All trainees shall have Responsible role on (creating a group for the training)

* /BS/SWPDevelopment project in Integrity: to be able to classify and decide the Issue.
* /BS/SWP project in Source

*Create a Top level Project <trainingName>(e.g. IMS\_Training\_cwxx\_loc) for Source and ICTM:*

* *Create the project Item*
* *Give access to the trainees to this project*
* *Create a simple Structure Specification with only 3 Structure Elements: Top Level Project, Appli, SWP\_Integration*
* Create a Release (<projectName>/Set up, set it to planned), an RO (set it to Planned) and a CP to be able to continue
* *Create n sub projects (1 per trainee <traineeName>) to simulate n customer projects (to be able to play the SWPLC)* using the created CP

Create in Source, under /BS/SWP/RtU a dummy project for the training e\_Rtu<trainingName>\_551x containing:

* 3 configuration subproject
* N delivery boxes (if you have N trainees planned for the training). Delivery boxes shall be shared from /BS/SWP/DB) **To check** if we need to have files inside
* Source folder + IDEAS.BAT and pdr file

using the created CP

Do a checkpoint of the RTU, in the current project configuration check and “configure” the 3 configuration subprojects as normal. Check that DBs are configure as Build on their last revision

Create 3 releases linked to this RTU on the structure element Z\_for\_Release\_only\_AR\_21\_551x. (AR\_21\_551x/0100\_trainingName, AR\_21\_551x/0200\_trainingName, AR\_21\_551x/0300\_trainingName on /BS/SWPDevelopment

* Link the first one to the created RtU checkpoint using the Released Checkpoint field from the Tab Relationships, (during the exercise ask the trainees to set the field Found in Release of the Issue with this value)
* The second release, is the release of the current RTU V cycle, (during the exercise ask the trainees to set the field Planned Release of the Realization Order created from the Issue with this value)
* The third release will be used in the exercise to start a new development

Create 2 Realization Orders for each trainee changing the Assigned user after creation setting them to Planned

* One Realization Order linked to the second Release (to be able to create his Change Package “preparation of environment” for the Rtu V cycle played during the training)
* One Realization Order linked to the third Release (to be able to create his Change Package “preparation of environment” for the set up of the second development cycle)

ChMa: decide if we do it or not: Scripts shall be copied on each trainee PC (send a mail to them few days before the training) For exercises slide 17: create a new view copying from the mks-bs-test-Source explaining how to create the buttons for the scripts.

Checkpoint script shall be installed; Release Note script shall be installed

Follow the slides and the associated exercises

For exercises slides 68, 69

* Create the Issue on **structure Element**=AR\_2.1
* **Found In Release** is set on the first Release you have created (AR\_21\_551x/0100\_trainingName)
* **Detected by**: Development or Internal Customer

**After Issue creation set**

* **Links to Source** must be set with the delivery box the trainee will have to work on
* **Functional Variant**=551x, 5560x
* **Project Variant**=the functionality corresponding to the delivery box
* Go through the workflow of the Issue till Accepted, create an RO with Planned Release=Second release (AR\_21\_551x/0200\_trainingName)

Slide 70 and 71 shall be explained (not played)

For exercises slide 94:

* Give naming convention for dev path uidxxxxx or FiLa or uidxxxxx\_BM or FiLa\_FR
* Give naming convention for Sandbox name <project\_name>\_[<DevPath\_name>]

For exercises slide 98:

* Do them in the created sandbox and DB assigned to the trainee
* Modify also some file in configuration subproject
* Don’t forget to ask them to use 3 CPs

Exercises slides 112 to 114

For exercises slide 121:

* Each trainee has to checkpoint the DB using Checkpoint and Apply Label
* CPs shall be close, the RO linked to the Issue to Realized
* Each trainee has to use Release Note with option subproject to establish traceability between new DB revision and corrected Issue(s)
* Show the SWBM activities using RO and (propagation) CPs for the BM, create the RtU new checkpoint, linked it to the second release
* Point 6 shall be explained

For exercises slides 122: Each trainee will have to create a new dev environment to develop on the third release

For exercises slides 124 to 128

* Each trained will play the SWPLC to create the customer delivery project in /BS/SWP/Integration/b\_TraineeName\_swp and integrate it in his customer project /BS/TrainingName/TraineeName
* Each trainee will play the SWTPL of the customer project in the project /BS/TrainingName/TraineeName

**Points to be highlighted during the training:**

SWP CM plan and SWP create Issue button

DBs shall be created empty under the subproject /BS/SWP/DB (or /BS/SWP/FBLDB) to be found easily on Add Shared Build then shared in RtU using the same name.

The SWP delivery to a customer project shall be created under /BS/SWP/Integration then shared in the customer project (the name can be changed).

Only SWP team can create Issue on /BS/SWPDevelopment. The Issue is created on a SWP Platform or SWP Platform version, the impact shall be analyzed on all SWP Platform/controller(RtUs) (functional variant shall be set accordingly), the implementation shall be decided (common to all RtUs: 1 RO for dev + x RO for Integration in other RtU, different for RtUs: x ROs for implementation + y ROs for integration) then finally on all Customer projects using this Platform/controller.

Several CPs shall be used to avoid problem on Apply CP or Resynchronize CP:

* one (per Release) for environment preparation (Add Shared Build, configure subproject as build or normal, create subproject, drop subproject…)
* one (per Issue) for modification of configuration files
* one (per Issue) for modification of DB files

each SWPDev develop on his dev path created for each new development from the last checkpoint of the RtU (variant sandbox is created once and retarget for each new development) :

* subprojects where everybody shall do changes are set on dev path (e.g. configuration subprojects, Source…)
* DBs where the dev has to perform the changes shall be set as normal
* Other DBs shall be kept as Shared Build
* Release/Issue(s)/RO(s) are created on /BS/SWPDevelopment project
* New version of DBs are delivered + CP(s) used to modify configuration files +pdr

each SWPLC develop on his dev path created for each new integration from the last checkpoint of the customer project

* subprojects where everybody shall do changes are set on dev path (e.g. configuration subprojects, Source…)
* application subprojects shall be kept as Shared Build
* SWP subproject shall be set as normal to reuse new version of DBs
* Release/Issue(s)/RO(s) are created on /BS/<customerProjectName> on SWP\_Integration SE
* New SWP version is delivered to the project + CP(s) used to modify the configuration files + pdr

On new DB version creation (to replace the link task/subproject existing in Synergy), the user has to run Release Note report to establish the traceability between the created revision and the Issue(s) solved by this revision