Maverick - Commit Workflow with the EKL Extractor

M. Ben Hammich
12 Mar 2025

AIRBUS

Agenda

1. Context
2. Current Workflow
3. New Approach with the Extraction Tool
4. Commit Workflow with the Tool
5. Conclusion

Context

In our 3DEXPERIENCE development workflow, we used to commit the `.3dxml` files with the source code copied manually from Data SetUp.

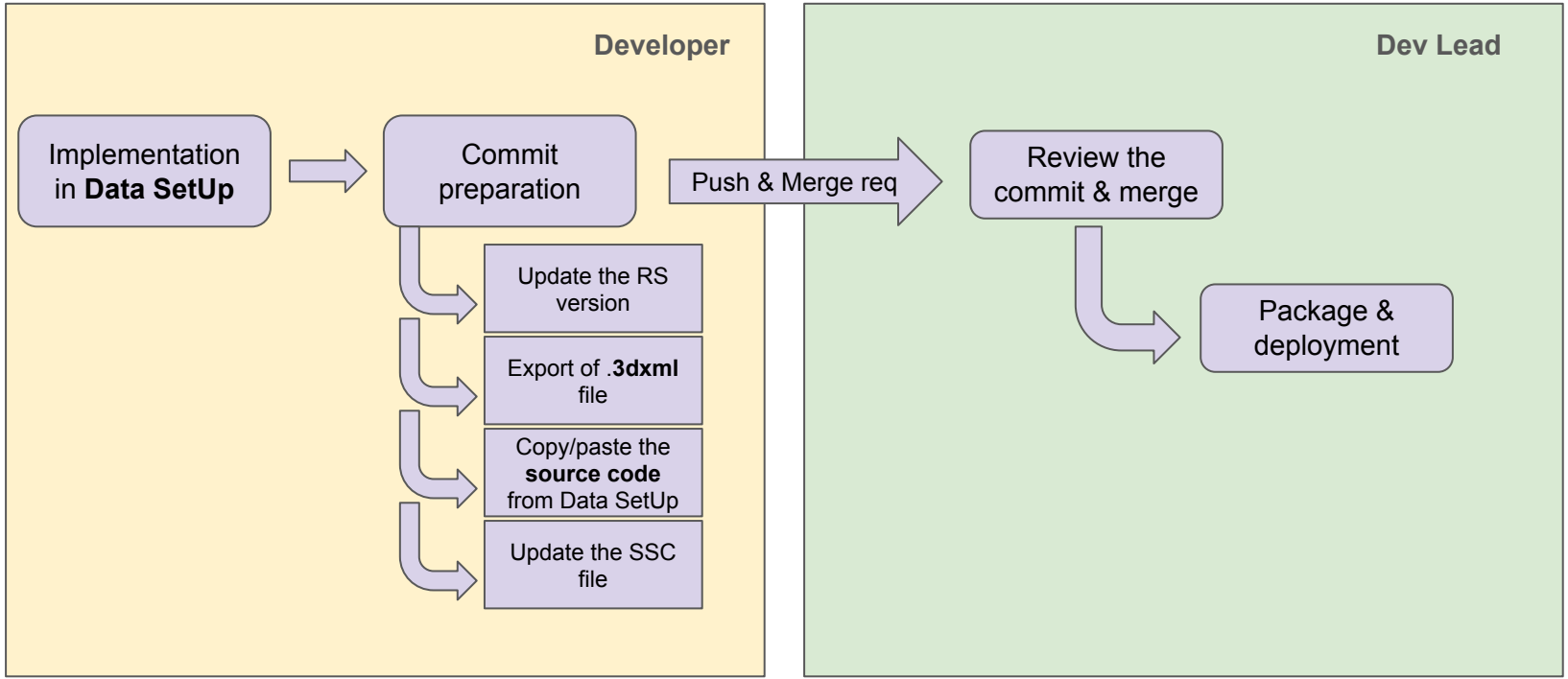
However, this process leads to some quality issues in our deliverables :

- **Lack of visibility**: Impossible to see what exactly changes in the `.3dxml`.
- **Risk of including unwanted modifications**: Another developer may modify the RS without the `.3dxml` exporter being aware.
- **Potential conflicts**: Risk of losing modifications if multiple developers work on the same RS without coordination.
- **Ineffective code review**: Impossible to precisely compare changes before committing.

Consequences

⚠️ Unintentional commits of modifications made by other developers.
⚠️ Difficulty in reverting changes or understanding the history of modifications.
⚠️ Risk of corruption or malfunction of RS due to uncontrolled changes.

Current WorkFlow



New Approach with the Extraction Tool

Objective:

- Enable developers to better secure the content of the exported `3dxml` files.
- Ensure that only the changes related to their own development are included.

Principle:

1. **Export the RS** from 3DEXPERIENCE.
2. **Use the tool** to extract all EKL code from the RS bound in the Collaborative Space.
3. **Compare differences** between versions to identify modifications.
4. **Commit only the relevant files**, excluding unintended changes.

Commit Workflow with the Tool

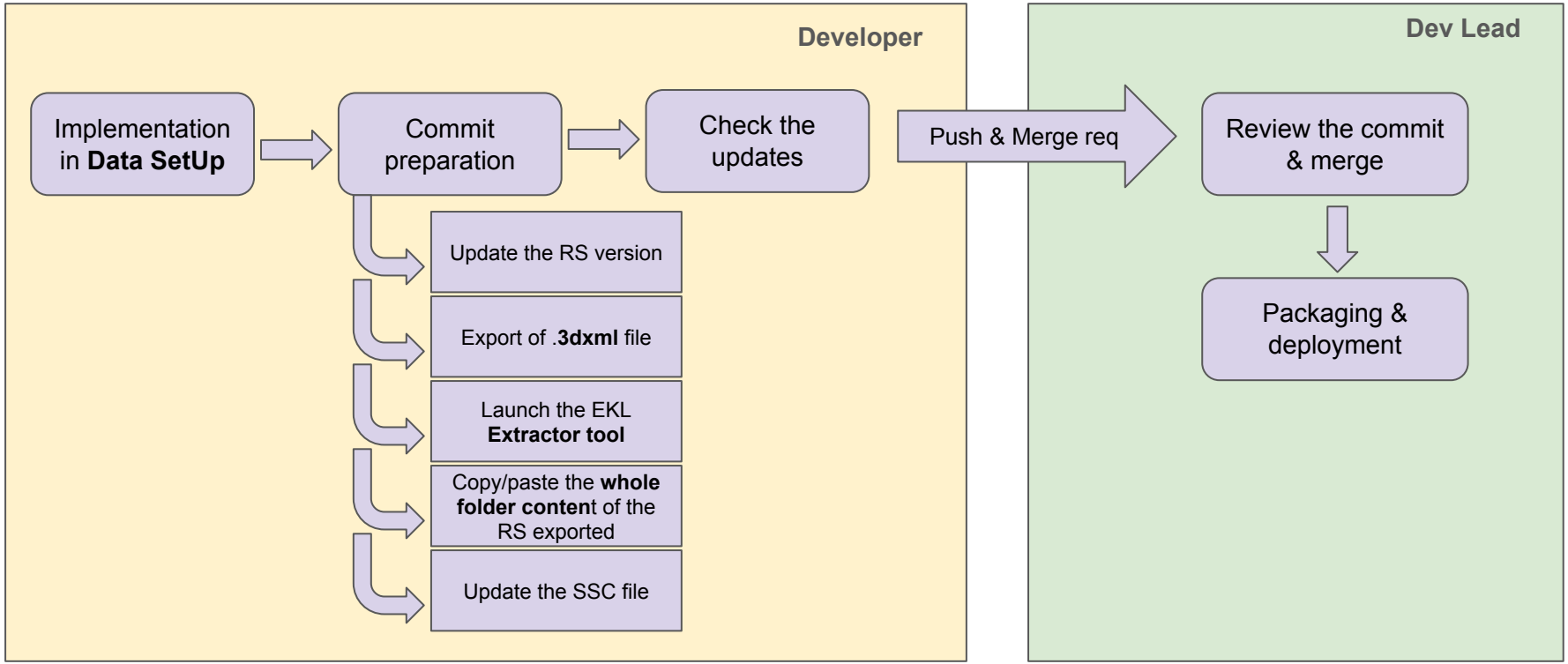
Detailed Steps:

1. **Export** the RS from 3DEXPERIENCE.
2. **Run the extraction tool** to retrieve all EKL code.
3. **Compare modifications** using a diff tool.
4. **Commit only the folder corresponding to the exported RS.**
5. **Ensure that the `.3dxml` file reflects only the intended changes.**

Benefits of This Approach :

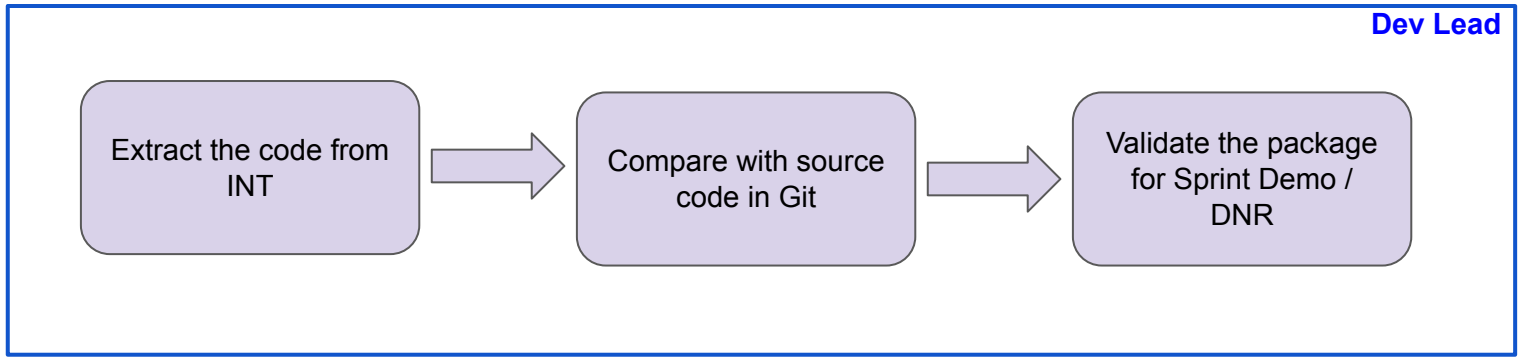
✅ **Visibility**: Each developer knows exactly what they are committing.
✅ **Fewer conflicts**: Prevents the unintentional integration of changes from other branches.
✅ **Better traceability**: Enables precise tracking of code evolution.

Commit Workflow with the Tool



Commit Workflow with the Tool

✅ **At the end of each sprint / Before DNR** : The Dev lead have to ensure the alignment of the code committed in Gitlab with the RseourceSets content in Integration platform



Conclusion

Adopting this tool and method will ensure a more rigorous commit process while reducing the risk of errors and conflicts. It is essential for every developer to make it a habit to use the tool before committing a `.3dxml` file.

Thank you !