Contents

[Training preparation 2](#_Toc448239933)

[1.1 Common 2](#_Toc448239934)

[1.2 Trainer activities for Developer training 2](#_Toc448239935)

[1.3 Trainer activities for Responsible training 4](#_Toc448239936)

[1.4 Trainer activity for High Level 4](#_Toc448239937)

[1.5 Trainer activity for NonSW 4](#_Toc448239938)

# Training preparation

## Common

Create a Top Level training project for the training session for Source and ICTM: (e.g.: *IMS\_Training\_cw40\_ING*)

For each trainee, create a subproject at the first level of Top Level project with the trainee name.

At the first level of Top Level project, create a subproject named Common, for trainees to see what happens when they work in the same current configuration.

NOTE: do we have to have files inside?

(Slide 4)Trainer explains definition of:

Version -> Revision

Configuration -> consistent set of file versions that go together

Work area -> Sandbox

## Trainer activities for Developer training

Create a group in Project Configurator, assigned to the training project as Enhanced Developer.

Trainees will be assigned to this group.

* Update the meta-data in the *Project Item* of the training project.
* Create a simple *Structure Specification*. (e.g.: IMS\_Training\_cw40\_ING Structure Specification)
* Create the *Structure Elements*
* Create the *Release* for delivery *Sample 1*, set it to state Planned.
* Create a release for delivery *Sample 2* (do not set it to Planned yet).
* Create a Release for bugfix - *Sample 1 Bugfix* (do not set it to Planned yet)
* For each trainee, create a Realization Order linked to the Release *Sample 1*, assign it to each trainee and promote it to State Planned. Set the Planned Release as the *Sample 1* Release.
  + For each RO, the trainees must create a Change Package (the exercises from the training slides are done – *input slide number*)
  + CM part – create sandbox – the sandbox has to be created from the Top Level for each trainee (not from their subproject) so they can see the differences and see how resynchronize works.
* The Trainer will explain the most common icons: delta, working and the filter of the sandbox.

After the actions done by the developer(close CP, set RO to state Closed), the Responsible (the Trainer) has to Verify and freeze the configuration and then deliver Sample 1 to customer (set Release Sample1 to Released ?)

* Verify the consistency of the configuration(the BM has to check if the ROs are implemented the expected way, if the configuration compiles)
* Create a checkpoint of the Top Level Project corresponding to the first delivery.(Give the checkpoint to CFT, Deliver to customer)
* Set Sample 2 Release to State Planned and create ROs for each trainee
  + For each RO, the trainees must create a Change Package. (the exercises from the training slides are done)

Unexpected requests from the customer: changes/feature requests, fixes for defects.

The trainer must set the Sample 1 Bugfix to state Planned

* Each developer will create and analyze an Issue
* The CCB (Trainer) will decide to implement some of them on Sample 1 Bugfix Release.
  + With For each accepted Issue, the trainees will create an RO.
  + The Planned Release of the RO is set to “*Sample 1 Bugfix*”
* The Responsible (Trainer) creates a development path from Sample1 Checkpoint and announce its name to developers
  + Developers now create a Variant Sandbox in parallel to their normal sandbox (if they did not create the CP from the Integrity part, they can create it now from the Source part)
  + Developers branch files, drop a member from the devpath, update member revision on devpath to see that there is no effect on the mainline when making modifications on the devpath
* Trainer has to check if there are any locks left on devpath.
* Responsible (the Trainer) will create a checkpoint on devpath (same as Sample 1)
  + Developer will close the ROs for Sample 1 Bugfix.
  + Developer will set the Issue to state Realized, if the Issue is assigned to him.

CFT start testing all Issues in state Realized and set them to Approved if the tests have been passed.

Bugfix Integration for Sample 2

* Developers will do merge (drag and drop not Apply CP)

## Trainer activities for Responsible training

Create a group in Project Configurator, assigned to the training project as Responsible.

Trainees will be assigned to this group

## Trainer activity for High Level

## Trainer activity for NonSW

Create a Top Level training project for the training session for ICTM: (e.g.: *IMS\_Training\_CW11\_TSR*).

Trainees will be assigned to the project with Responsible role. If you have Test Managers as trainees you can assign them Responsible + the Test Manager role so they will see how the Issues notifications work.

* Update the meta-data in the *Project Item* of the training project.
* Create a *Structure Specification*. (e.g.: *IMS\_Training\_CW11\_TSR* *Structure Specification*)
* Create *Structure Elements* for disciplines and functionalities:

1.Top Level Project

2. SW

3. SW\_Funct\_1

4. SW\_Funct\_2

5. ME

6. EE

7. System

8. CFT\_Test

* Put the Structure Elements to Approved (optionally you can assign them to the trainees)
* Create a Product *Release (Top level project/01.00),* select the Top Level Project SE and set it to state Planned. In the training when arriving at slide 24 (Create a Release Item) create a discipline Release on *EE (EE/01.00)* or *ME(ME/01.00)* and link it with the Used by field to the Product Release.
* For slides 25 to 34 during training create an Issue for a discipline with one Analysis Task for System and 2 RO(s) one for discipline and one for System.
* For slide 51 select the Issue created before and use it to create a new Issue with Copy item option.