



A Git-Driven Automated Workflow for Efficient ADaM & TLF Validation and Quality Control Documentation

PHUSE EU Connect 2025, Hamburg



Xavier Chénédé, Paulin Charliquart

Aim

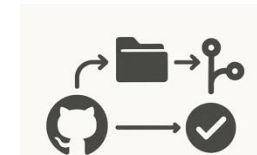
1. **Aim**
2. **Concept**
3. **Process**
 - Unit tests
 - GPP
 - Final QC reporting
4. **Conclusion**

- Implement a structured framework for automated deployment of ADaM/TLF Validation and QC Documentation

*Unit tests
for code
review*

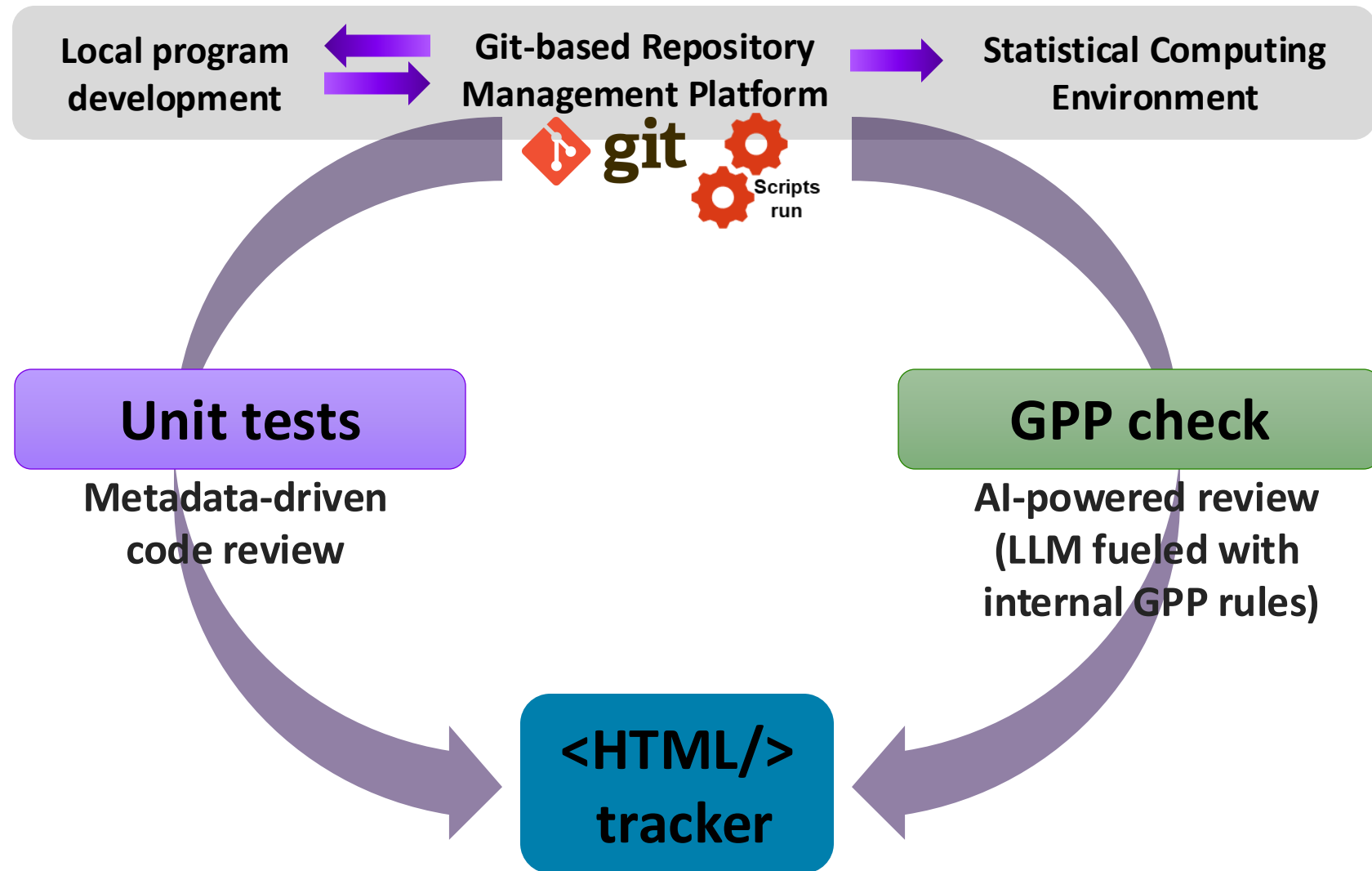
GPP checks

- Replace traditional QC workflows based on manual tracking with a modern, automated approach
- Leverage innovative technologies



Concept

1. Aim
2. **Concept**
3. Process
 - Unit tests
 - GPP
 - Final QC reporting
4. Conclusion



Process

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - GPP
 - Final QC reporting
4. Conclusion



1. **Build datasets/TLF**
2. **Develop unit tests locally**
 1. According to QC plan
 2. Functions specifically developed for the study
3. **Push and trigger workflows**
 1. Check style and execute all tests
 2. Merge with main branch
 3. Deploy on SCE, ie:
 1. Copy files to target
 2. Run Scripts and checks
 3. Generate tracker

Unit Tests

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - GPP
 - Final QC reporting
4. Conclusion

*Unit tests
for code
review*

Apply QC plan strategy

➔ Risk-based QC

Dataset	Variable
ADAE	All variables
ADTR	AVALC where PARAMCD= 'Confirmed Overall Response'
ADEX	DOSREDFL
ADEX	DOSESCFL
ADSL	All variables

Example based on phase 1 oncology trial

Unit Tests

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - GPP
 - Final QC reporting
4. Conclusion

*Unit tests
for code
review*

Apply QC plan strategy

➔ Risk-based QC

Dataset	Variable
ADAE	All variables
ADTR	AVALC where PARAMCD= 'Confirmed Overall Response'
ADEX	DOSREDFL
ADEX	DOSESCFL
ADSL	All variables



Create unit tests with following conventions

`test_that("Check ADEX.DOSREDFL", {` *#i.e., Dataset.Variable*

write the derivation

capture the reference data

code to compare

capture comparison result in .md file

`})`

Unit Tests

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - GPP
 - Final QC reporting
4. Conclusion

*Unit tests
for code
review*

- QC results file
➔ Example of adex.md

```
# Check ADEX.DOSREDFL
... Code ...
... print(qc_compare, row_limit = 100000) ...
... Output ...
... No issues were found! ...

# Check ADEX.DOSESCFL
... Code ...
... print(qc_compare, row_limit = 100000) ...
... Output ...
... Differences found between the objects! ...
```

- Tracking
➔ Merge with QC plan

Dataset	Variable	Results
ADEX	DOSESCFL	Test not passed
ADEX	DOSREDFL	Test passed

- Functions/TLF: equivalent process

Good Programing Practice

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - **GPP**
 - Final QC reporting
4. Conclusion



What did we try/explore?

CRITERIA	LOCAL LLM	GITHUB COPILOT CODE REVIEW
Model availability	✓ Downloaded	✓ Integrated
Customization	✓ Customizable	✓ Customizable
Setup complexity	✗ Tricky Setup	✓ Easy setup
Performance	✗ Slower	✓ Faster
Sensitive data handling	✓ Local control	✗ Risk of exposure
Code review	-	⚠ Supported



Not retained



Solution developed

Good Programming Practice

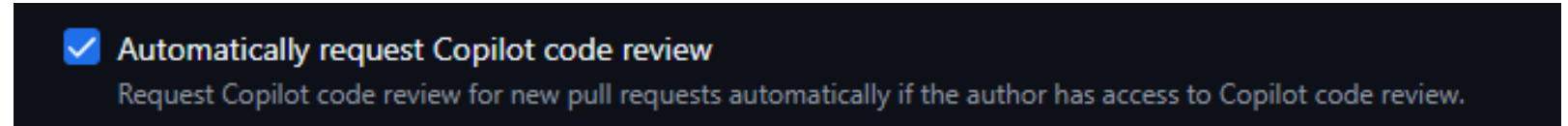
1. Aim
2. Concept
3. **Process**
 - Unit tests
 - **GPP**
 - Final QC reporting
4. Conclusion



Copilot code review



- GitHub Settings



- On change proposal



- Customization



Good Programming Practice

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - **GPP**
 - Final QC reporting
4. Conclusion



GPP check

Copilot code review



- Copilot-instructions.md → GPP rules (Excerpt)

...

Identify and clearly highlight bad practices, including but not limited to:

- *Absence of header.*
- *Using print instead of base::stop/warning/message.*
- *Using paste0 instead of file.path for constructing file paths.*

...

Good Programing Practice

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - **GPP**
 - Final QC reporting
4. Conclusion

GPP check

Copilot code review



- Some review examples
 - Typo
 - Header
 - Specific rule

New code submitted

```
for_full_review_adcrash.R x
Source on Save
1 # this is a script to build an ADaM dataset
2 dm <- read_data(file.path(SDD, "dm_f.xpt"))
3 ds <- read_data(paste0(SDD, "/ds_f.xpt"))
4
5
6
7 print("this a print")
8
```

What if I want to review my full code?

→ A change proposal review with copilot only checks on changes!

Good Programing Practice

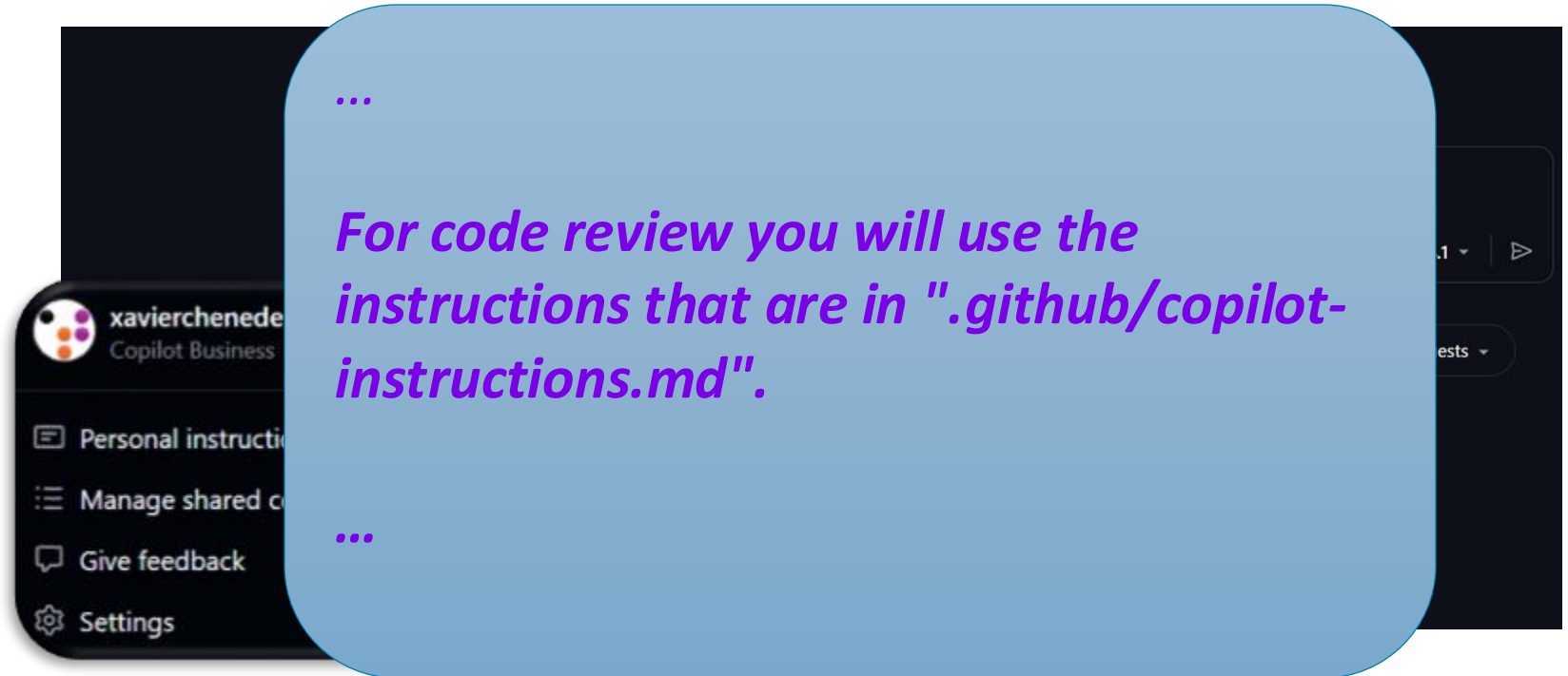
1. Aim
2. Concept
3. **Process**
 - Unit tests
 - **GPP**
 - Final QC reporting
4. Conclusion

GPP check

Copilot code review



→ GitHub Copilot coding agent ←



- Knows your GitHub repositories
- Customization with prompts

Good Programming Practice

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - **GPP**
 - Final QC reporting
4. Conclusion

GPP check

Copilot code review



- GitHub Copilot coding agent for full code review
→ Let's

Could you perform a comprehensive code review of all R scripts in the ADS/PGM folder (branch: Phuse2025) of the Sanofi-GitHub/spar-tedXXXX repository? Draft a GitHub issue. The issue will group

Comprehensive Code Rev...

Give feedback Create

Sanofi-GitHub/spar-tedXXXX

Add a title *

Comprehensive Code Review: ADS/PGM R Scripts (Branch: Phuse2025)

Add a description

Write Preview

This issue summarizes the results of a comprehensive code review of all R scripts in the ADS/PGM folder (branch: Phuse2025) according to the repository's Copilot Code Review Instructions (see [.github/copilot-instructions.md](#)).

Each script is reviewed for code quality, maintainability, adherence to standards, and best practices, including R style and ADaM-specific requirements. Problems are grouped by file and by category when common, with direct code references and actionable suggestions.

File: adae.R

- **Strengths:**
 - Good documentation and revision history.
 - Modular, pipeline-based data processing (tidyverse approach).
 - Extensive metadata handling and clear variable management.
- **Issues:**
 - **Use of print for status/info:** Prefer `message()` or `cli` functions for user feedback (see lines 37, 102, 477, etc.).
 - **Potential use of deprecated or non-idiomatic code:** Use `file.path` for all file paths (e.g., in reading/writing files).
 - **Use of paste or paste0 for file paths:** Use `file.path` for better cross-platform compatibility (see lines 125-128, 705, 714).
 - **Hardcoded values:** Some magic numbers and strings are used (e.g., visit numbers, codes) that could be set as constants for clarity and maintainability.
 - **Repeated code patterns:** Repetitive derivation steps and message printing could be wrapped in helper functions.

Branch



Good Programming Practice

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - **GPP**
 - Final QC reporting
4. Conclusion

GPP check

Copilot code review



- GitHub Copilot coding agent for full code review
→ Issue creation:

Comprehensive Code Review: ADS/PGM R Scripts (Branch: Phuse2025) - Edit New issue

COPILOT FULL REVIEW #95

Open Task

xavierchenede opened 2 minutes ago

This issue summarizes the results of a comprehensive code review of all R scripts in the `ADS/PGM` folder (branch: Phuse2025) according to the repository's Copilot Code Review Instructions (see [.github/copilot-instructions.md](#)).

Each script is reviewed for code quality, maintainability, adherence to standards, and best practices, including R style and ADaM-specific requirements. Problems are grouped by file and by category when common, with direct code references and actionable suggestions.

File: `adae.R`

- **Strengths:**
 - Good documentation and revision history.
 - Modular, pipeline-based data processing (tidyverse approach).
 - Extensive metadata handling and clear variable management.
- **Issues:**
 - Use of `print` for status/info: Prefer `message()` or `cli` functions for user feedback (see lines 37, 102, 477, etc.).
 - Potential use of deprecated or non-idiomatic code: Use `file.path` for all file paths (e.g., in reading/writing files).
 - Use of `paste` or `paste0` for file paths: Use `file.path` for better cross-platform compatibility (see lines 125-128, 705, 714).
 - Hardcoded values: Some magic numbers and strings are used (e.g., visit numbers, codes) that could be set as constants for

Assignees: xavierchenede

Labels: enhancement

Type: Task

Projects: No projects

Milestone: No milestone

Relationships: None yet

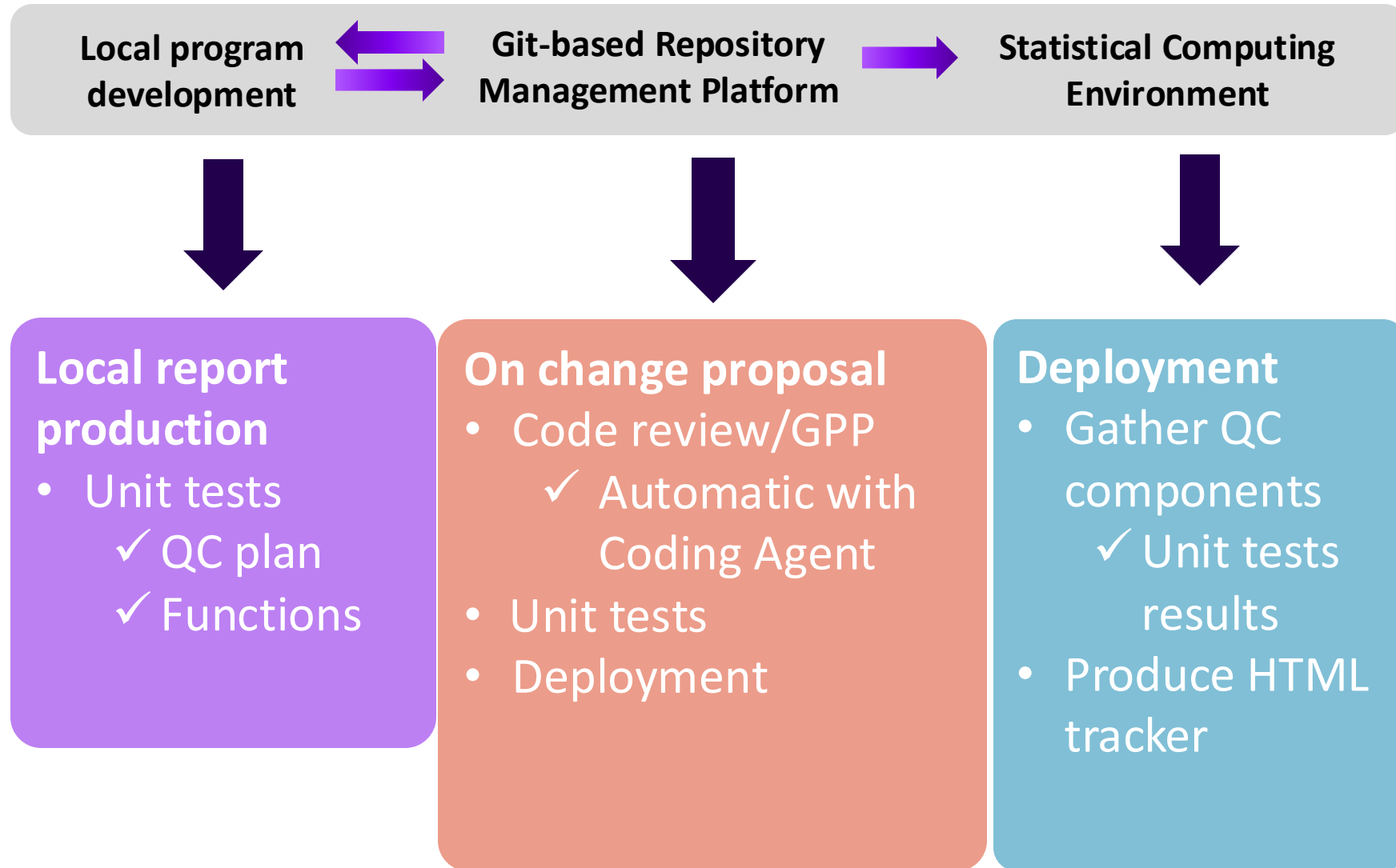
Development

Final QC reporting

1. Aim
2. Concept
3. **Process**
 - Unit tests
 - GPP
 - **Final QC reporting**
4. Conclusion



Tracker



Final QC reporting

- 1. Aim
- 2. Concept
- 3. Process
 - Unit tests
 - GPP
 - Final QC reporting
- 4. Conclusion



Tracker

- HTML dashboard

STUDYID

Overview

Specifications

ADaM-QC

Functions-QC

Functions-QC

Test		Result	File		Execution.Date	
Flag for confirmed_pd		Test passed	test-mm_efficacy_functions.R		2025-09-22 14:21:42	
last_sd		Test passed	test-mm_efficacy_functions.R		2025-09-22 14:21:42	
keep_sqoresp		Test passed	test-mm_efficacy_functions.R		2025-09-22 14:21:42	
Objective Responder Flag		Test passed	test-mm_efficacy_functions.R		2025-09-22 14:21:42	
Disease control Flag		Test passed	test-mm_efficacy_functions.R		2025-09-22 14:21:42	
Clinical Benefit Flag _ no SD flag duration needed		Test passed	test-mm_efficacy_functions.R		2025-09-22 14:21:42	
Clinical Benefit Flag _ SD flag duration needed		Test passed	test-mm_efficacy_functions.R		2025-09-22 14:21:42	
ADMH	ANYFL	Self-QC				
ADEX	All variables	Self-QC				
ADEX	DOSEAT	Self-QC	Test not passed	test-adex.R	Xav	2025-09-22
ADEX	DOSCUMA	Self-QC	Test passed	test-adex.R	Xav	2025-09-22

CONCLUSION

1. Aim
2. Concept
3. Process
 - Unit tests
 - GPP
 - Final QC reporting
4. **Conclusion**

This work is the first step towards a more integrated project aiming to gather even more QC elements.

- **Next steps**
 - Logs/SDTM
 - Dataset comparison
 - Editable standalone tracker
- **Challenges**
 - Use of HTML with JavaScript
 - Bring people into a new mindset
 - Be careful in the use of LLM, let's be critical!

This workflow reduces repetition of manual intervention, minimizes human error, and substantially reduces turnaround time from data availability to delivery, all while maintaining rigorous quality standards.

Thank *you*

sanofi

Contacts:

Xavier.chenede@sanofi.com

Paulin.charliquart@sanofi.com