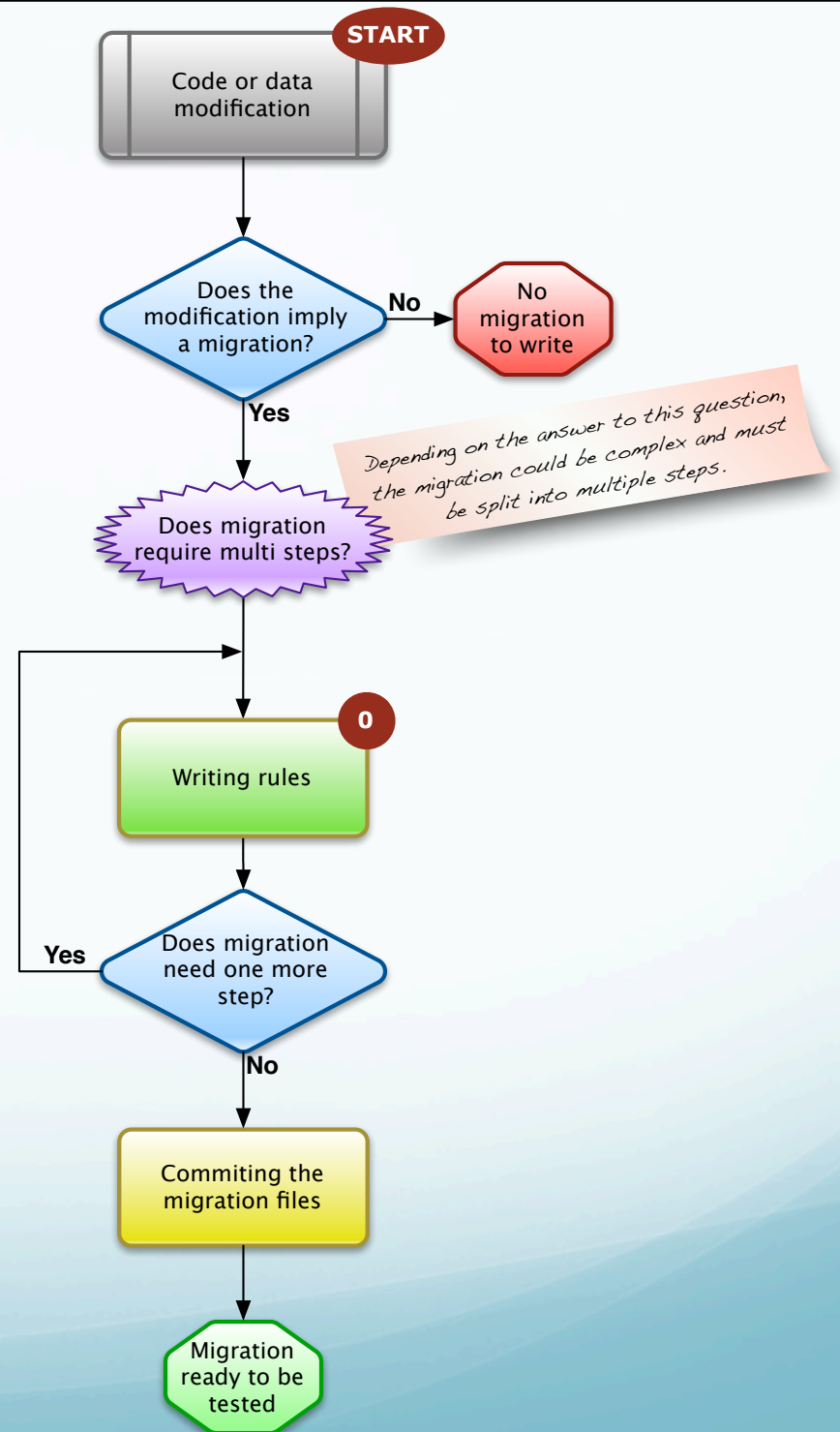


LMSWR

Lotaris Migration Scripts Writing Rules

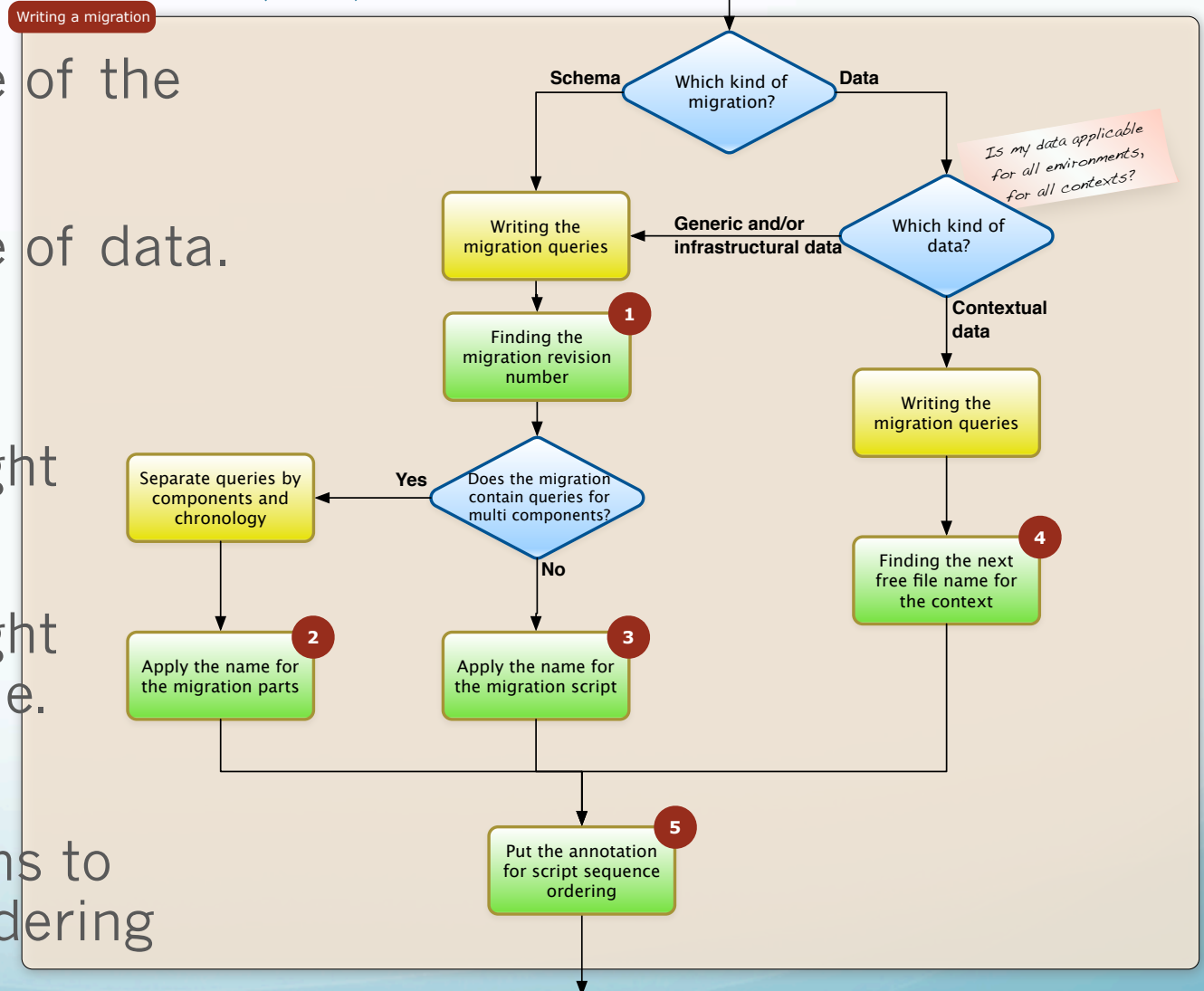
Whole process

- Does the code modification require a database modification?
- Determine whether the script must be split or not.
- Write the queries and place them in the right file.
- Name the file using the revision that introduced the modification.



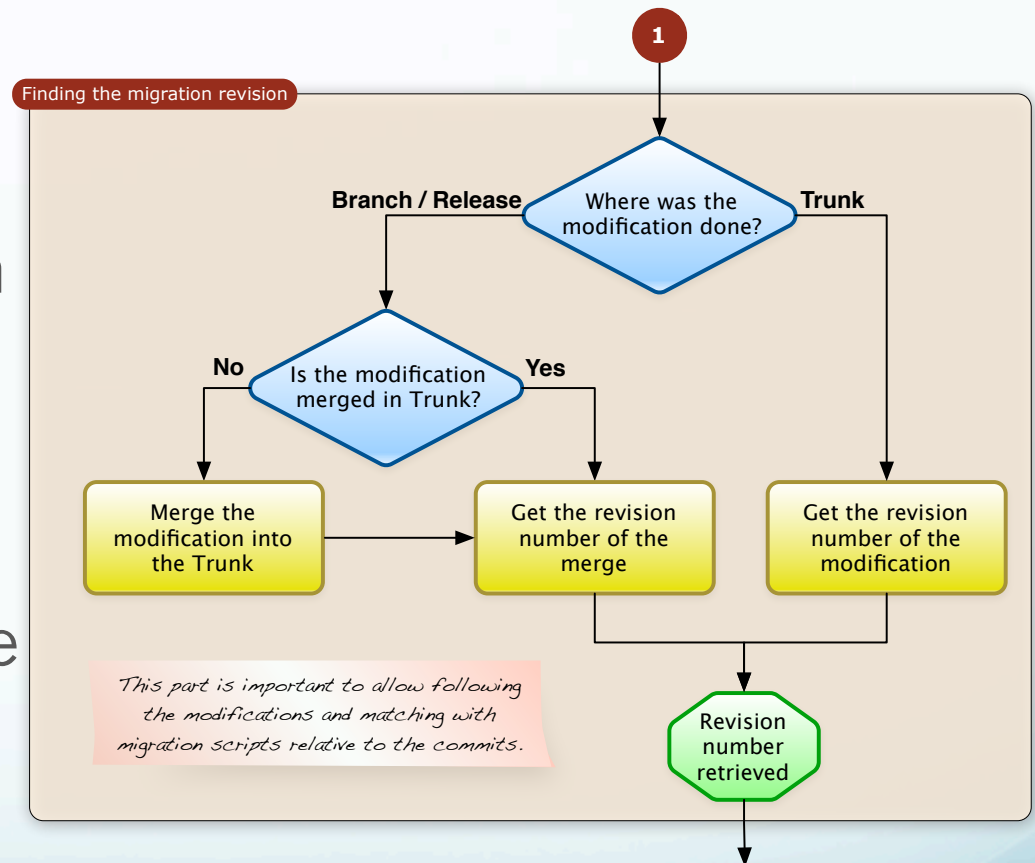
Global rules (0)

- Define the type of the migration.
- Define the type of data.
- Write queries.
- Retrieve the right file name.
- Retrieve the right place for the file.
- Enrich the file with annotations to improve the ordering algorithm.



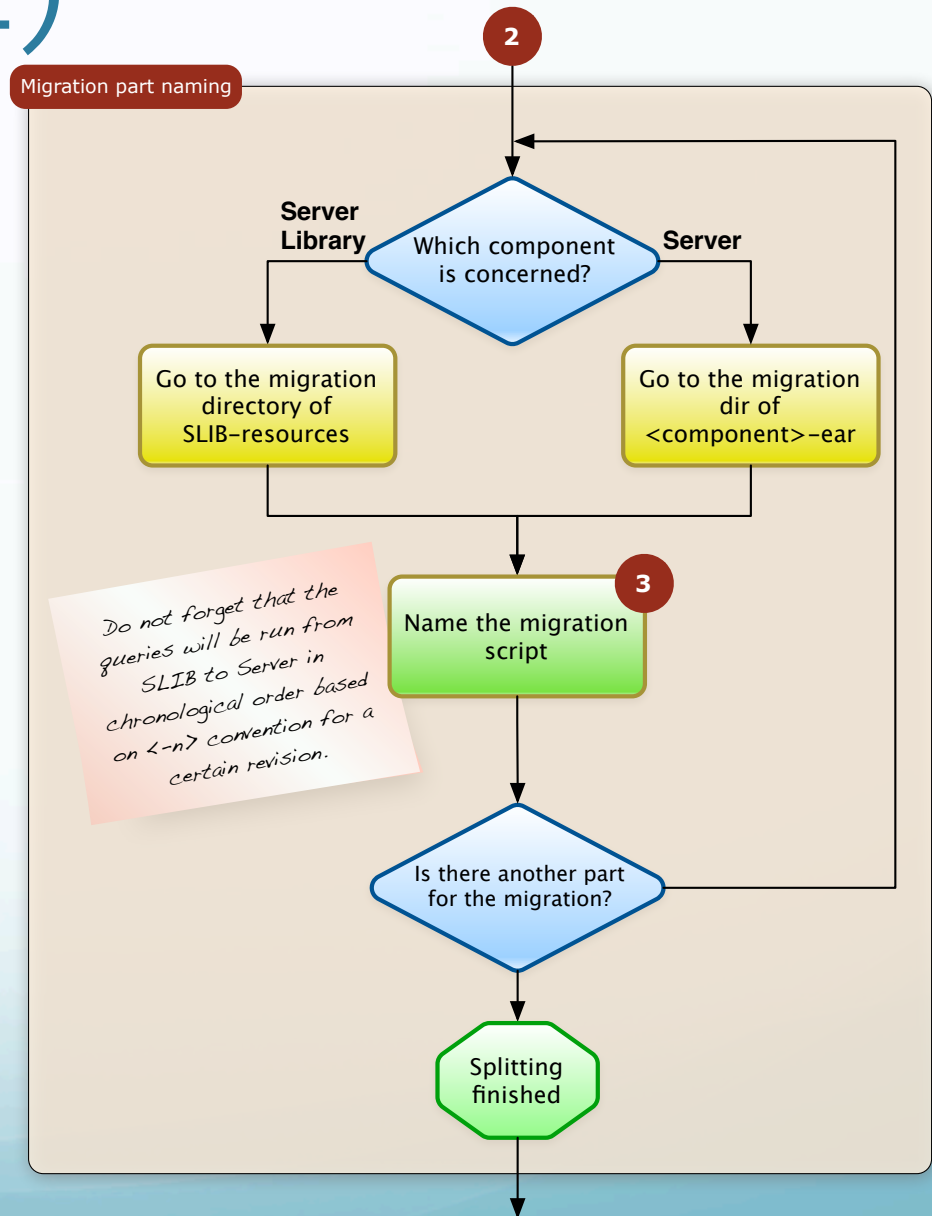
Find the revision (1)

- Check where the modification was done.
- Merge the modification into the trunk if necessary.
- Use the revision number where the code was committed or merged in the trunk.



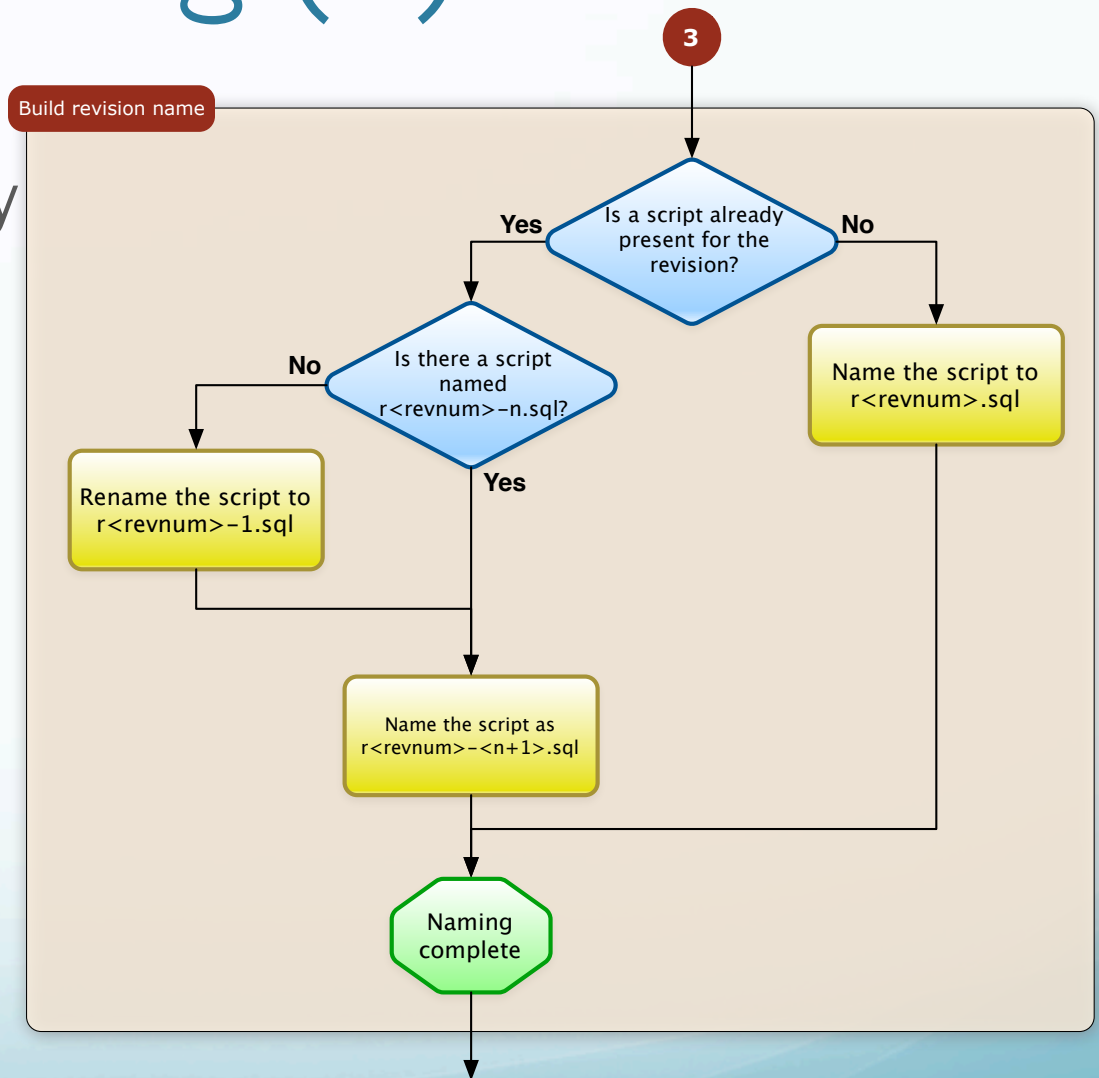
Part naming (2)

- Is the modification applicable to the server library?
- Name the migration scripts.
- Determine whether the script must be split because of constraint or data.



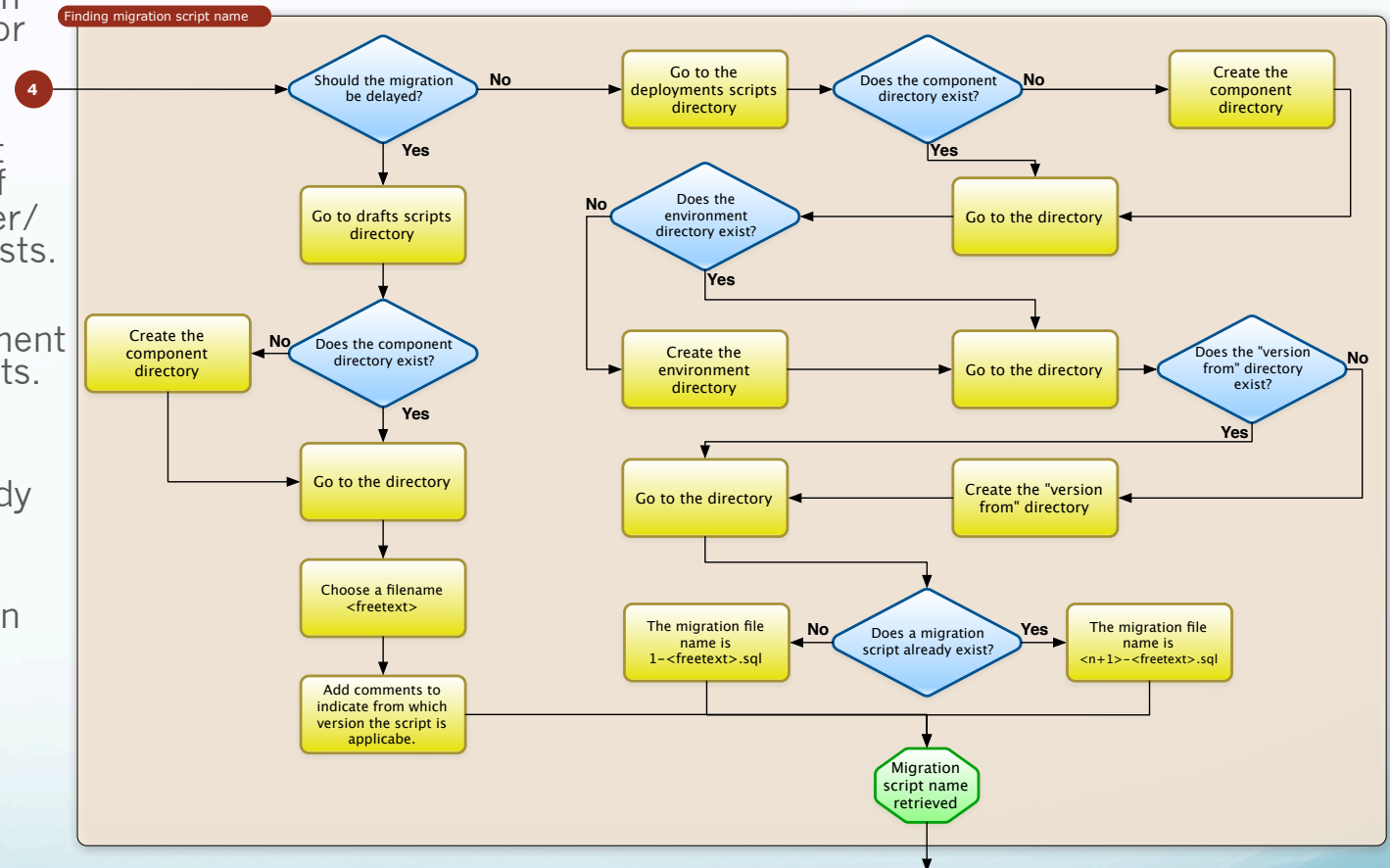
Revision naming (3)

- Check if a file for the same revision already exists.
- Check if a file with suffix “-n” already exists. Use the next number (n+1) if it does.
- Apply the naming convention.



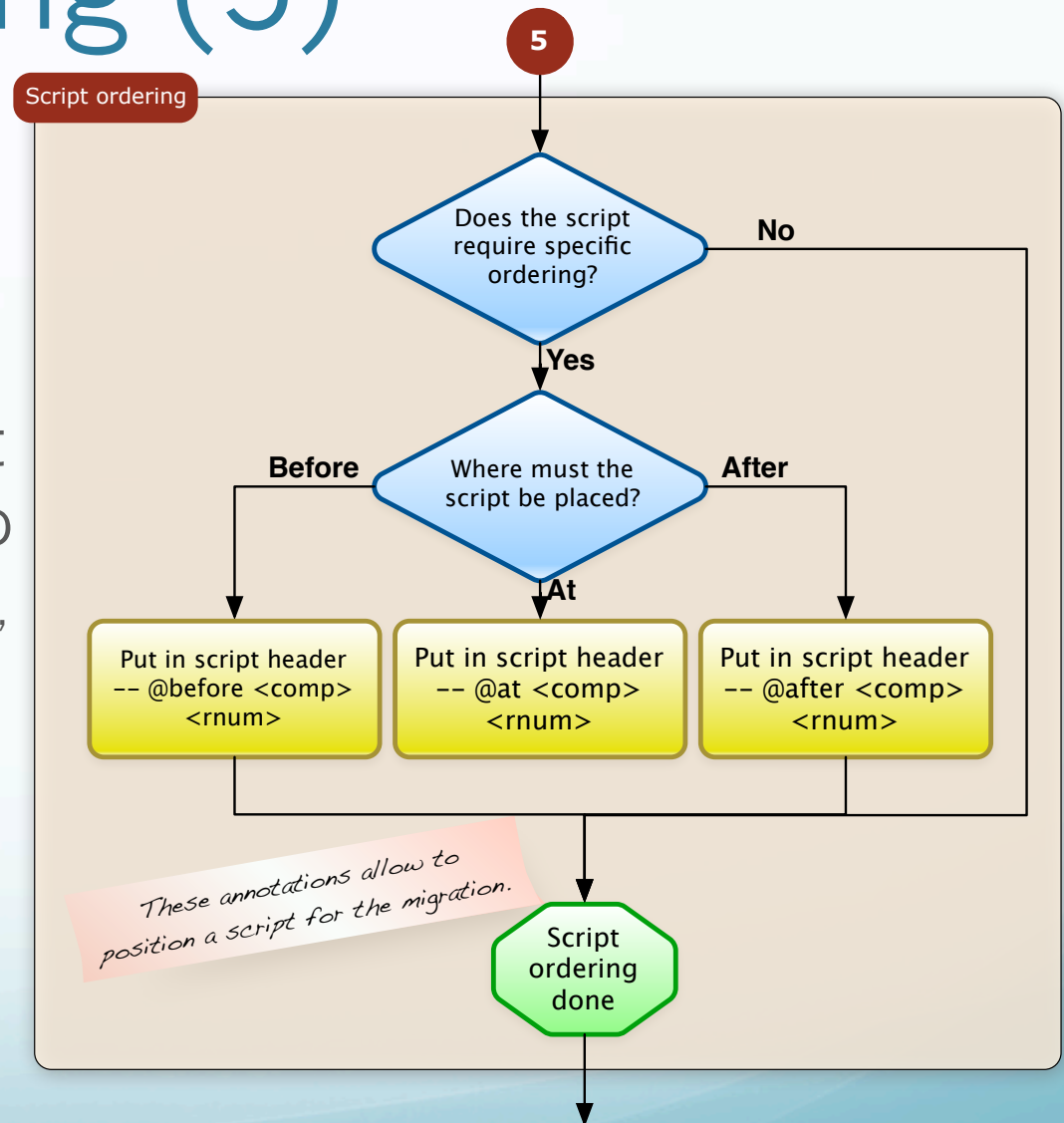
Contextual script naming (4)

- Check if the migration script must be kept for a later release.
- Go to the deployment directory and check if the component (server/database) already exists.
- Check if the environment directory already exists.
- Check if the “version from” directory already exists.
- Check if the migration script already exists.
- Apply the naming convention.



Script ordering (5)

- Check if the script must be run at a specific revision.
- Define when the script must be run relative to the revision (@at, @before, @after).
- Apply the annotation convention.



That's it

- A revision migration file always refers to a different revision (the one that introduced the modification).
- Any code/data modification must have its migration script(s) (atomicity/ immutability).
- Corrections on previous scripts should be in their own new scripts (immutability).
- Comments are the best friend of the Migration Manager. Add them to help him.
Ex: SELECT queries to complete contextual scripts.
- Avoid forgetting the database name in queries (or placeholder for the server library scripts).
- At this stage, a code part moved from a component to the server library is tricky to handle. Ask the Migration Manager for further details on how to handle the migration.
- Feel free to propose new ideas to improve the whole migration process.