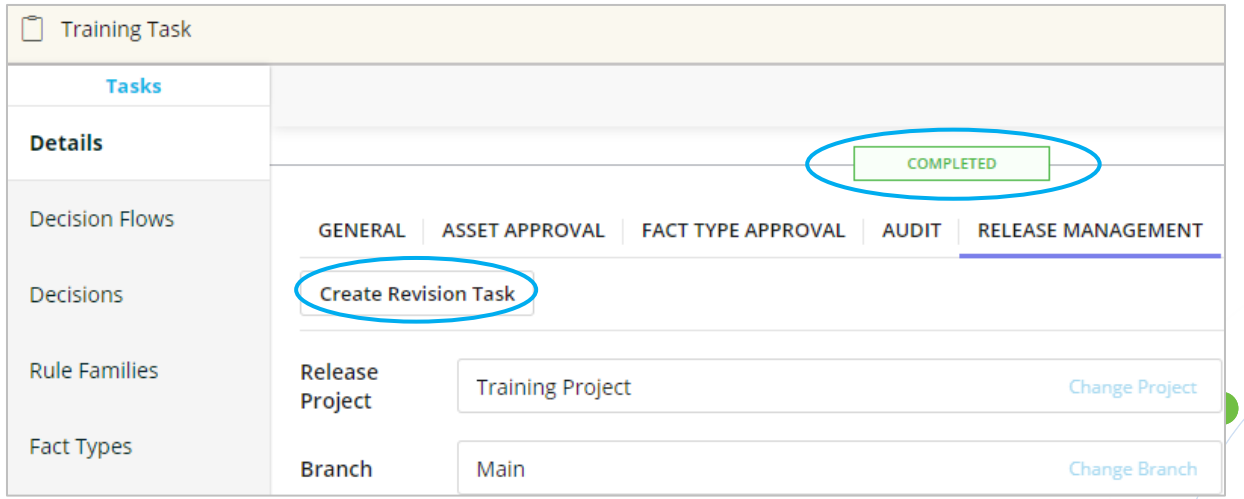
SAPIENS

Add Assets to Revision Task from a Modeling Task

- Another way to create a Revision Task and add assets to it is from within a Modeling Task



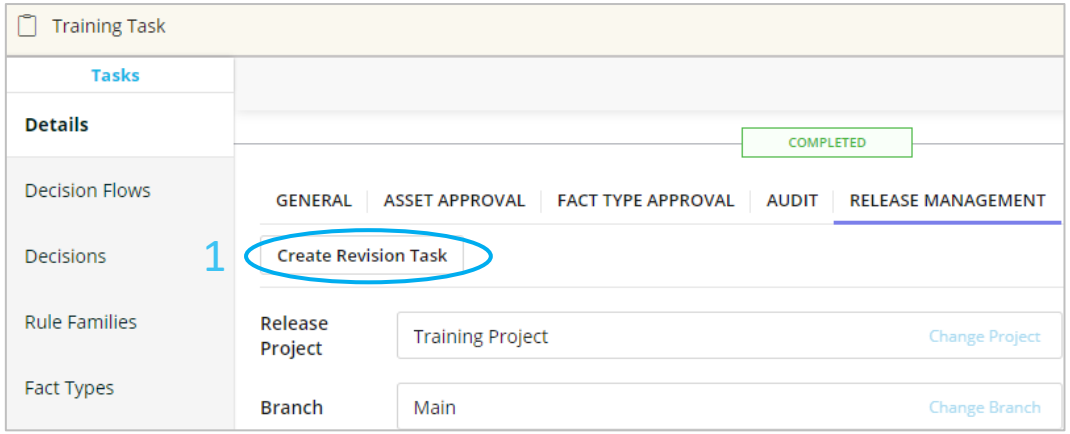
- When a Revision Task is in Candidate, Approved, or Completed status, an option to generate a Revision Task becomes available

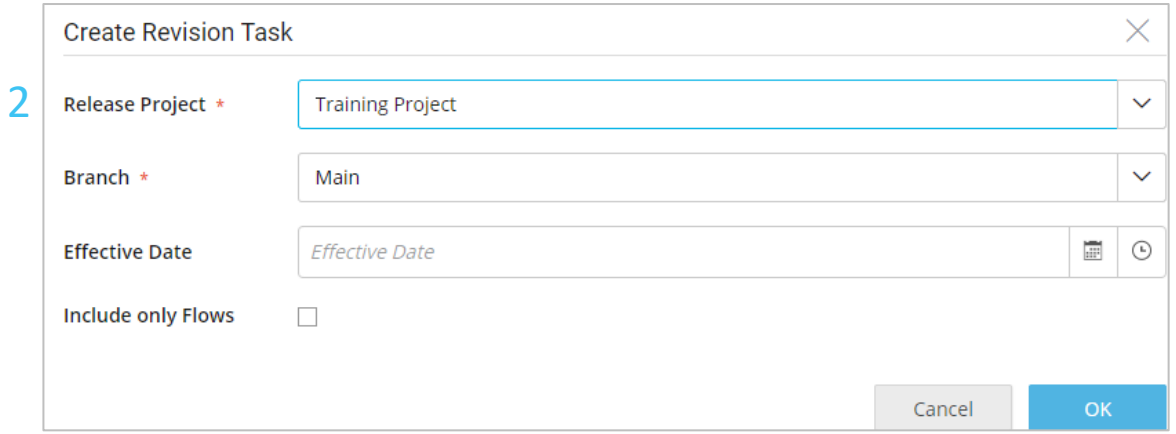


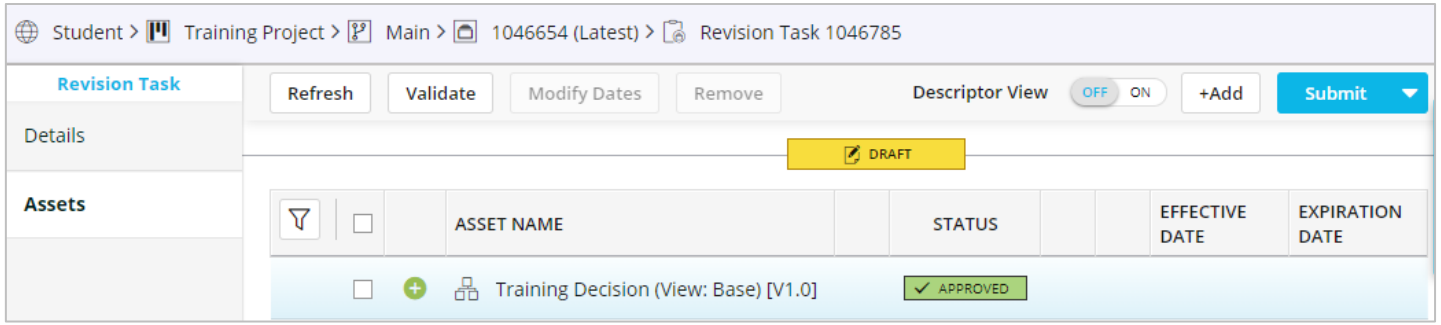
SAPIENS

Add Assets to Revision Task from a Modeling Task

1. Click Create Revision Task
2. Choose Project and Branch and click OK
3. Revision Task is generated

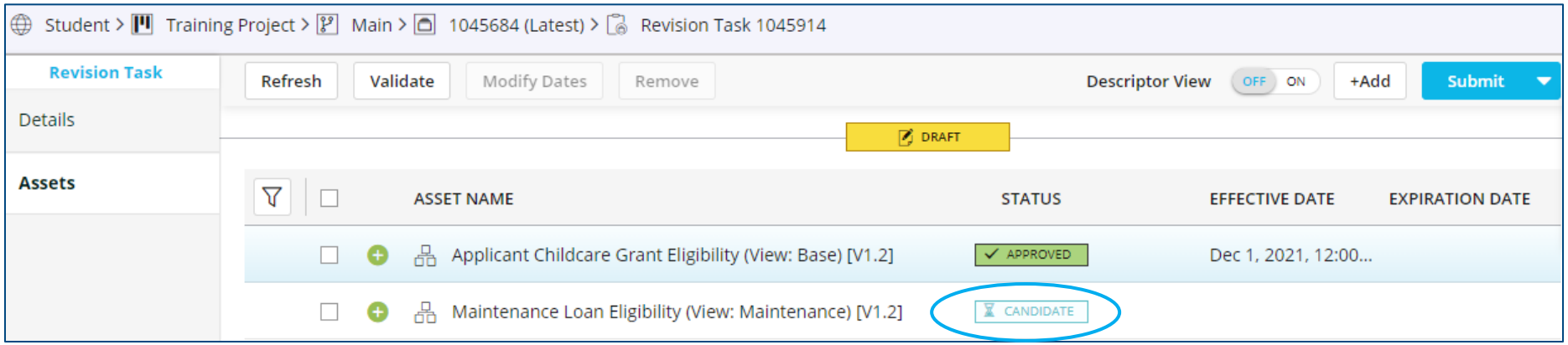






SAPIENS

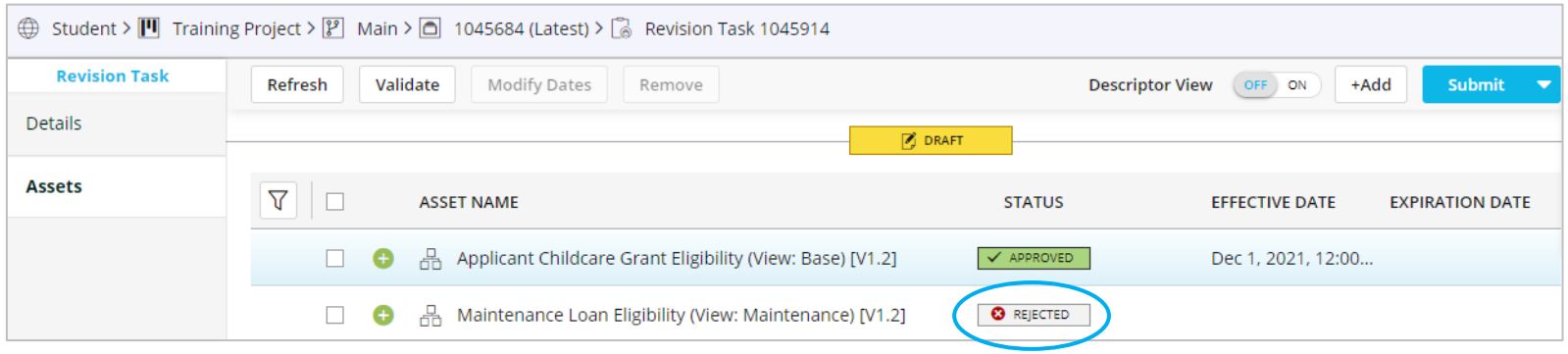
Revision Task with Candidate Assets

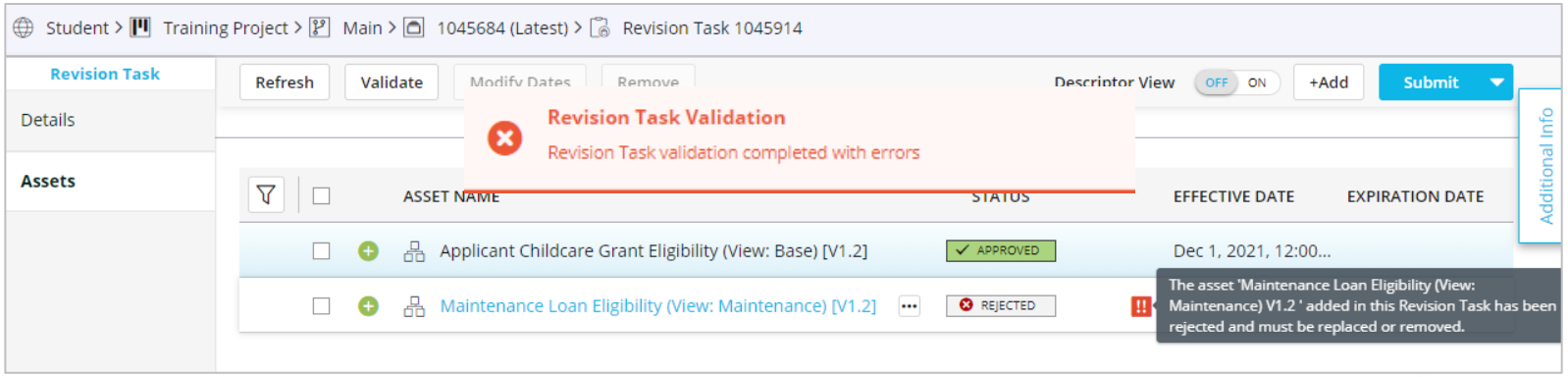


- A Candidate asset can be added to a Revision Task and the request can be submitted
- After submission, the status of the asset may change through the workflow to Rejected or Approved

Revision Task with Candidate Assets

- If an asset is rejected, it is not possible to approve the Revision Task and a validation error icon will be displayed
- The requestor should change the assets and resubmit
- If the rejected asset was resubmitted unchanged, then it becomes a Candidate again (or Approved) and the Revision Task can be resubmitted
- If the rejected asset was changed and resubmitted, then the rejected asset should be removed from the Revision Task and new asset can be added instead

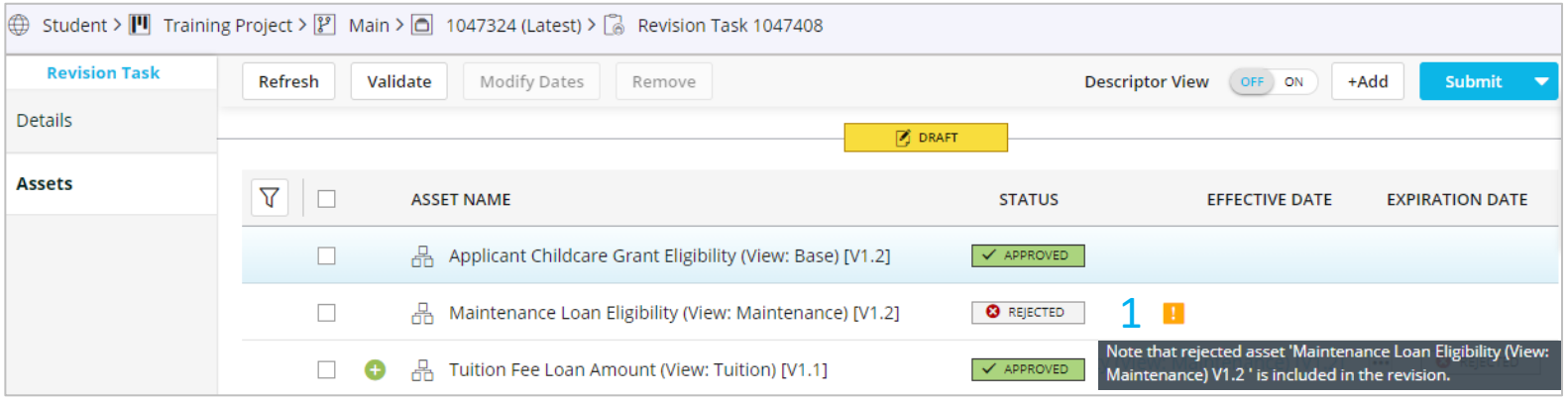




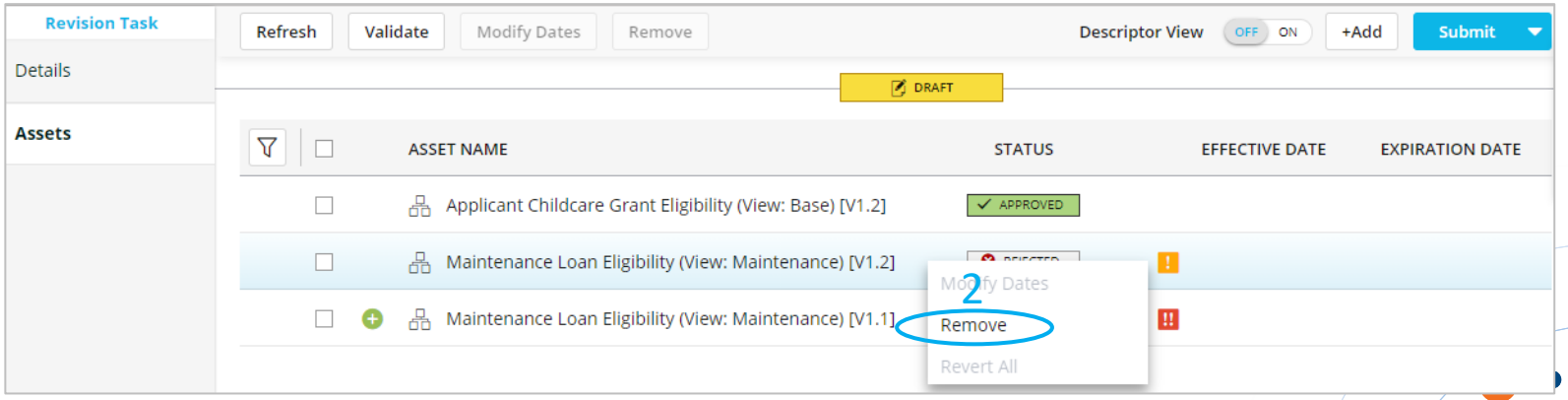
Revision Task with Candidate Assets

If there is a rejected asset in a Revision:

1. Adding another asset (which is not a version of the rejected asset) and submitting is supported and there will be a validation warning



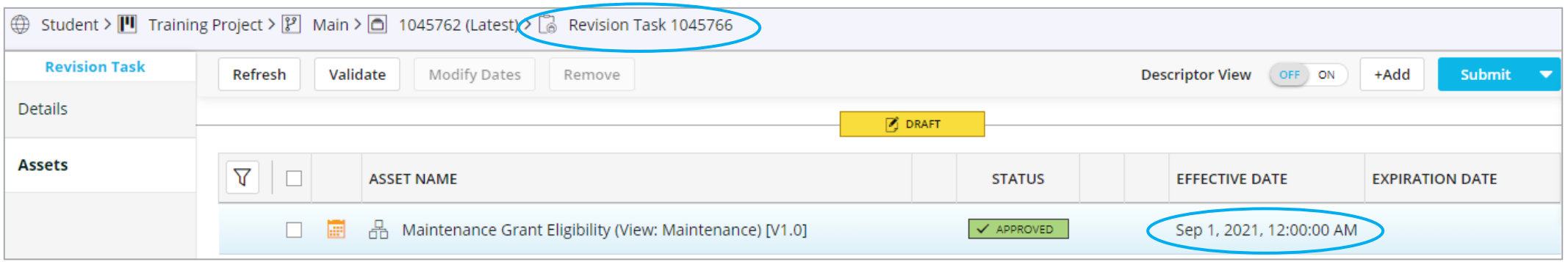
2. If adding another version of the rejected item - the rejected item must be removed before submission



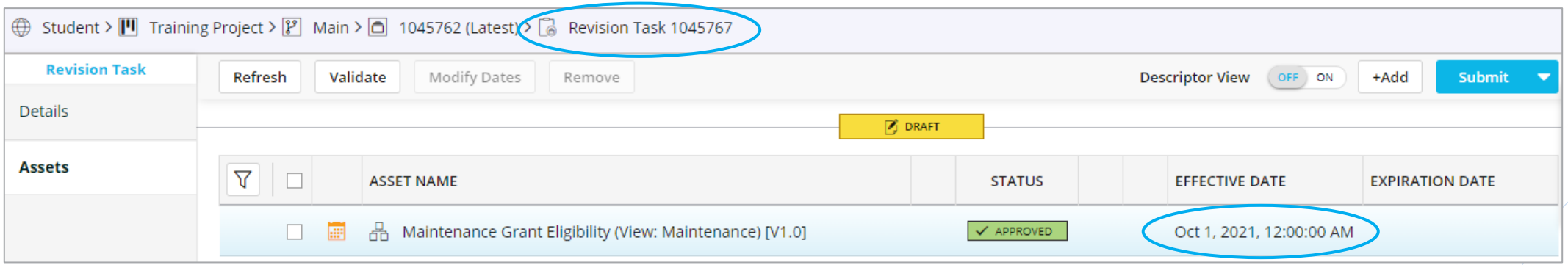
Revision Task Conflict

- In case two Revision Tasks are submitted in parallel and both change the Effective/Expiration Dates of the same asset, a conflict will occur
- The conflict occurs regardless of which request was submitted first

Revision Task created to modify Asset with Effective Date of Sep 1st with no Expiration Date

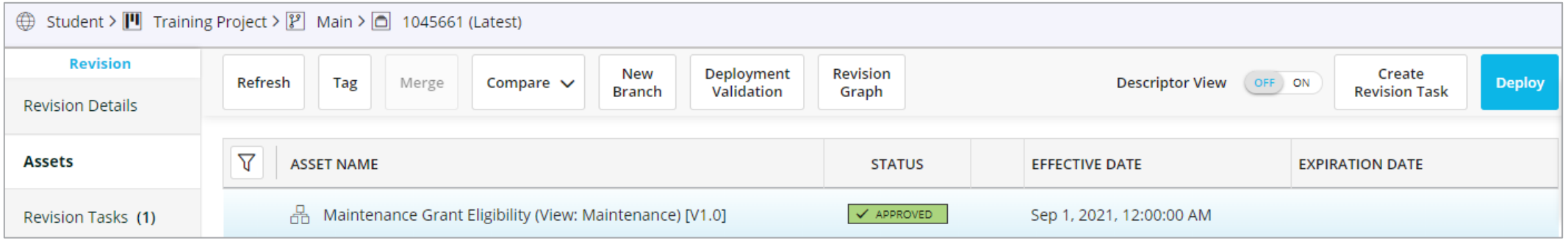


New Revision Task created to modify same Asset with Effective Date of Oct 1st with no Expiration Date

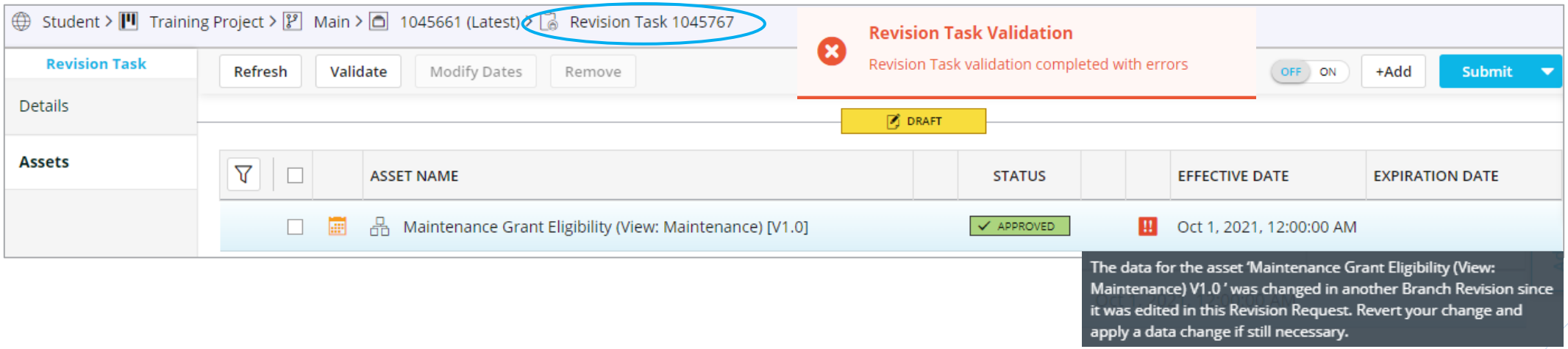


Revision Task Conflict

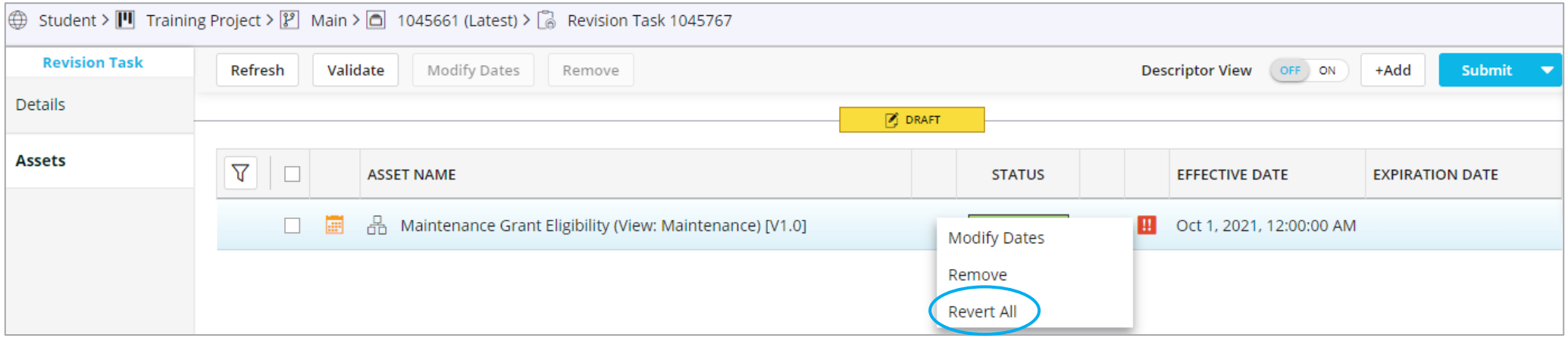
The first Revision Task was submitted with Effective Date of Sep 1st



When trying to submit the second Revision Task with Effective Date of Oct 1st, a Validation Error occurred



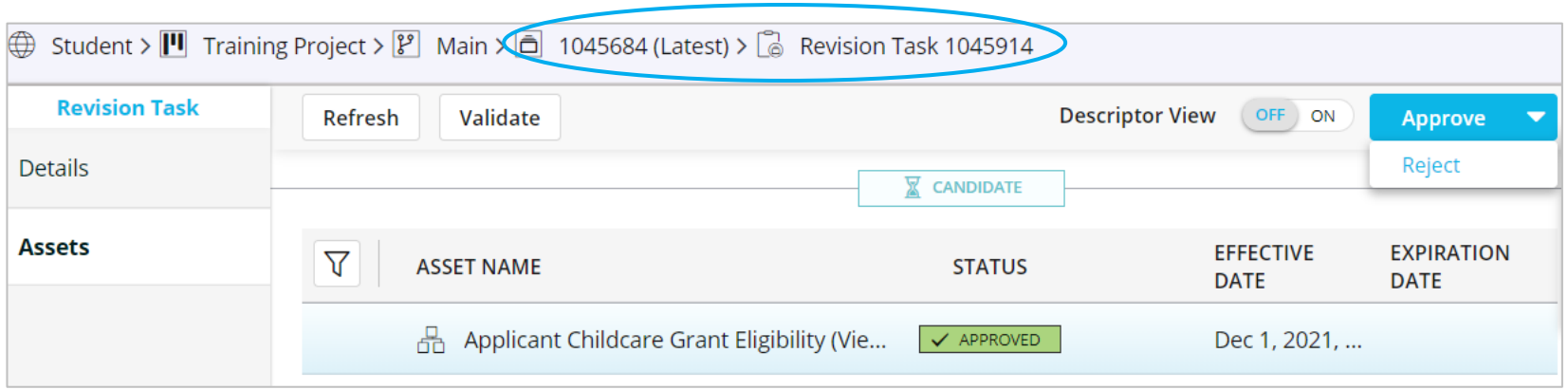
Revision Task Conflict

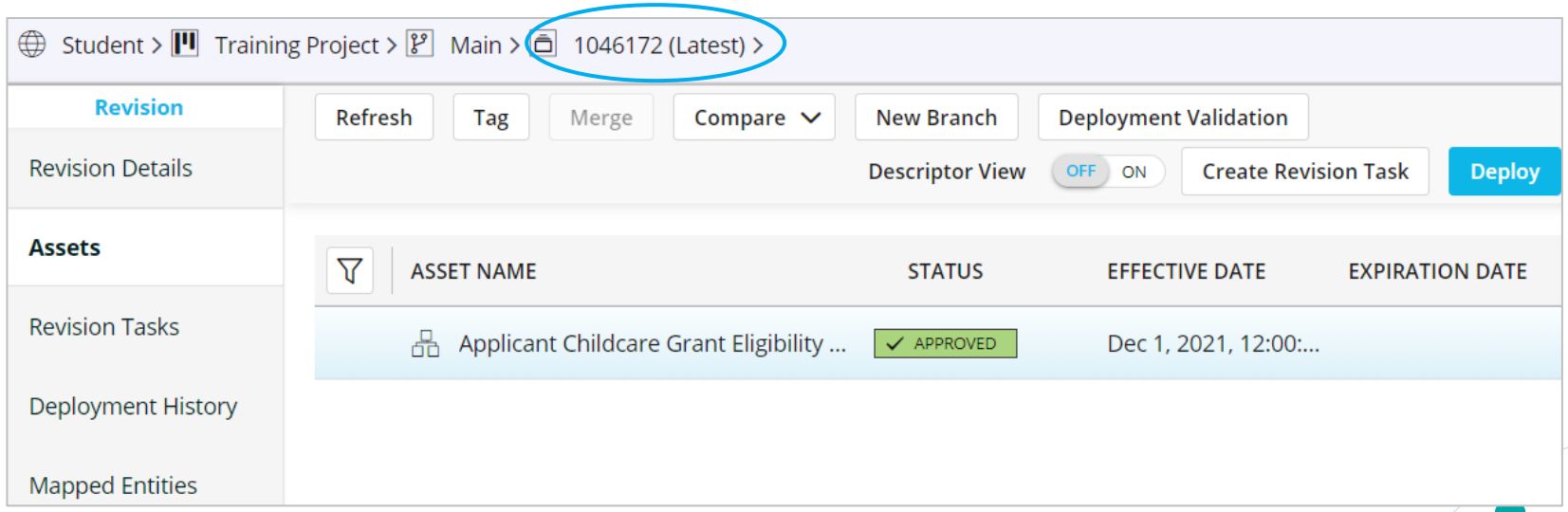


- The second revision task must be modified, by **reverting** and updating as needed
- The **revert** action:
  - Undoes the requested asset date change
  - Refreshes the asset dates from the latest approved Revision Task (= latest Revision)
- After clicking 'Revert' it is possible to edit the Revision Task, if still necessary

Branch – Revision

- Once the Revision Task is approved, a new Revision is created
- Revision content is not editable
  - Only Revision custom properties are editable

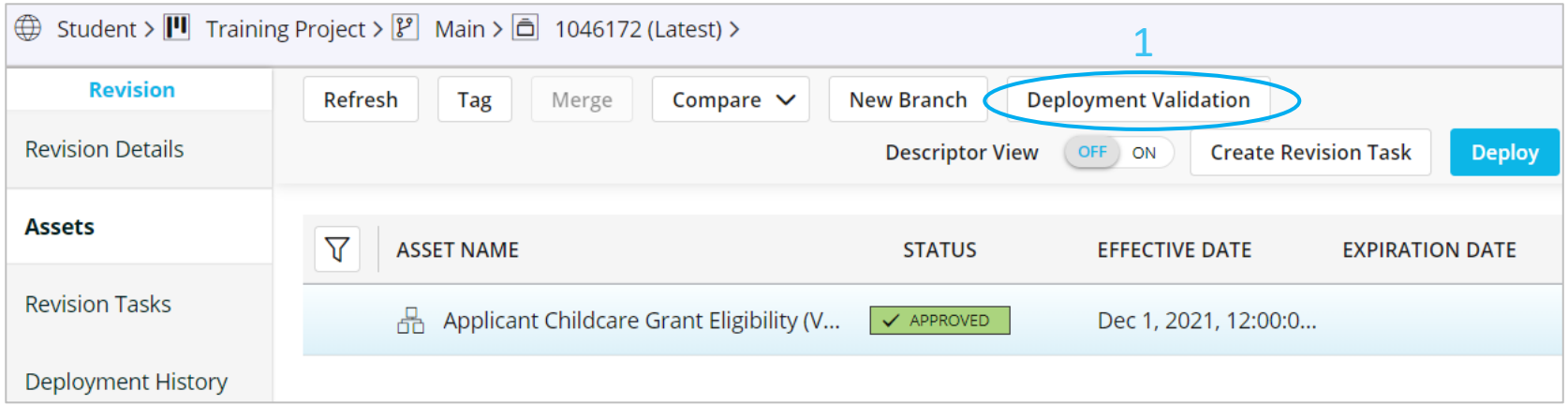


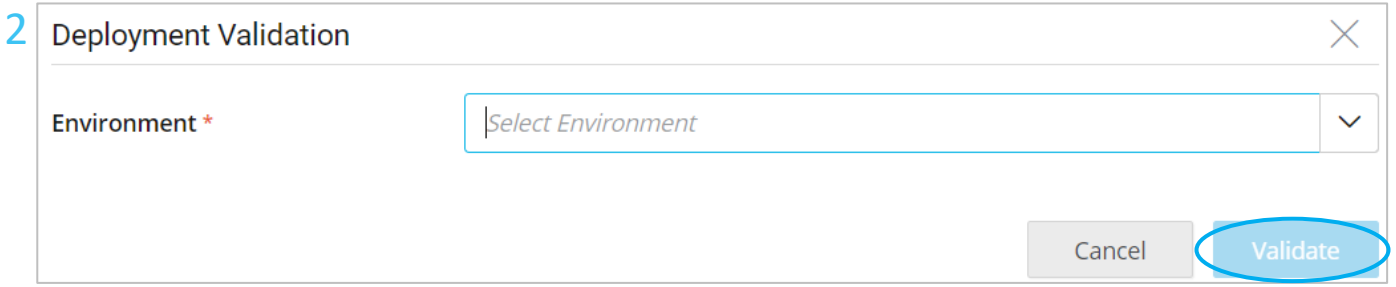


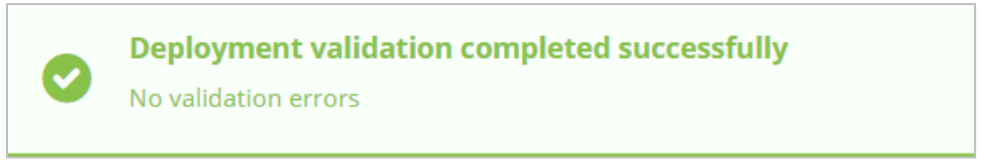
Deploy Revision – Validation

A Revision can be validated prior to submission to confirm the status of its assets and ensure there are no conflicts

1. A Revision can be validated prior to deployment by clicking Deployment Validation option from the menu
2. Select Environment from the list and click Validate

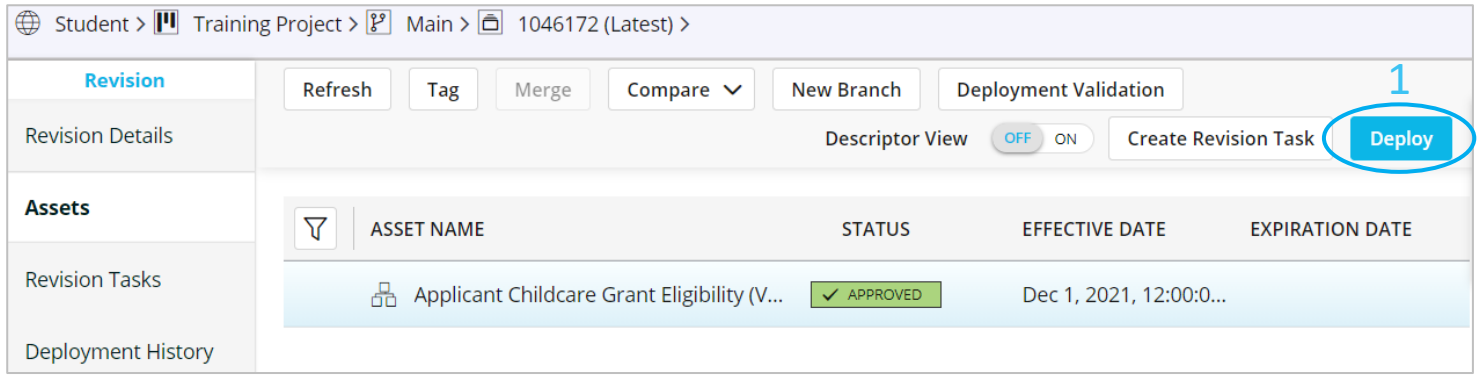


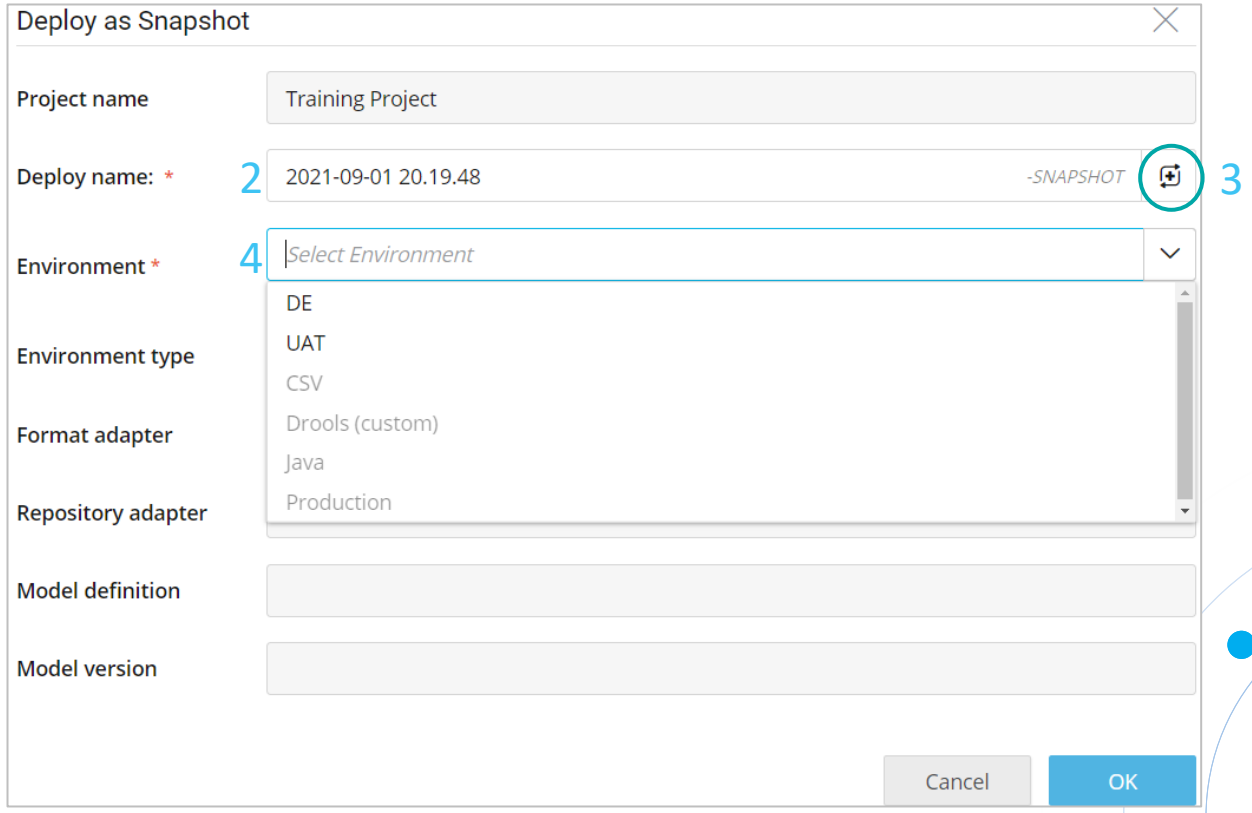




Deploy Revision – Snapshot

1. To Deploy the Revision, click **Deploy** button from the menu
2. To change the Revision's name from Revision timestamp to a meaningful name, type the name in the **Snapshot** Tag text box
3. Click Snapshot icon to receive a Date & Time Tag
   - The Tag does not set the name of the Revision - this is just for deployment
4. Select an Environment from the list
   - Snapshot deployments are available only for environments which allow snapshots



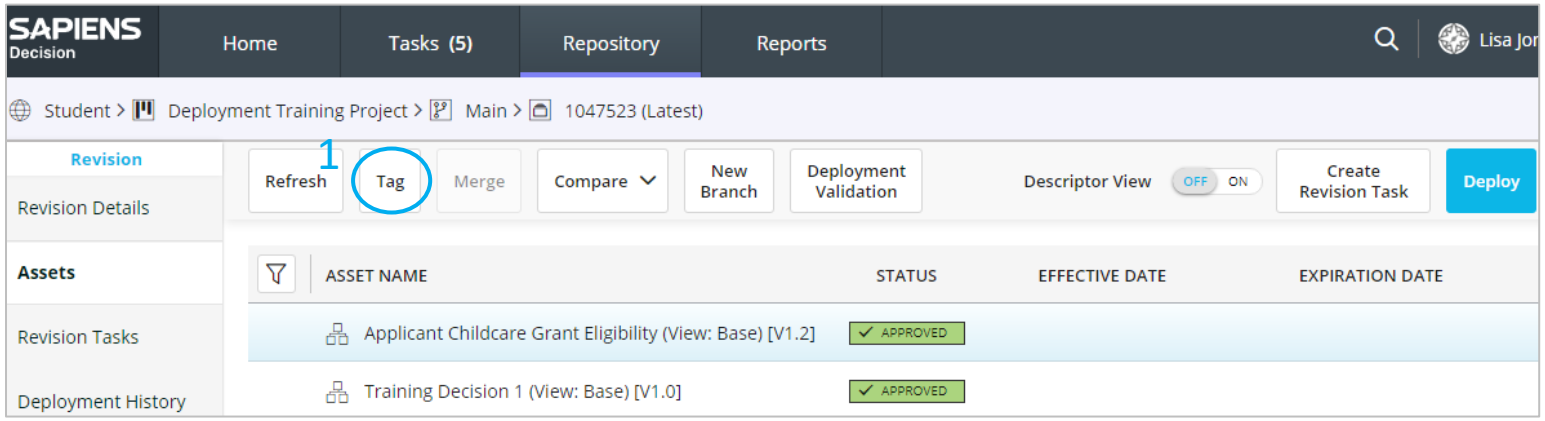


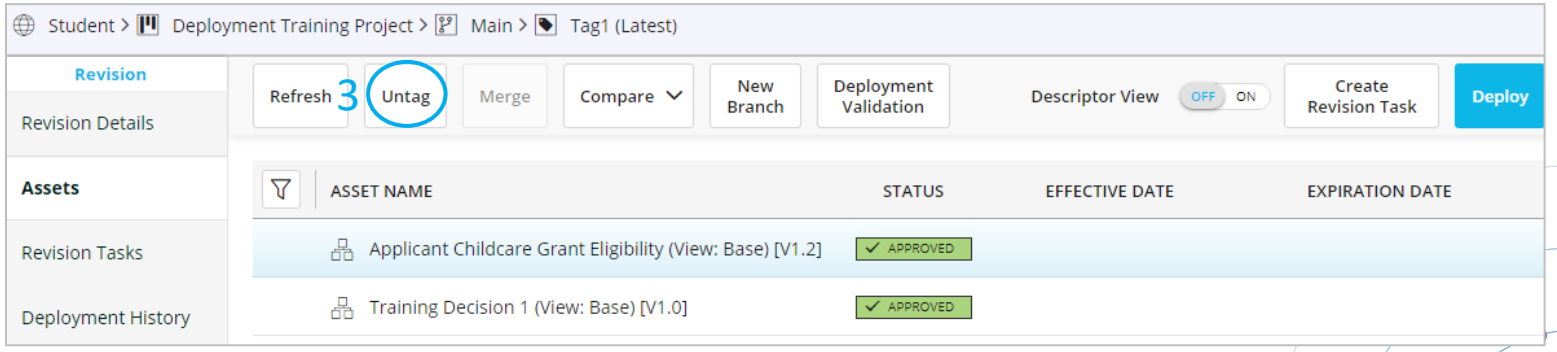
SAPIENS

Tag

- You can change a Revision's number to a meaningful name through a **Tag**
- Tags are Revisions that cannot be modified once created
- **Tags** are typically used for final deployment packages while **Snapshots** are typically used for testing purposes of assets still under development
- A Tag name must be unique within a Project

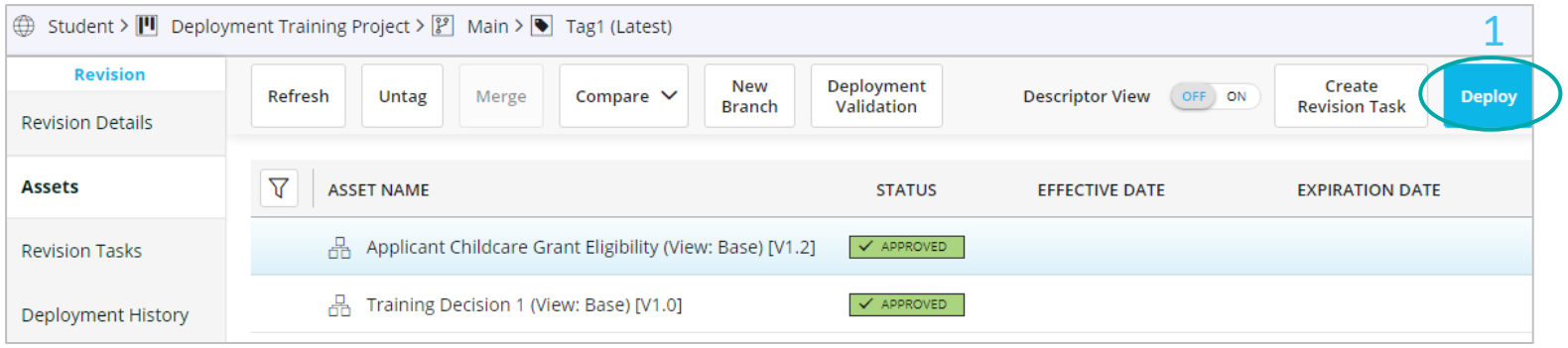
1. Click Tag button from menu
2. Insert Tag Name
3. It is possible to 'Untag' the revision by clicking Untag button

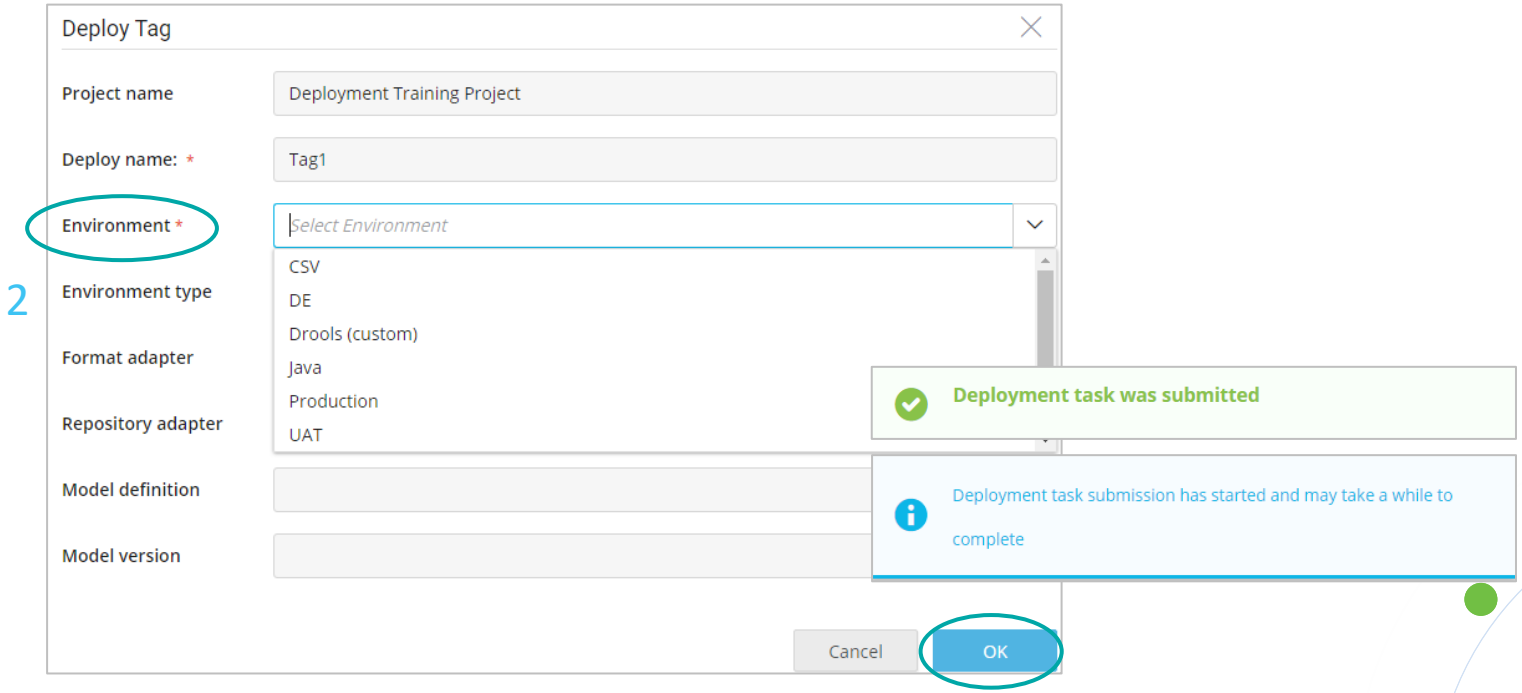




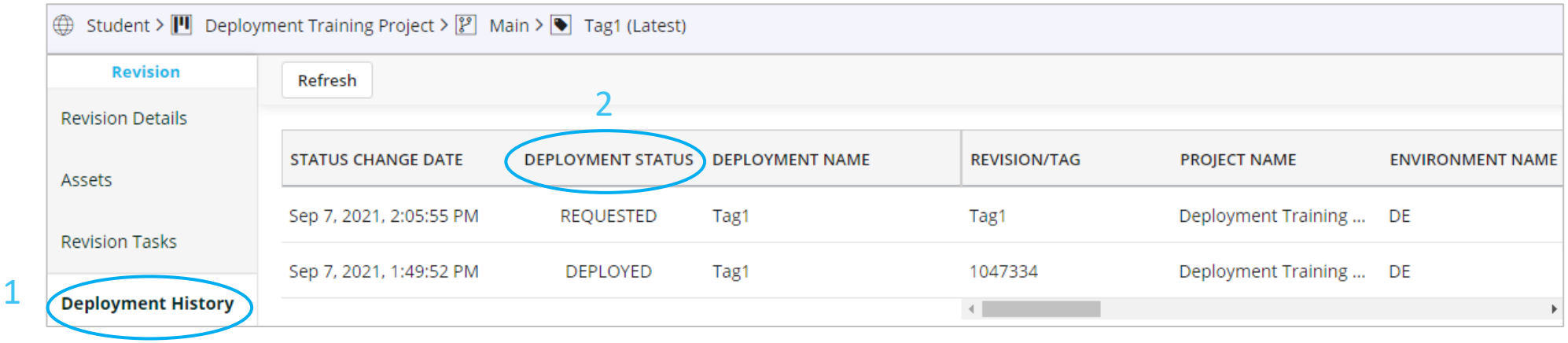
Deploy Revision – Tag

1. Click **Deploy** button
2. Select Environment from the list and click **OK**





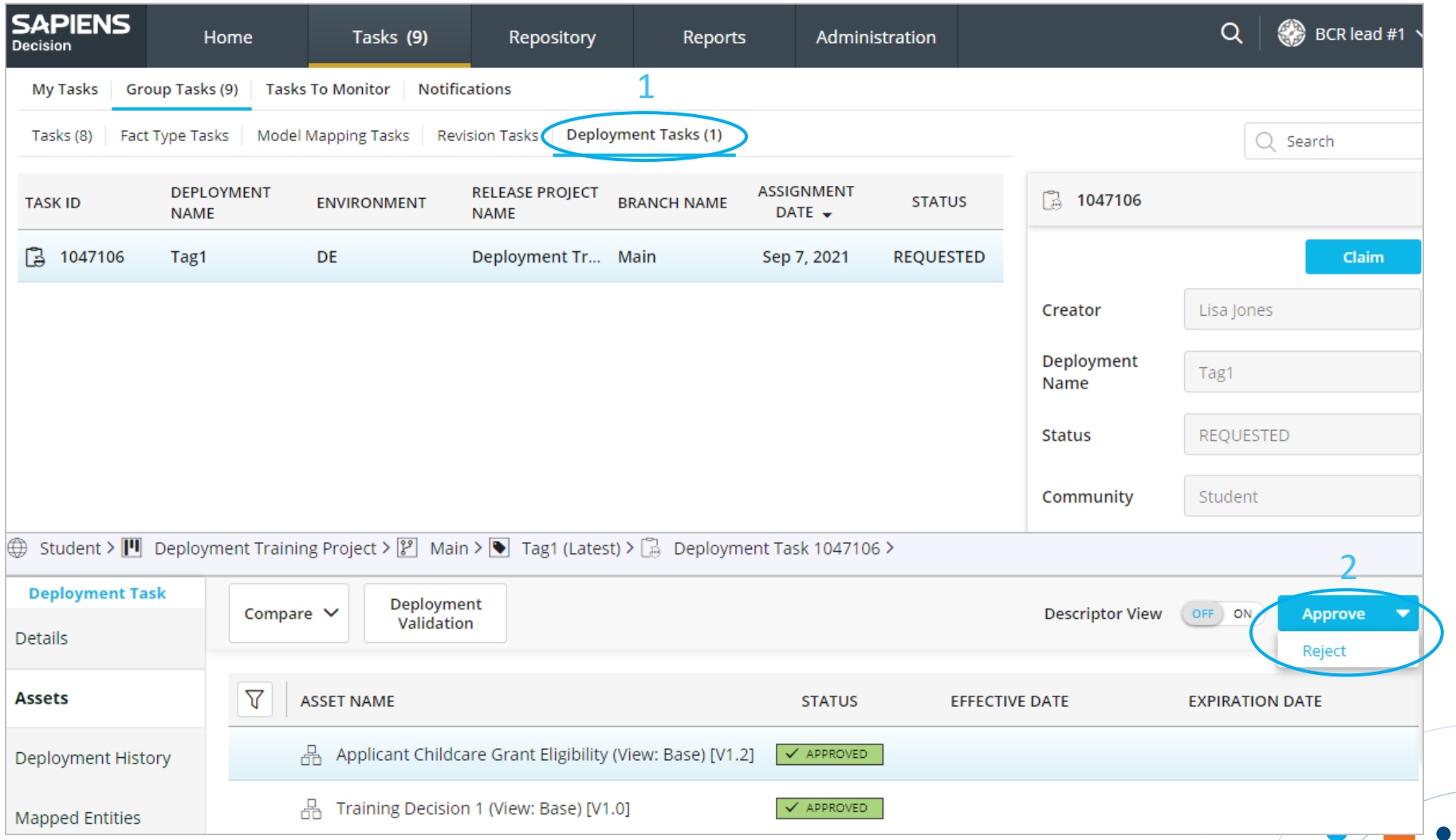
Deployment Status



1. To see deployment request status, select Deployment History from the Revision tab
2. Approval Strategy for every environment is defined in the Community Admin
   - When the deployment is approved, the status will be changed to ‘Deployed’

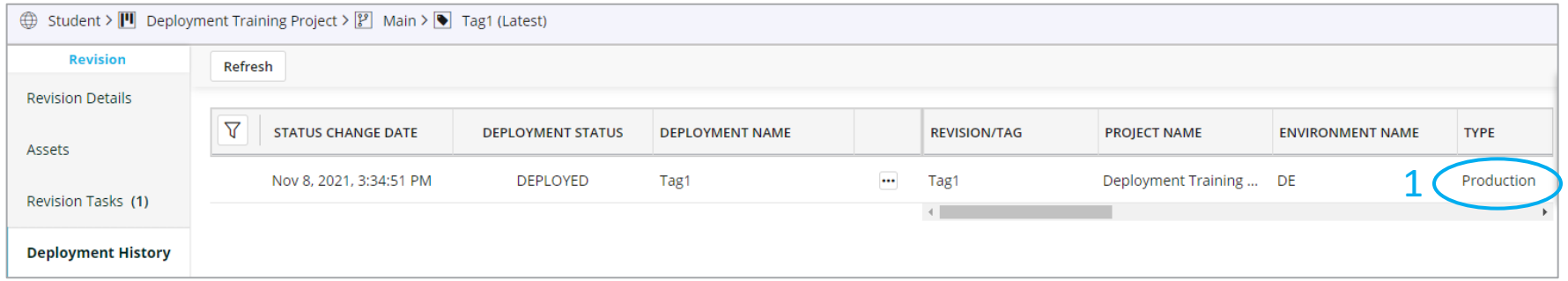
Approve/Reject Deploy Request

1. To approve: the owner can access the Deployment Task from their Task List in Deployment Tasks
2. The Owner(s) can Approve or Reject the Deployment Request

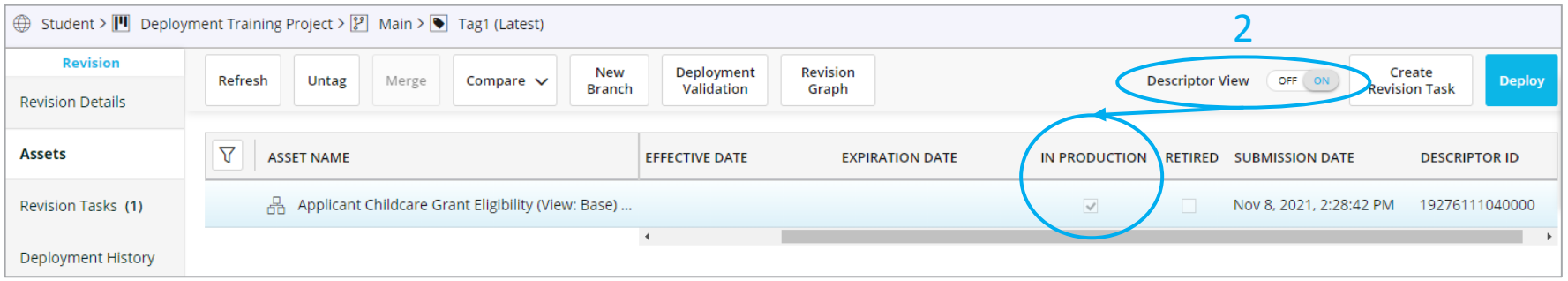


Deployment Environment – Production

1. The Tag was deployed to the environment 'Decision Execution' which is defined as a production environment (viewed in Deployment History on Revision tab)



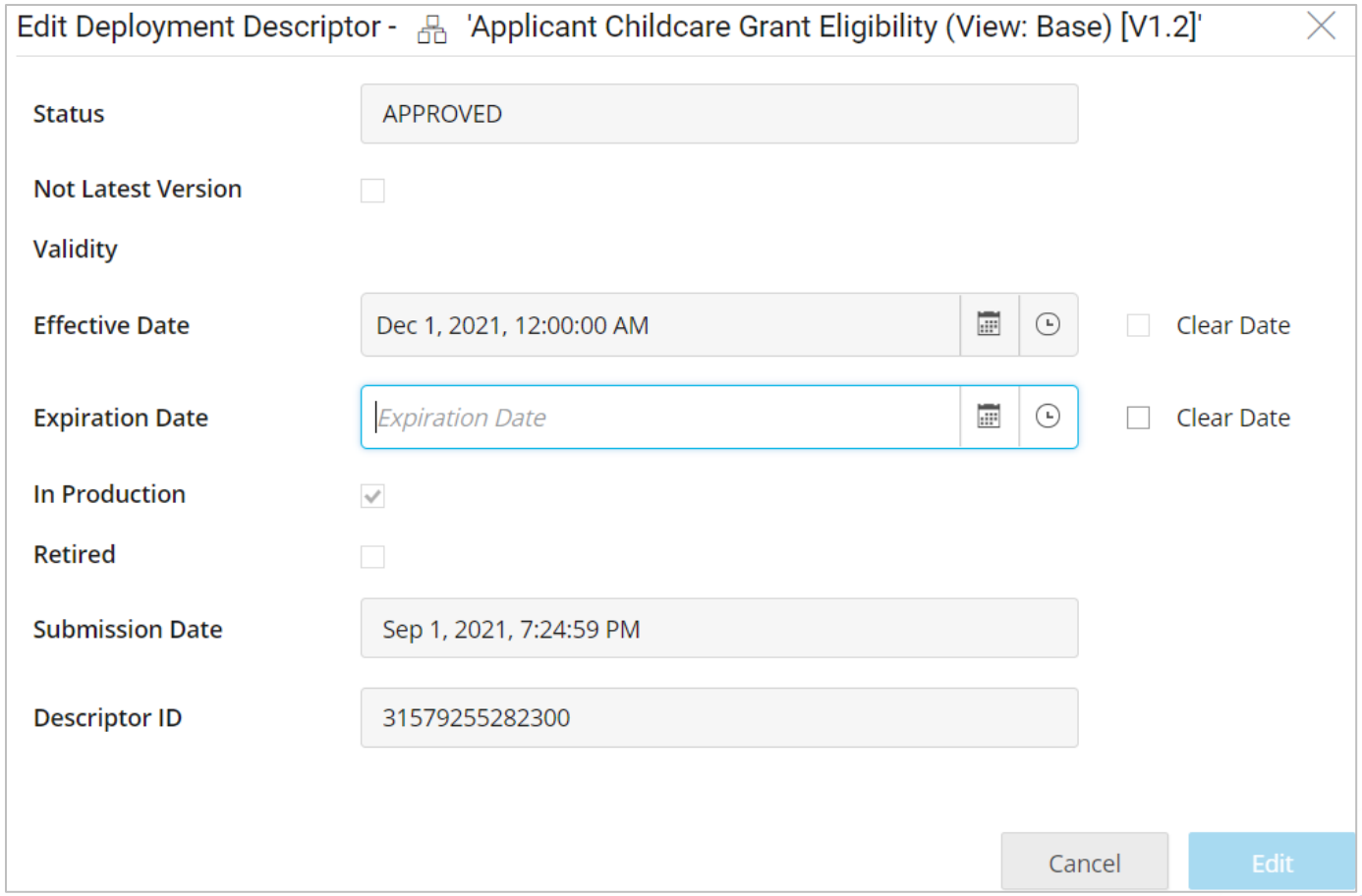
2. Open Descriptor View and the asset in the Tag is now marked as 'In Production'



SAPIENS

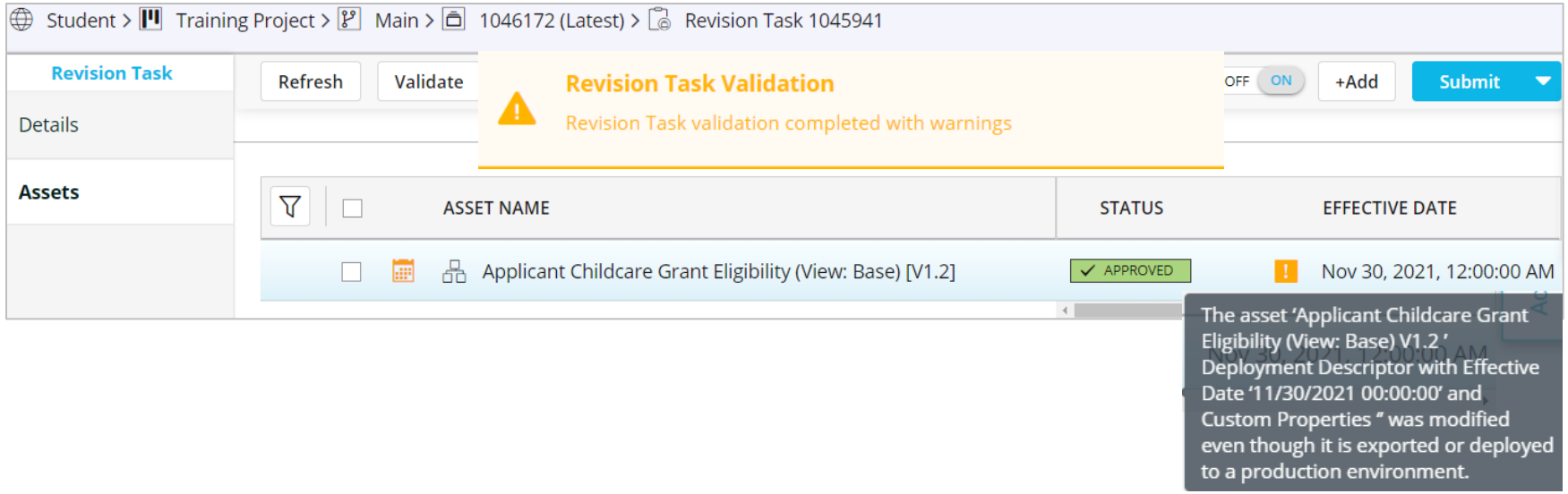
In Production Assets and Effective Dates

- Changing of Effective Dates of assets that are in production is restricted to users with special permissions
- By default, a user can only set the Expiration Date to a future date or null (meaning unrestricted)
- Permissions may be changed to allow Effective Date changes in the Administration Tab



In Production Assets and Effective Dates

- With appropriate permissions, a Validation Warning will occur when an in-production asset's Effective Date is changed



Demo

- Check Allow Snapshots on Environment Management screen
- Deploy Tag1
- Deploy latest Revision as Snapshot

SAPIENS

Deployment History – Statuses

Student > Deployment Training Project > Main > Tag5 (Latest)

Revision

Revision Details

Assets

Revision Tasks (1)

Deployment History

Mapped Entities

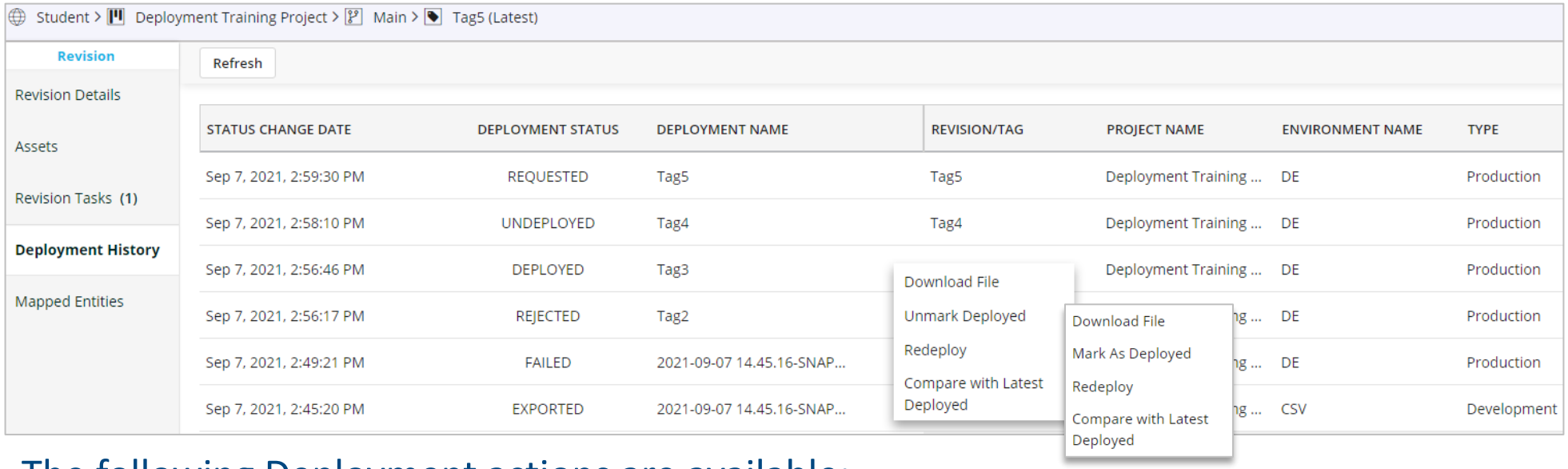
Refresh

STATUS CHANGE DATE	DEPLOYMENT STATUS	DEPLOYMENT NAME	REVISION/TAG	PROJECT NAME	ENVIRONMENT NAME	TYPE
Sep 7, 2021, 2:59:30 PM	REQUESTED	Tag5	Tag5	Deployment Training ...	DE	Production
Sep 7, 2021, 2:58:10 PM	UNDEPLOYED	Tag4	Tag4	Deployment Training ...	DE	Production
Sep 7, 2021, 2:56:46 PM	DEPLOYED	Tag3	Tag3	Deployment Training ...	DE	Production
Sep 7, 2021, 2:56:17 PM	REJECTED	Tag2	Tag2	Deployment Training ...	DE	Production
Sep 7, 2021, 2:49:21 PM	FAILED	2021-09-07 14.45.16-SNAP...	1047593	Deployment Training ...	DE	Production
Sep 7, 2021, 2:45:20 PM	EXPORTED	2021-09-07 14.45.16-SNAP...	1047598	Deployment Training ...	CSV	Development

The following Deployment statuses are available:

- Requested: Deployment request is waiting for approval
- Exported: Revision successfully exported
- Failed: Revision failed to deploy
- Rejected: Deployment request was rejected by Approver
- Deployed: Status can be inserted manually (see next slide) or automatically (configurable in the Repository Definition in Environment Management)
- Undeployed: Status is inserted manually (see next slide)

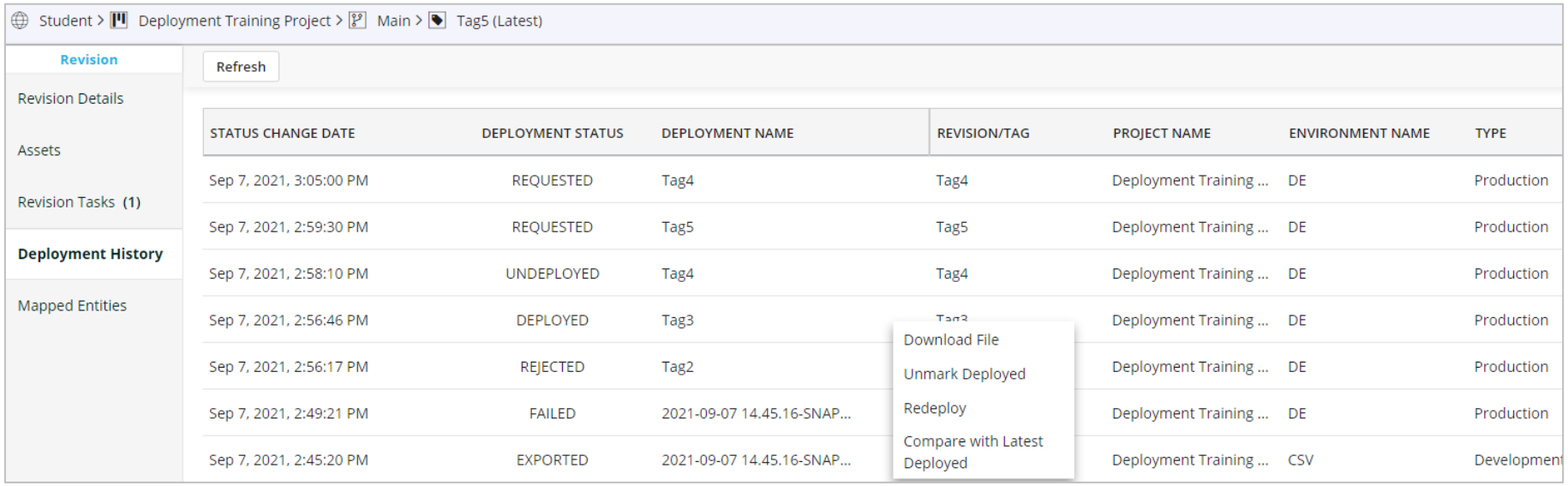
Deployment History – Actions



The following Deployment actions are available:

- Download File: Download the file
- Mark As Deployed: Mark the exported Revision as Deployed
- Unmark Deployed: Option replaces ‘Mark As Deployed’ when status is “Deployed”
- Redeploy: Redeploy package to another compatible environment or the same environment (re-deployment)
- Compare with Latest Deployed: compare highlighted deployed Revision with the latest deployed Revision to the same environment

Deployment History – Redeployment



- Redeploy uses the already generated deploy package but with updated Release Information XML (Deploy Date, Environment Information and Deploy Owner Approvals)
- The target environment settings should be the same as the source environment settings (same Model Mapping, same Format definition)
- The target environment should have a repository definition
- The Redeploy action creates a new row in the deployment history
- Deploy to new environment will follow the new environment's Approval Strategy

SAPIENS

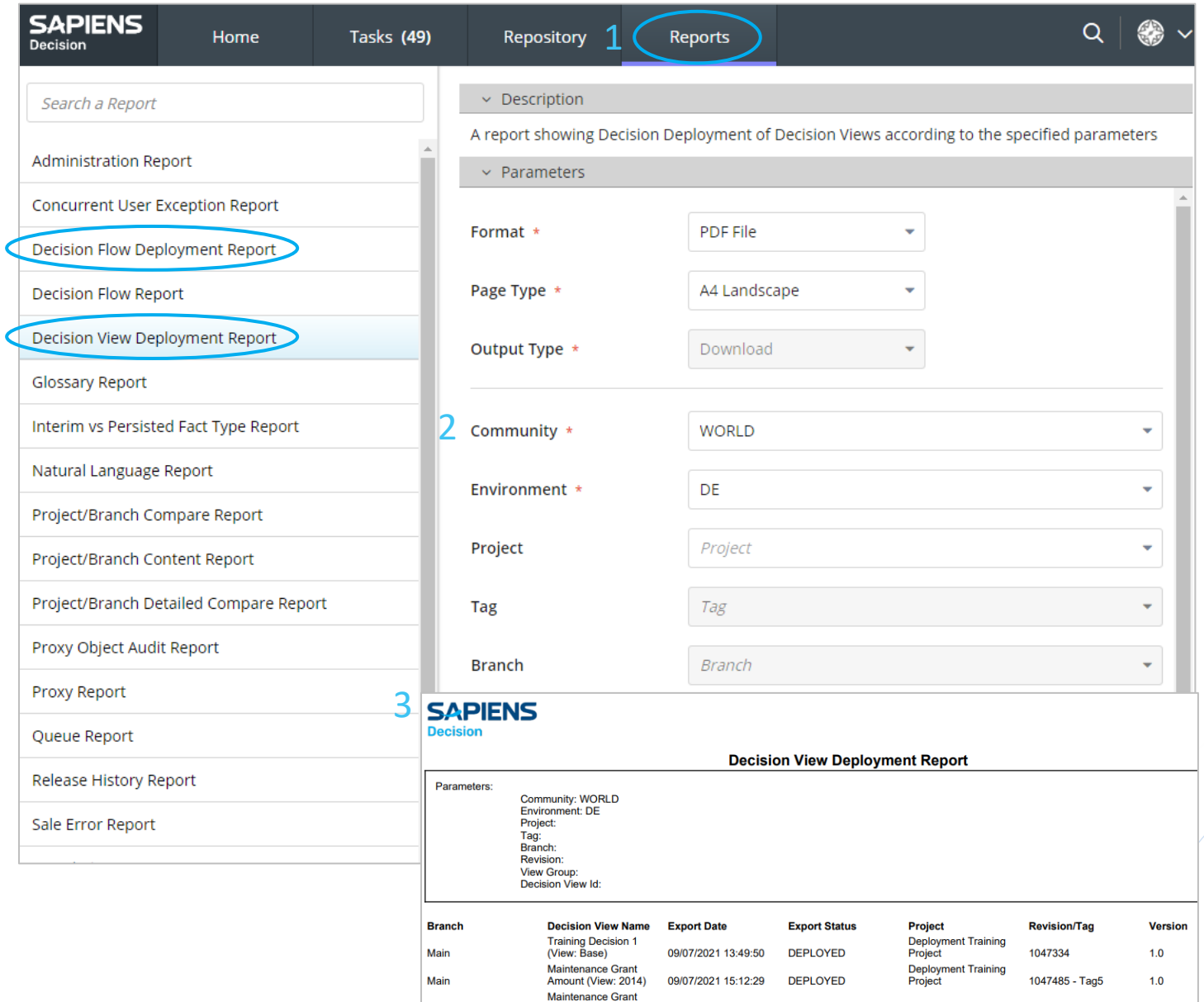
Demo

- View Deployment History from current Branch
- Mark Tag1 as “Deployed”
- Show Additional Info for asset A
  - Show “Deployments” then “Environments”
- Click “More…” to view the full details of a deployment

SAPIENS

Deployment History from Reports Tab

1. It is possible to access a history of Deployments from the Reports tab
2. Select Community, Environment, Project, and Tag from the list
3. All deployments from this Environment will be in this report

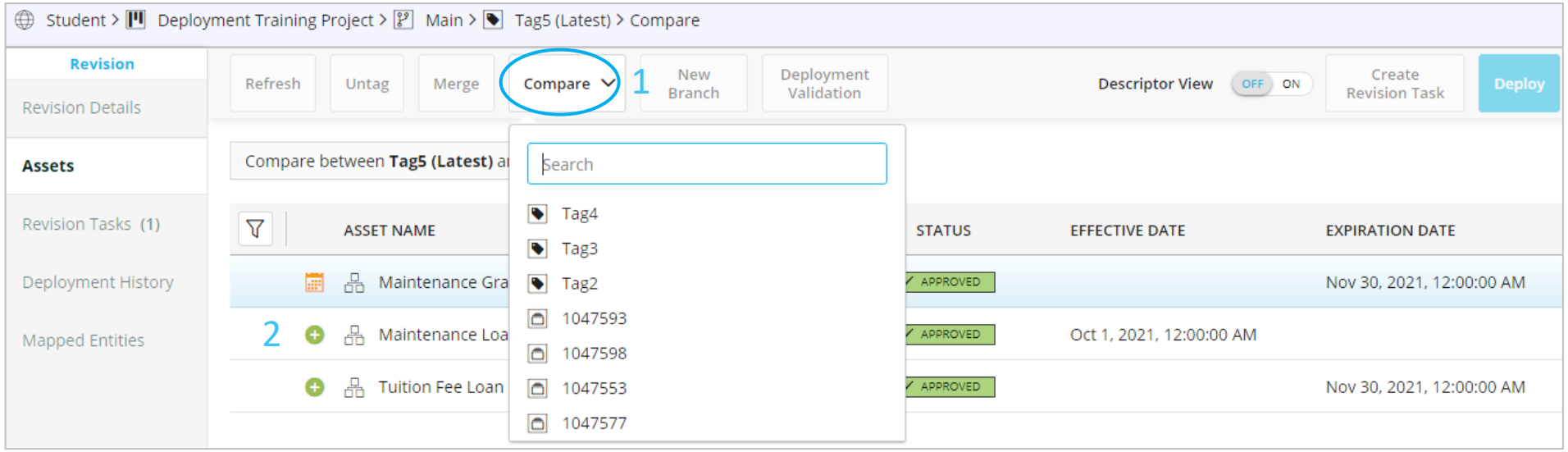


Demo

- Show Environment Deployment History from Reports tab
- Select the Environment to which we deployed

SAPIENS

Compare Revisions



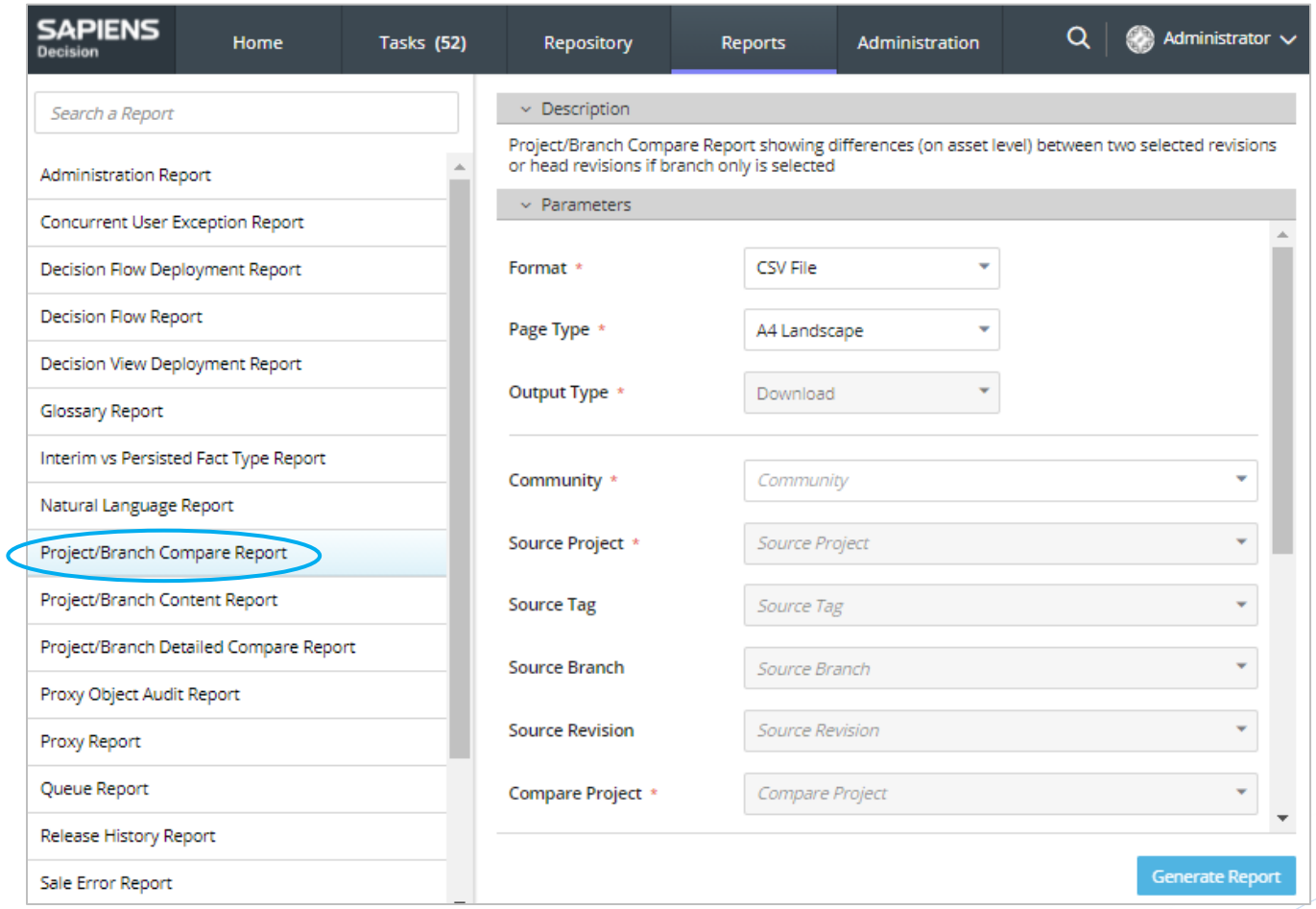
1. To compare a Revision with one of the previous Revisions of the same Branch, use the Compare menu and select the Revision to compare with
2. The changes between the Revisions are marked with icons

Icon Legend

- Asset Added
- Asset Removed
- Deployment Descriptor Added
- Deployment Descriptor Changed
- Deployment Descriptor Removed

Compare Revisions Report

- The ‘Project/ Branch Compare Report’ lists asset-level differences between two selected Revisions or two latest Revisions if an entire Branch is selected
- The Compare Report includes a list of added assets that exist in the ‘compared’ Revision but not in the ‘source’ Revision
- Assets that do not appear in the compare Revision, but do exist in the source Revision are marked as deleted assets
- Unchanged assets (included if “Show changes only” is not checked) are marked as “No Change”



Compare Revisions Report

SAPIENS
Decision

Project/Branch Compare Report

Parameters:

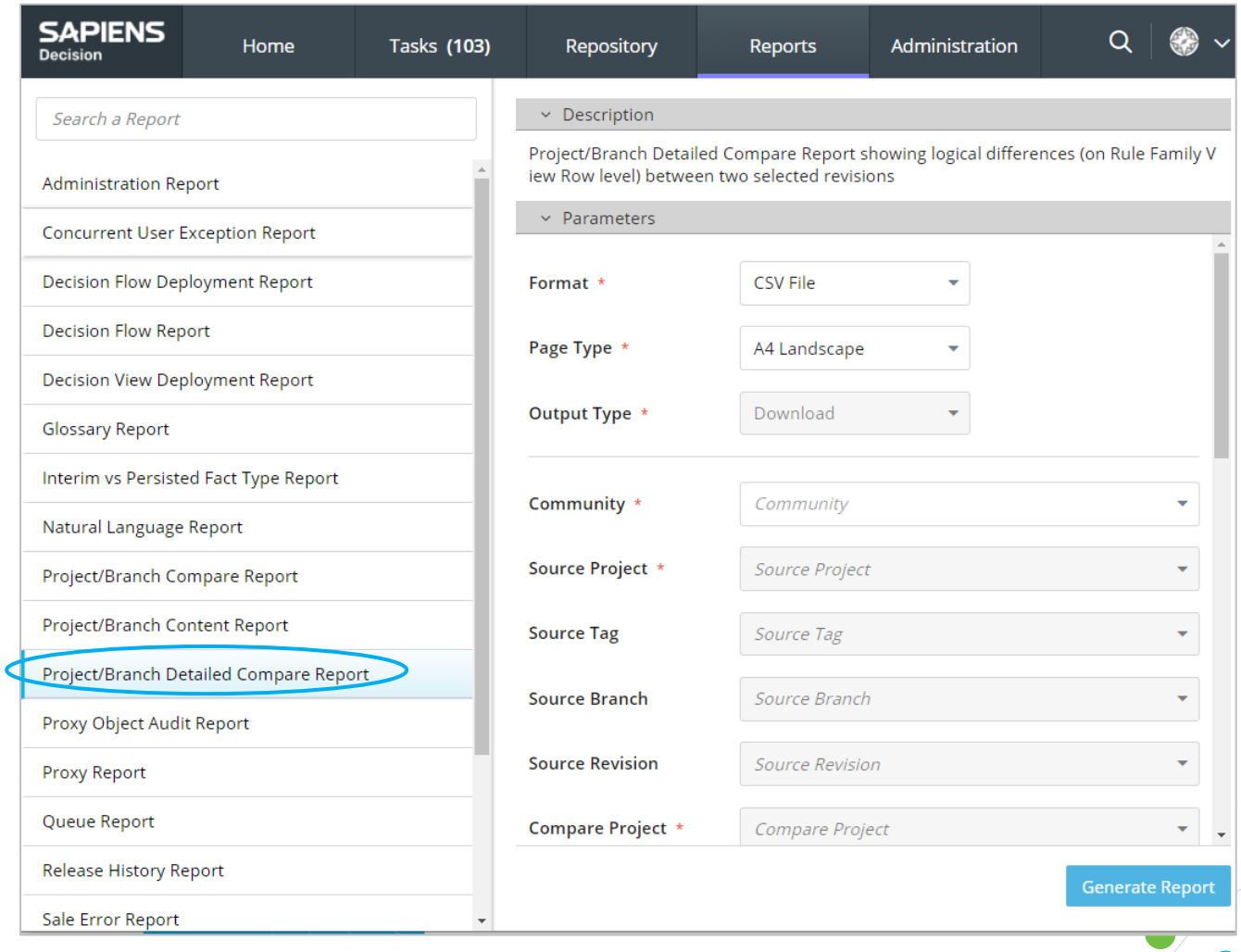
Community: WORLD
Source Project: Deployment Training Project
Source Tag:
Source Branch: Main
Source Revision:
Source Head Tag/Revision: Tag 4
Compare Project: Deployment Training Project 2
Compare Tag:
Compare Branch: Main
Compare Revision:
Compare Head Tag/Revision: Tag 2
Descriptor Custom Properties:
Show Changes Only: false
Include Deployment Descriptor Details: false

* Columns with no values will not appear in the report

Change Status	View Group	Asset Type	Asset Name	Asset Ver.
New	Life Underwriting Folder	Decision	Determine Training Deployment Decision 1 (View: Base)	1.2
Delete	Life Underwriting Folder	Decision	Determine Training Deployment Decision 3 (View: Base)	1.0
Delete		Decision Flow	Training Flow 1	1.0
Delete		Decision Flow	Training Flow 2	1.0
Change	Life Underwriting Folder	Decision	Determine Training Deployment Decision 1 (View: Base)	1.0
Change	Life Underwriting Folder	Decision	Determine Training Deployment Decision 2 (View: Base)	1.0
Change	Life Underwriting Folder	Decision	Determine Training Deployment Decision 1 (View: Base)	1.1

Compare Revisions – Detailed Report

- The ‘Project/ Branch Detailed Compare Report’ lists logical differences on the Rule Family Row level between two selected Revisions
- The report shows only the rows that are different across all versions of Flows, Decisions, and Rule Families in the compared Revisions



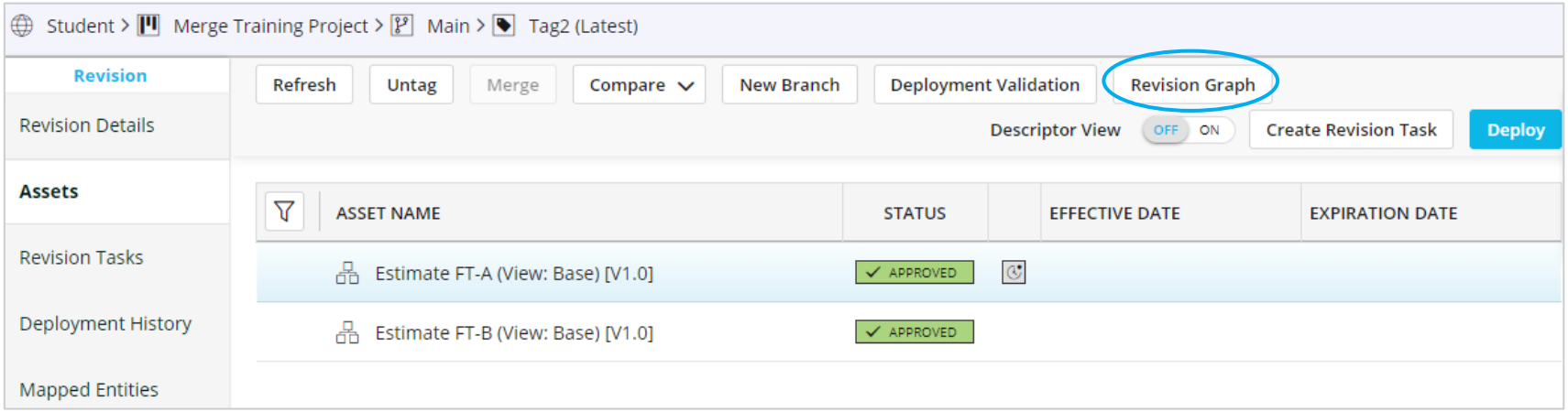
Compare Revisions – Detailed Report

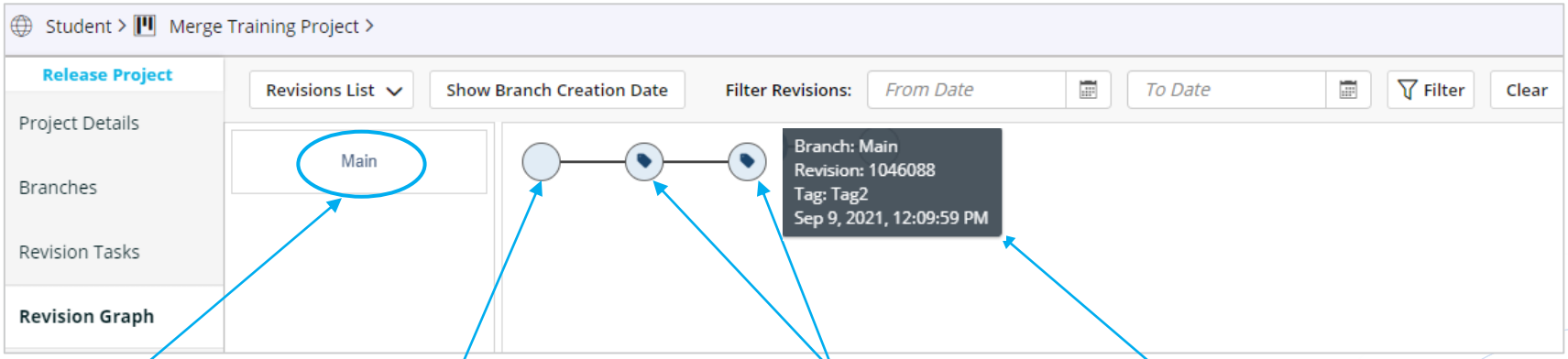
Project/Branch Detailed Compare Report																
Parameters																
DF Status	Change Status	Message (	View Grou	Decision Flow	Source DF	Compare	Decision View	Source DV	Compare	Rule Fami	RFV Ver.	Row Id	RFV Row Natural Language			
DF Deleted	DV Deleted	Deleted		Training Flow	1		Determine Training Decision 1 (View: Base)	1		Training D	1	1055551	(Training Decision 1 = 1)			
DF Deleted	DV Deleted	Deleted		Training Flow	1		Determine Training Decision 2 (View: Base)	1		Training D	1	1055575	(Training Decision 2 = 2)			
	DV Deleted	Deleted	Life Underwriting Folder				Determine Training Deployment Decision 3 (View: Base)	1		Training D	1	1056581	(Training Deployment Decision 3 = 3)			
	Row Changed	Changed	Life Underwriting Folder				Determine Training Deployment Decision 1 (View: Base)	1.1	1.2	Training D	1.2	1064606	(Training Deployment Decision 1 = 10)			

SAPIENS

Revision Graph

Click the Revision Graph button [Revision Graph] from the Revision tab or select from the left vertical tab on the Project Screen to see all Revisions and Branches in the Project





Branch name

Revision without Tag

Revisions with tag

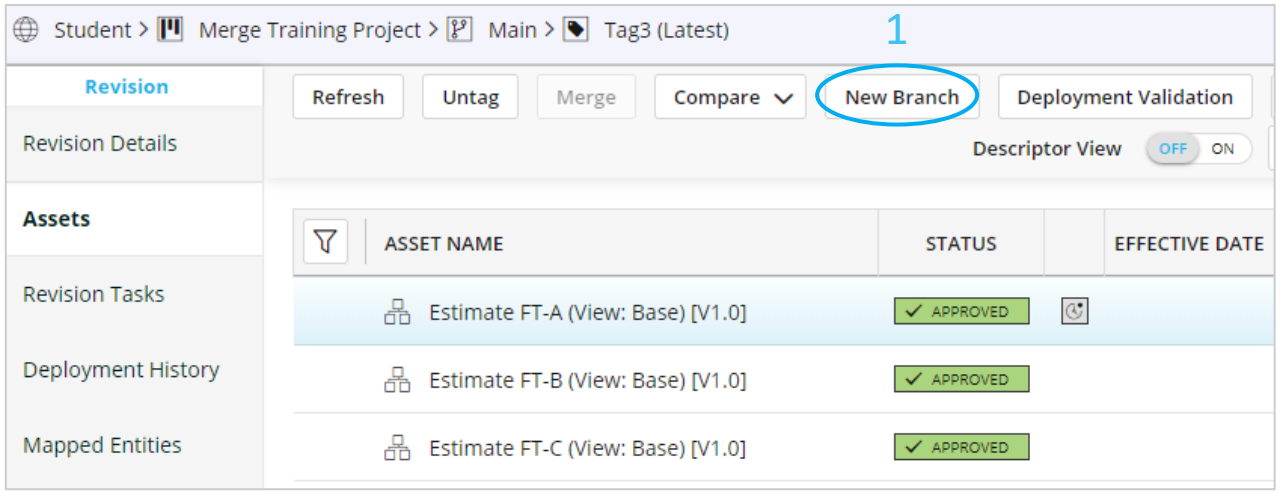
Revision details
(hover over)

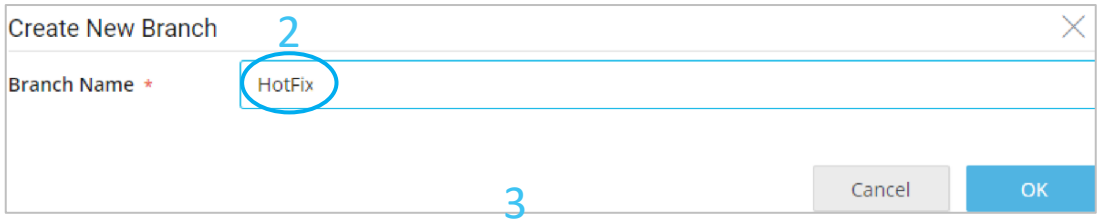
SAPIENS

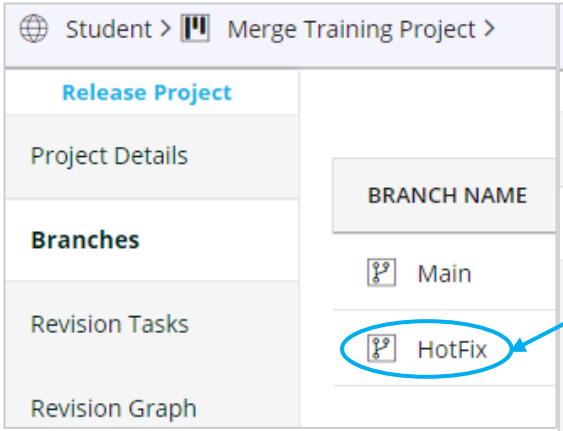
Create New Branch

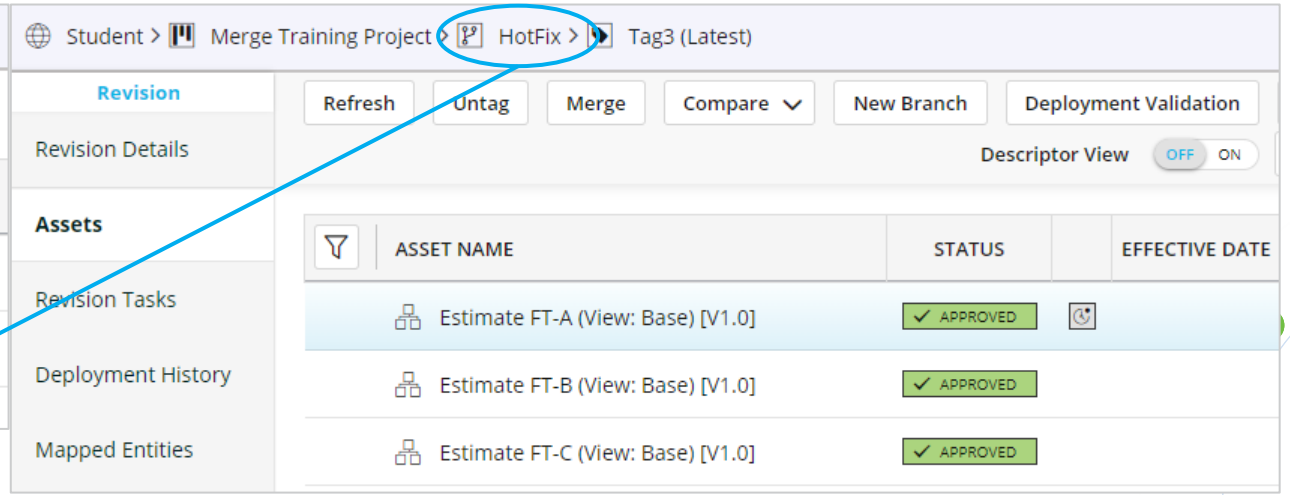
Open the Branch and Revision from which the new Branch will be created

1. Click New Branch button from the menu
2. Type the name of the new Branch
3. New Branch is created



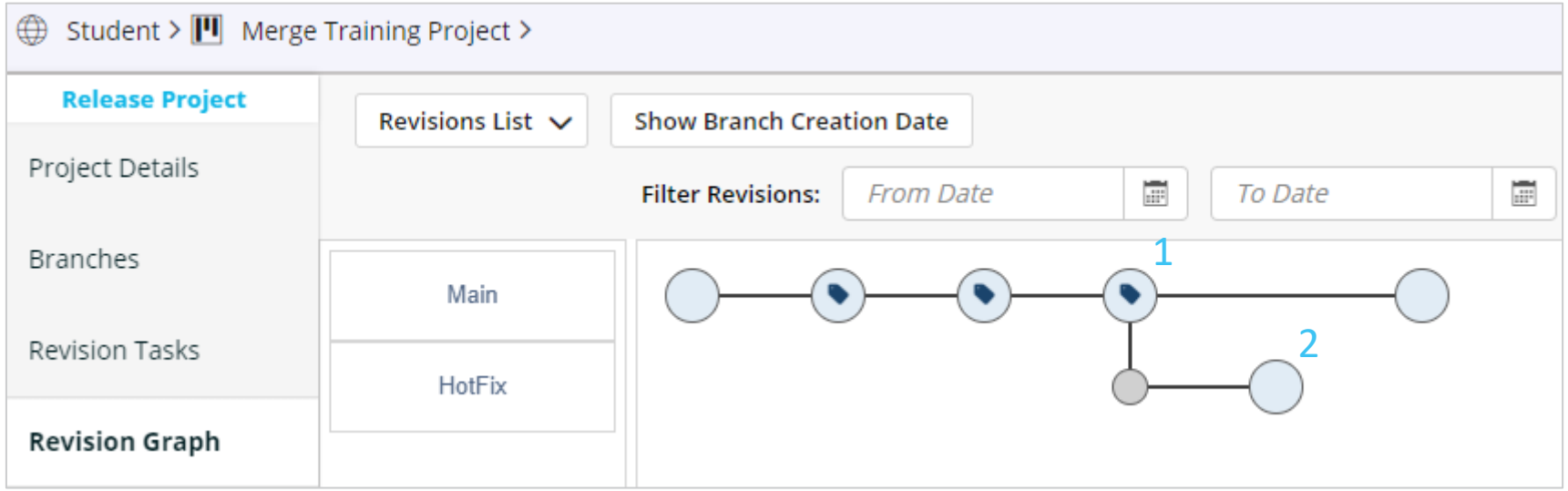






SAPIENS

Merge Branches



- **Merge** is between different branches within the same project
- Merge is supported from any Revision of one Branch to the latest Revision of another Branch
- The merge creates a Revision Task consisting of the accumulated actions on assets (additions, changes, and removals) made from the junction point of the branches (point 1) to the 'from' Revision (point 2)

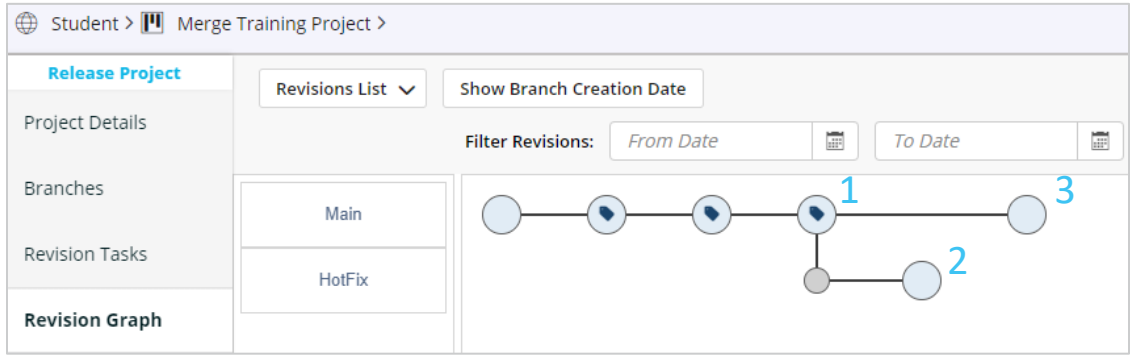
Merge Branches

Example for Accumulation of Actions:

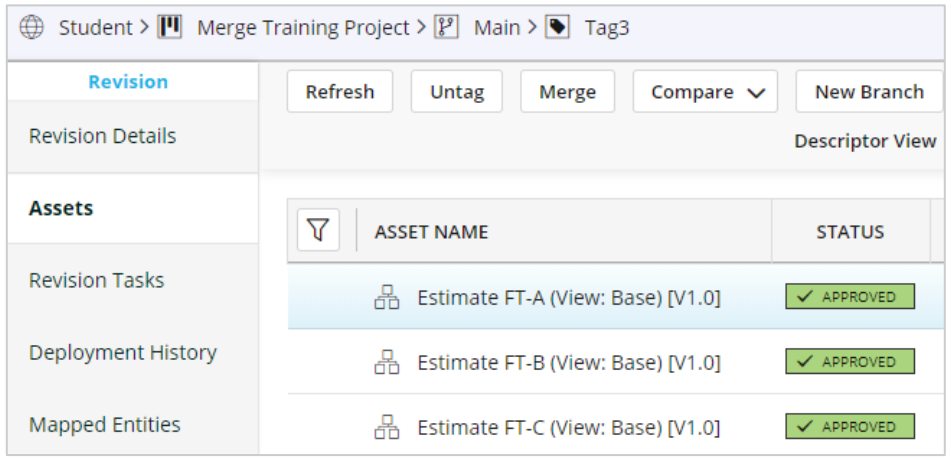
1. Junction Revision (Main Branch having Assets A,B,C)

2. ‘From’ Revision (HotFix Branch having Assets B,C,D)

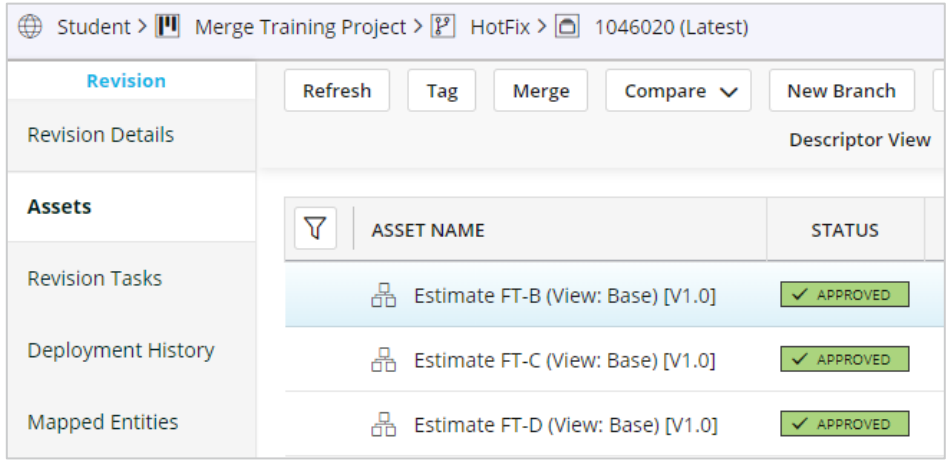
3. ‘To’ Revision – latest Revision of Main Branch (Assets A,B)



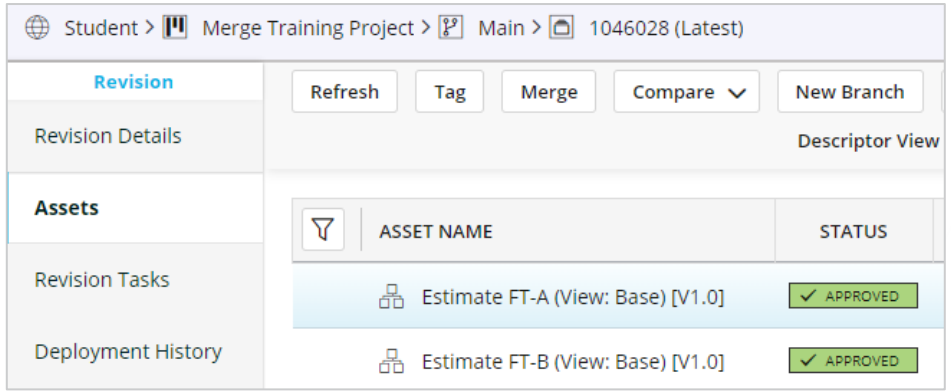
1



2



3



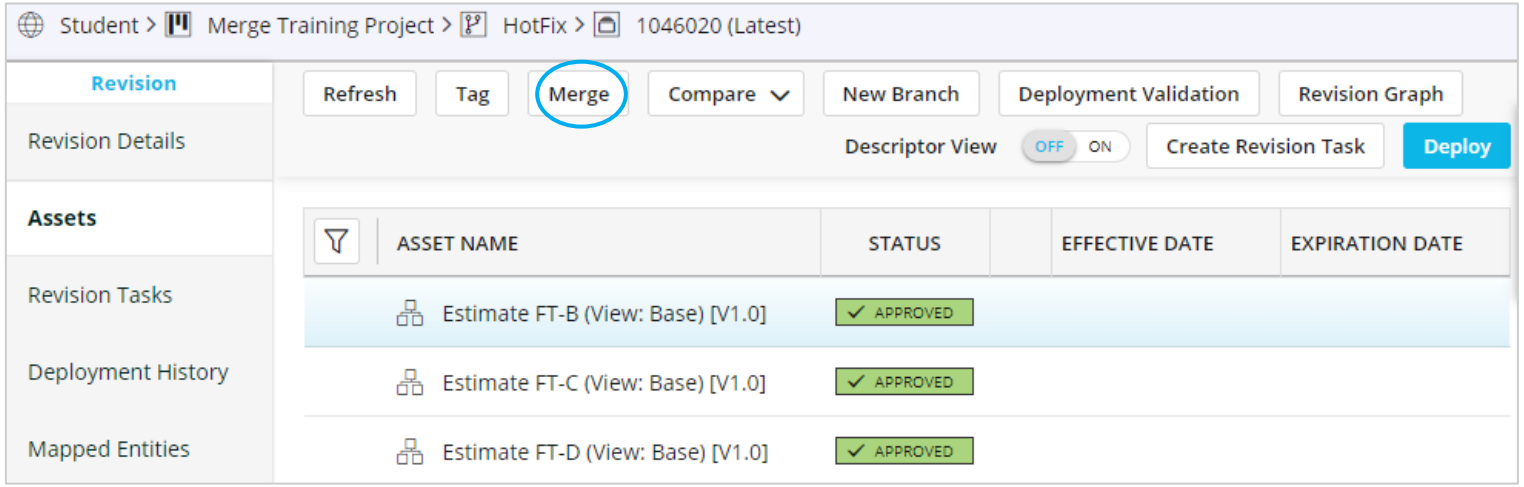
SAPIENS

Merge Branches

1. To merge, open the 'From' Branch Revision and click Merge

2. Select Destination 'To' Branch to merge and click Merge

1



2

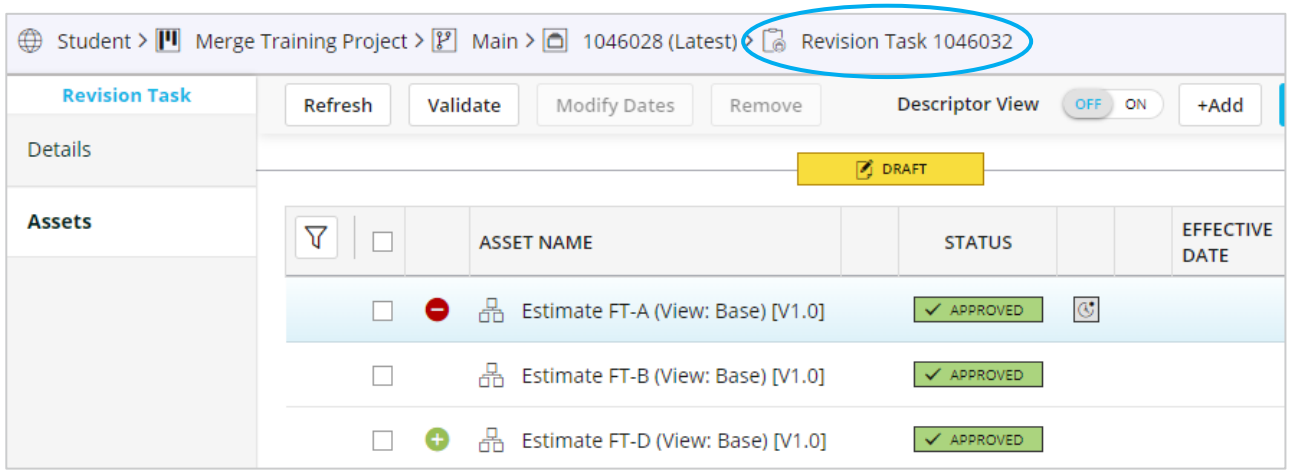
Merge Branches

1. Resulting Revision Task on the Main branch with the accumulated changes (-A +D)

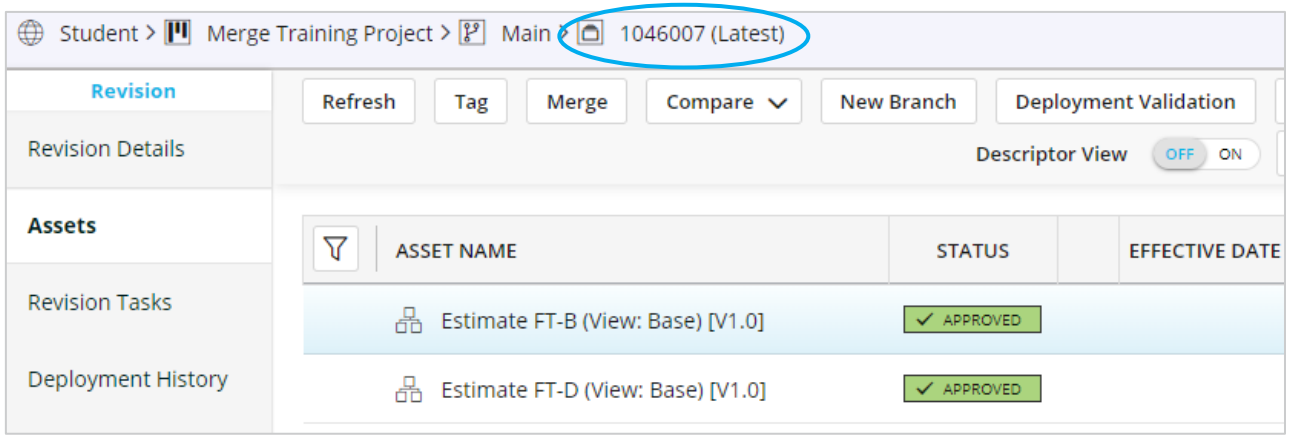
2. After submitting the Revision Task, the merged revision is created

3. Dotted arrow represents the merge

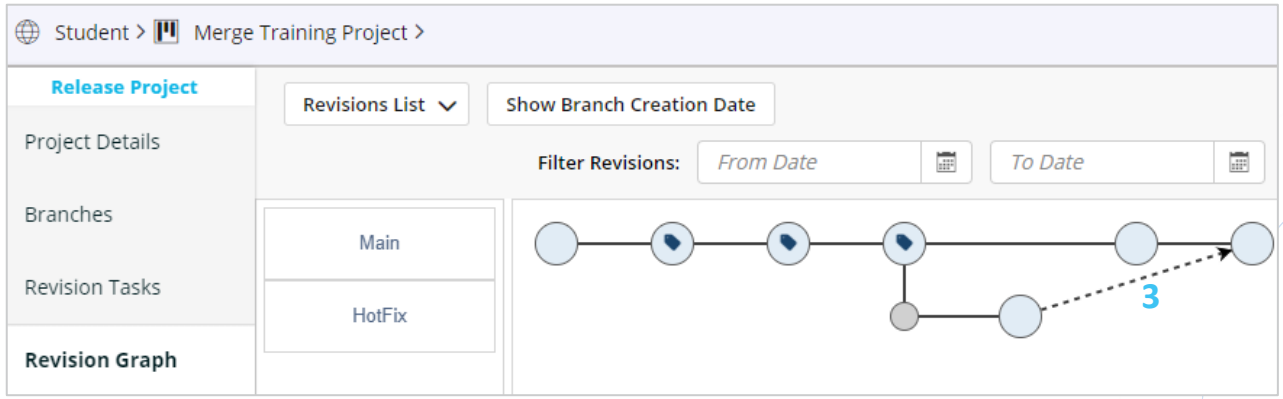
1



2

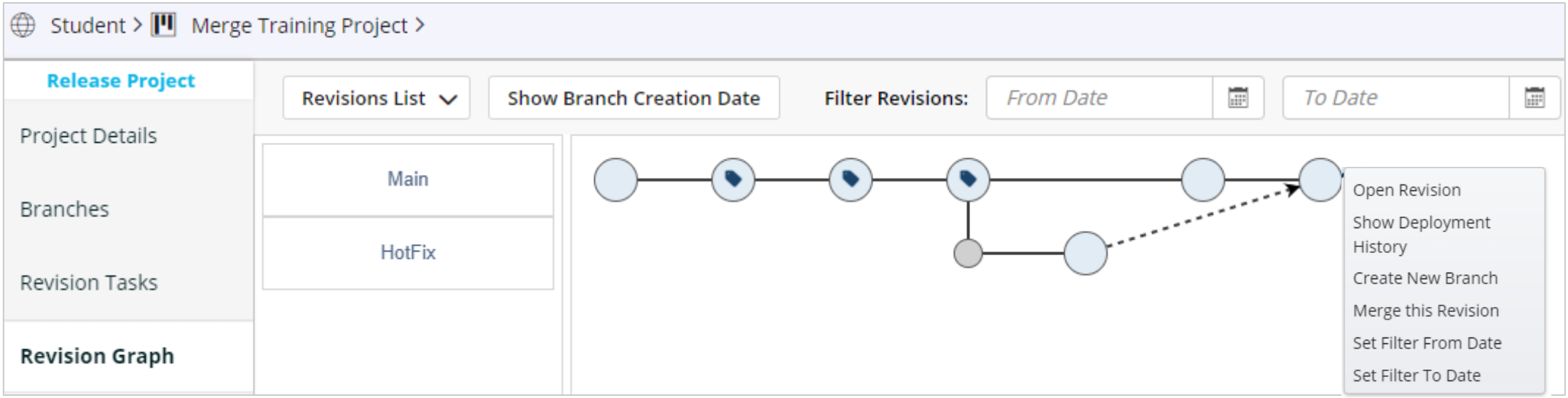


3



SAPIENS

Revision Graph Model Actions



When you right click on a Revision in the **Revision Graph**, you can make the following actions:

- Open Revision: Click Open Revision to view the Revision details of the asset
- Show Deployment History: Click Show Deployment History to view the deployment history of the asset
- Create New Branch: Click Create New Branch to add a new branch from this Revision
- Merge this Revision: Click Merge this Revision to select the target branch to merge into; a Revision Task will be created with the accumulated changes performed in the branch from the junction point to this Revision
- Set Filter From Date: Select Set Filter From Date to set a filter from the date of the revision
- Set Filter To Date: Select Set Filter To Date to set a filter to the date of the revision

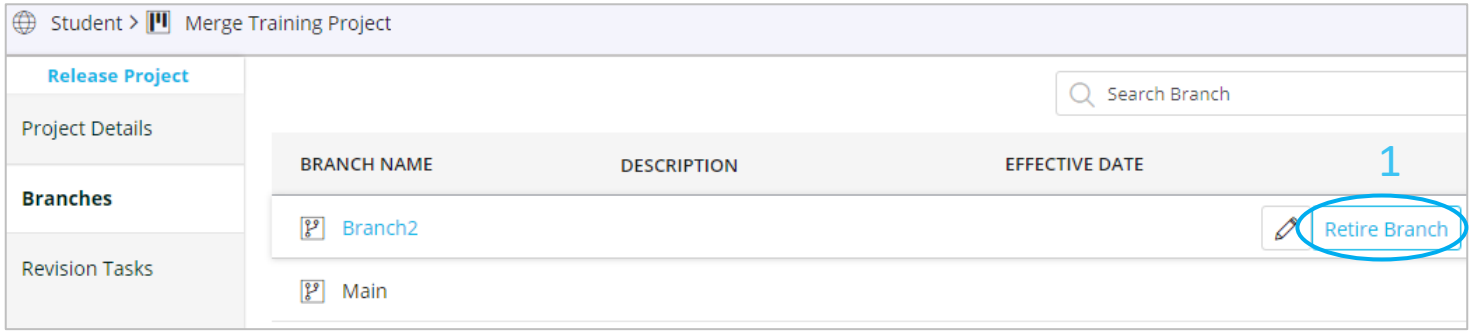
Demo

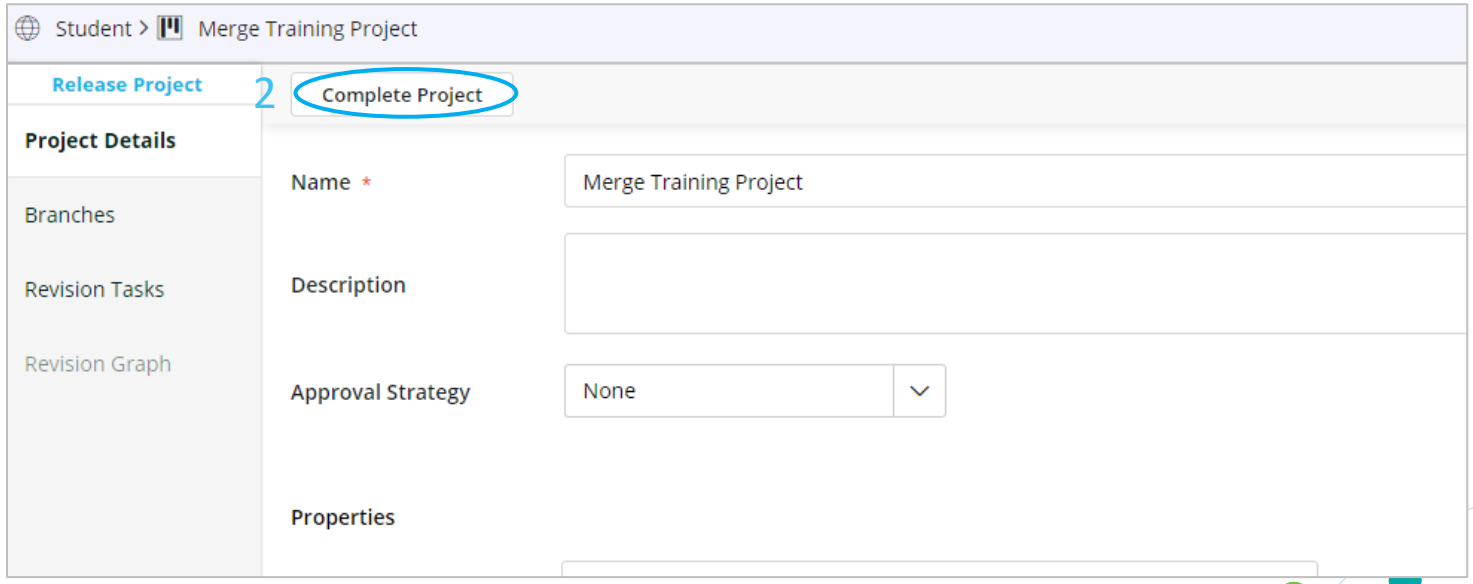
- Create HotFix Branch from Tag1
- Create Revision Task 4 on the HotFix Branch and approve it
- Open Revision Graph and merge to Main Branch from Hot Fix
- Show Revision Graph after Merge

SAPIENS

Retire Branch and Complete Project

1. On Branch screen click Retire Branch button
2. On Project Details screen click Complete Project button

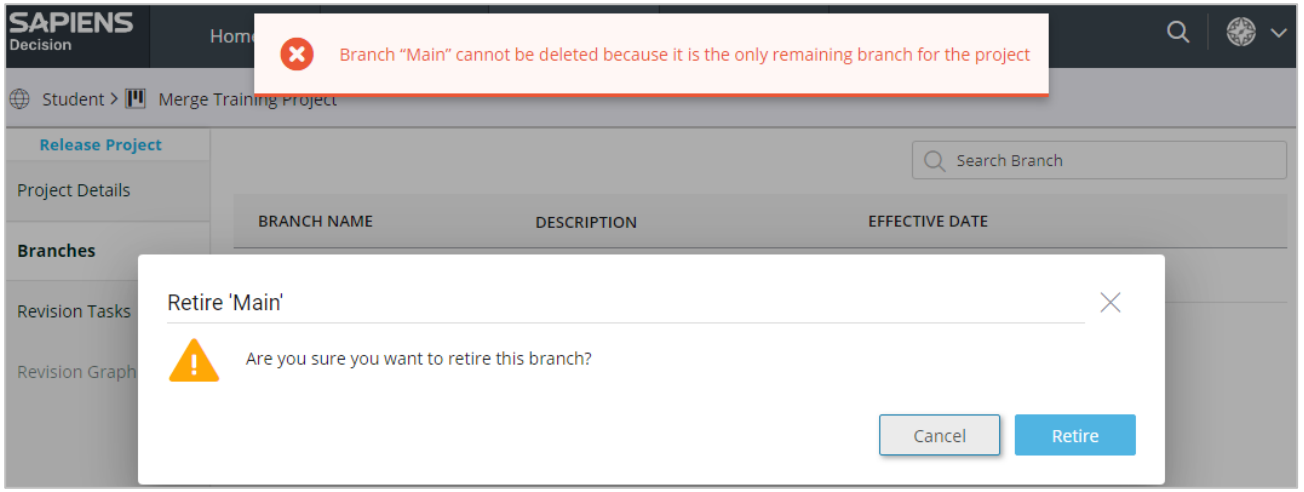




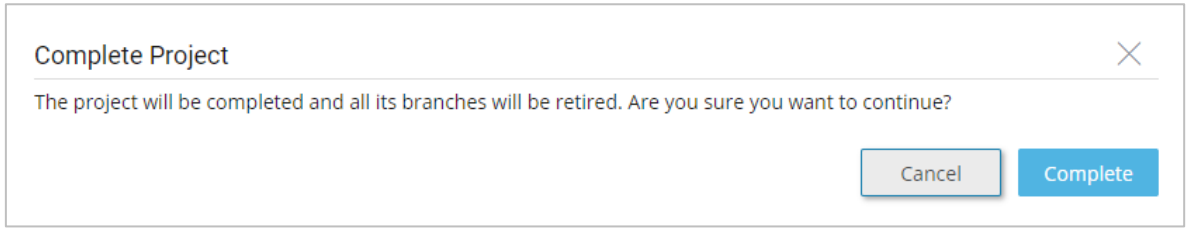
Retire Branch and Complete Project

1. If there is only one Branch - it cannot be retired
2. When a Project is completed, all its associated Branches are retired
3. All submitted Revision Tasks or deployments are automatically rejected
   - All pending Revision Tasks are no longer available since the Branch has been retired

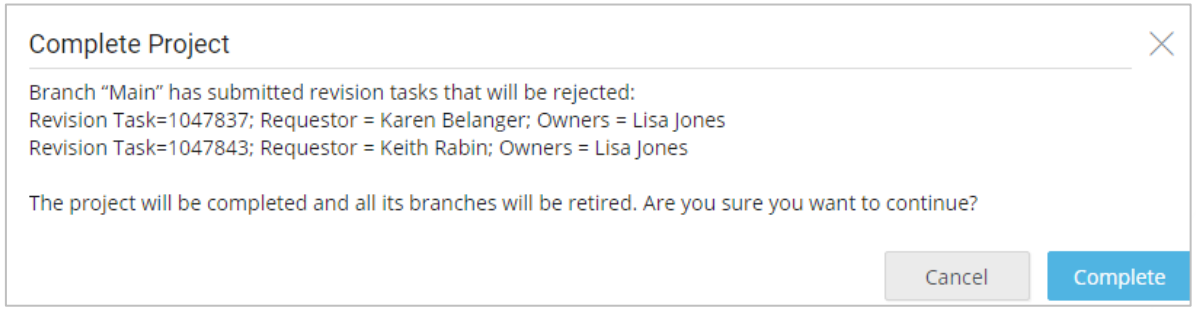
1



2



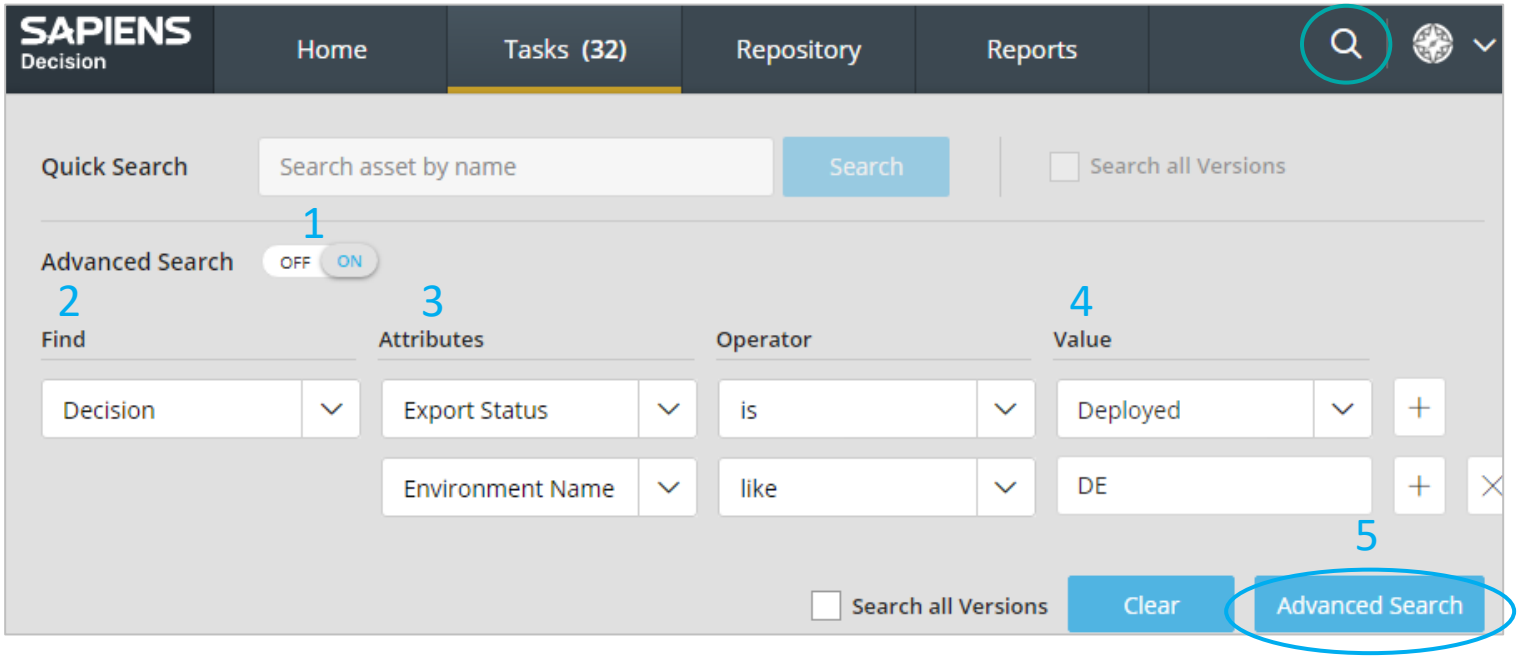
3

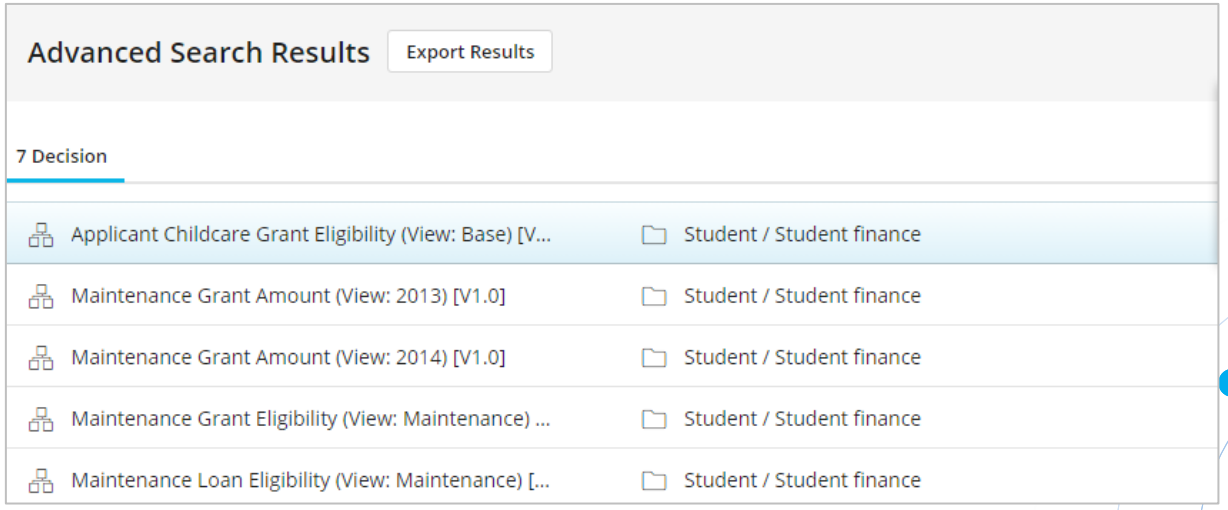


Advanced Search

Search for assets that were deployed using the search tool

1. Turn Advanced Search on OFF ON
2. Select which asset type to find
3. According to which attributes
4. Based on the attribute, some values can be selected, and others must be manually input
5. Click Advanced Search to run the query





Agenda

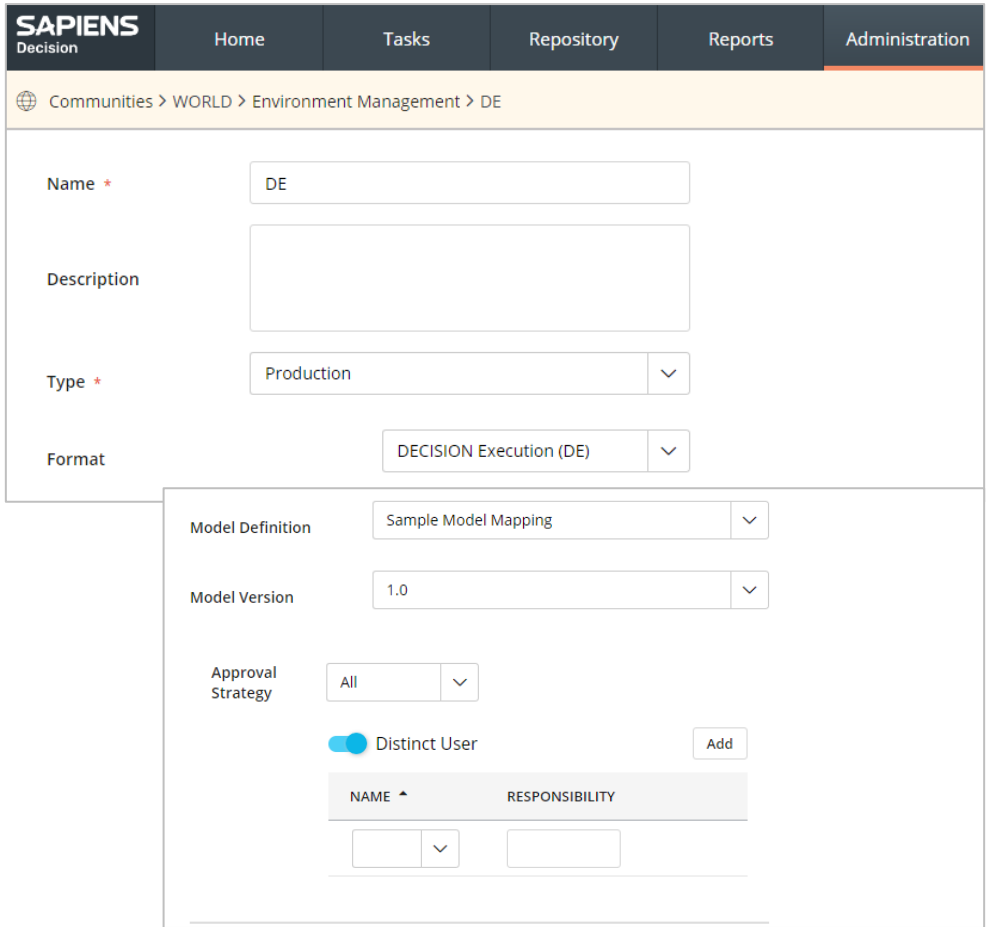
- Introduction
- Release Management
- **Approval Strategy**

SAPIENS

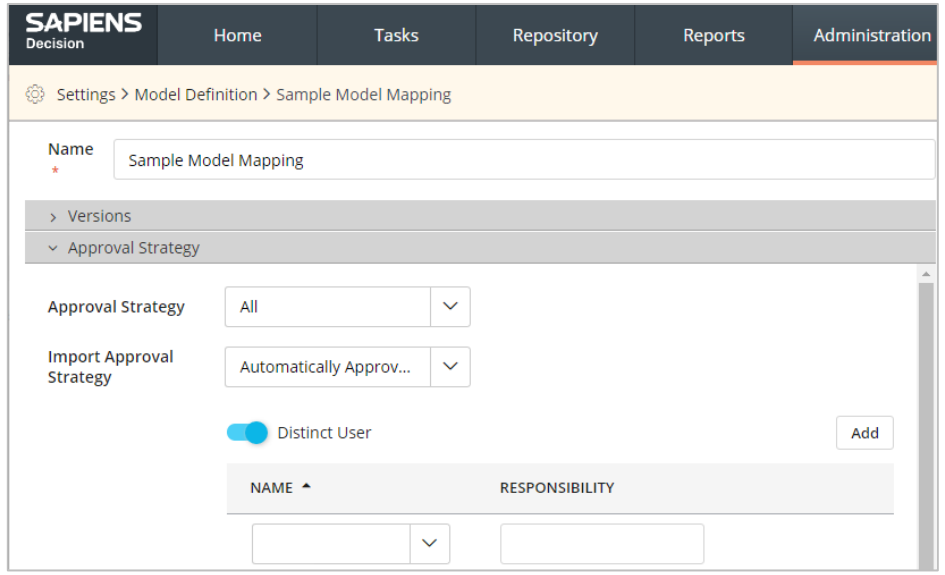
Approval Strategy

- Approval Strategy is a generic mechanism that governs business process requests
- Approval Strategy applies to the following modules:

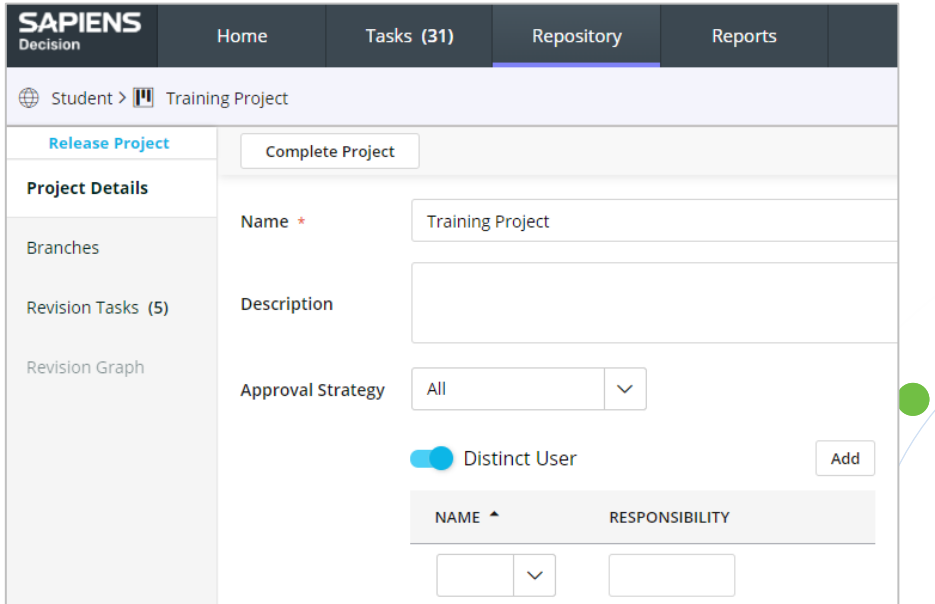
Environments: govern requests for deployment to Environments



Modeling Approval Process: govern changes in Fact Type Model Mapping



Projects: govern changes in Branch content applied via Revision Tasks



Approval Strategy

- In order to appear in the owners list, the user must have the relevant permission:

  For Model Mapping - *MODEL_MAPPING_APPROVAL*
  For Environments - *ENVIRONMENT_APPROVAL*
  For Projects - *BRANCH_REVISION_APPROVAL*

- For Environments and Projects, in addition to the approval permission, the user must have permission for the Community

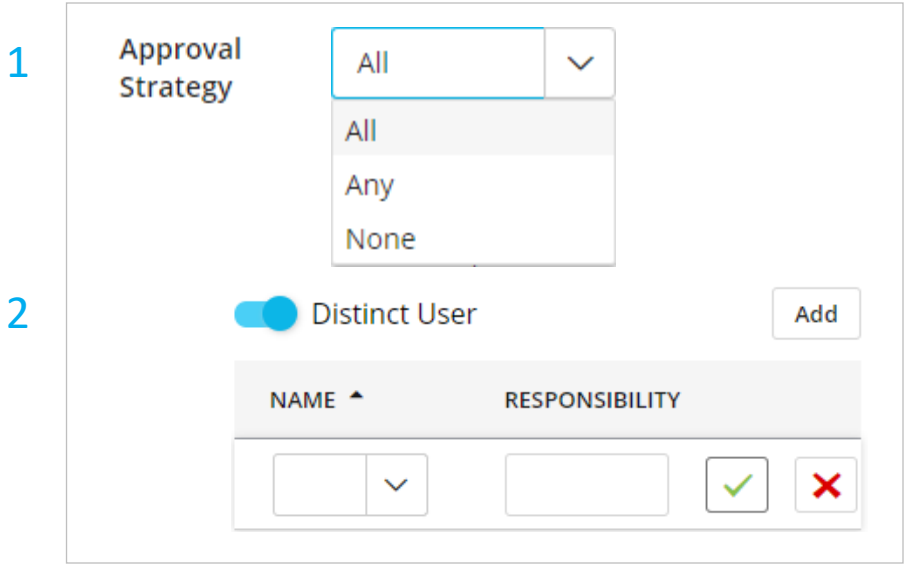
Approval Strategy Template

The Approval Strategy template definition has 2 sections:

1. **Set Approval Strategy Type**
   - None: request instance is immediately approved and performed upon submission (no owners)
   - All: all assigned owners must approve the request before it is performed
   - Any: one of the assigned owners must approve the request before it is performed

   Distinct User (applies to both All and Any): the user that submits the request cannot be the only approver of their own request

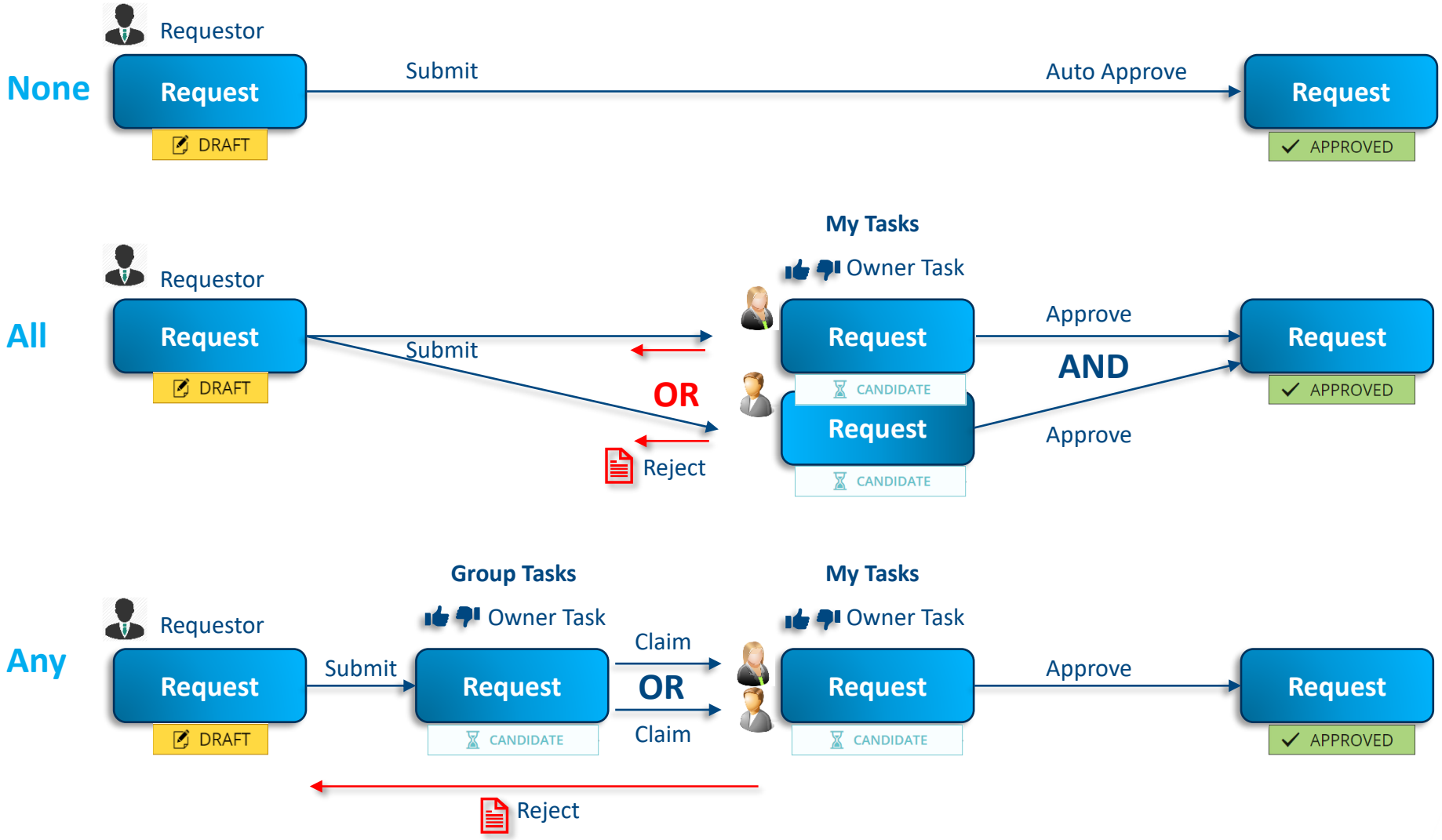
2. **Set Owners and Responsibilities**
   - Assign users with permission to perform request
   - Define responsibility (optional)



If one of the assigned owners rejects the request, then request is rejected altogether

SAPIENS

Approval Strategy Workflow



Questions?

SAPIENS
Partnering for Success

Thank you!

© Copyright Sapiens International 2022