Paper PP23

# How can agile R Package validation workflows be achieved?

## Turning role matrices into transparent, agile validation processes.

Sophia Stahl-Toyota, HMS Analytical Software GmbH, Heidelberg, Germany
Maximilian Kreienbaum, HMS Analytical Software GmbH, Heidelberg, Germany
Dr. Christoph Frank, HMS Analytical Software GmbH, Heidelberg, Germany

**ABSTRACT**
Getting from feature requests or bug reports to the next validated package release in an agile fashion is the vision of many R package product owners, developers, validation managers and users. Add further colleagues from quality departments, deployment pipeline operations and management to the list of stakeholders and we see that the topic heavily relies on alignment of people and processes in addition to tools. We present how the tooling landscape for managing software development can be set up with sophisticated use of the technical possibilities offered by platforms such as GitLab™, Jira®, Microsoft Azure DevOps and GitHub. They include workflows, issue statuses, labels, scoped labels, tags, components and timing related iterations, sprints, and milestones. However, efficient configuration and use of those options is the art to master for effective tracking of development and validation activities that involve many review steps.

**KEYWORDS**
Validation, agile, R package, workflows, risk assessment, traceability, specifications

## INTRODUCTION

Challenge: Software validation requires coordination between experts, developers, product owners and validation managers. Many teams still use colorful Excel sheets with “x” marks – often without legends, causing confusion.

Solution: Responsibilities from validation or change plans are tracked directly in common development platforms. This enables iterative reviews, transparency, and avoids static Excel lists. The approach is illustrated with an R package validation example, but it applies broadly to collaborative validation tasks.

Benefit: Less coordination effort – focus shifts from tracking status to discussing content. Teams gain transparency on progress and stay motivated by using familiar tools. Iterative reviews ensure that specifications and validation artefacts and be continuously refined during development.

## THE WORKFLOW BY EXAMPLE

### SCENARIO

This example scenario is intentionally chosen to spark the discussion on potential feature extensions of existing R package validation aiding tools, such as {valtools} and {thevalidatoR}.
The users want to convert contents from an R-code file to an excel file. The input consists of test cases defined in a {testthat} setup. The desired output is an excel file containing the test ids and descriptions.

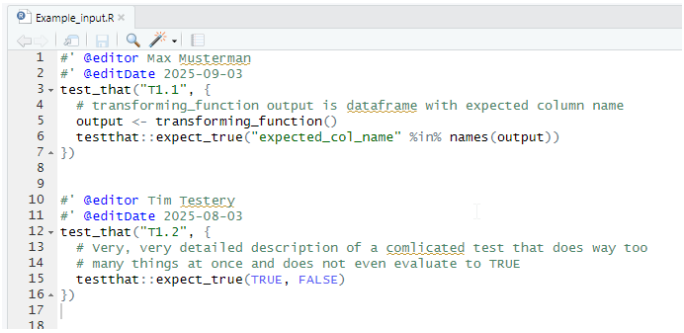

Figure 1. Example input.

| | A | B |
|---|---|---|
| 1 | Test_ID | Test_Description |
| 2 | T1.1 | transforming_function output is dataframe with expected column name |
| 3 | T1.2 | Very, very detailed description of a comlicated test that does way too many things at once and does [cutoff] |
| 4 | | |

Figure 2. Example output.

### THE STARTING SETUP

This example demonstrates how an iterative review workflow could function in practice. The principle applies broadly to multiple validation artefacts. Among the deliverables that could be derived from the Computerized System Life Cycle Phases (ISPE GAMP® 5, 2022) as a minimal list are Specifications, Risk Assessment, Tests and Traceability Matrix. Here, we show two examples for a review iteration for functional specifications (FS) and risk assessment (RA). The concept could also be applied to other deliverables, such as test case collections.

Screenshots from GitHub illustrate how responsibilities and approvals are tracked step by step.
Typical status columns in an agile board include TODO, IN PROGRESS and DONE.

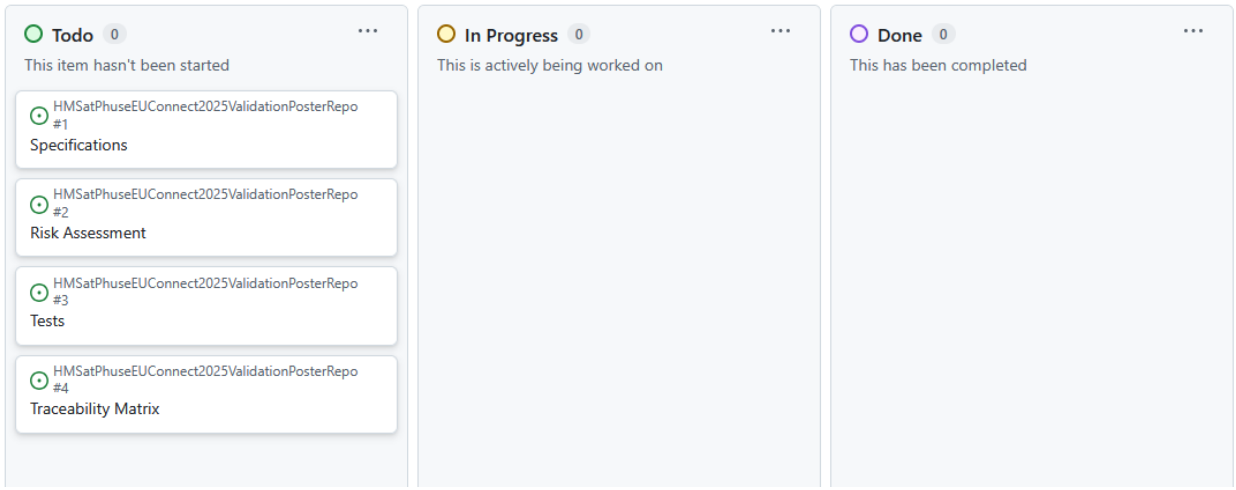

Figure 3. Start setup in board view.

Typical roles of stakeholders involved in producing a validated R Package may include Experts (E), Developers (D), Product Owners (P), Validation Managers (M) and Testers (T). Beyond the functionality itself that the product owner envisions, the developers produce, the testers test, and the experts use, the validation deliverables should aid the team in the development process by structuring the communication flow. The activities performed on the validation deliverables by the stakeholders move from writing / authoring / drafting to reviewing to approving. The activities can be performed in parallel by multiple stakeholders and at the end the “approved by” flag should be true for each assigned role. Therefore, tagging or labeling systems should be used to track the activities that allow for multiple active tags. The transitions between labels can also be automated with workflows, such as Github Actions.

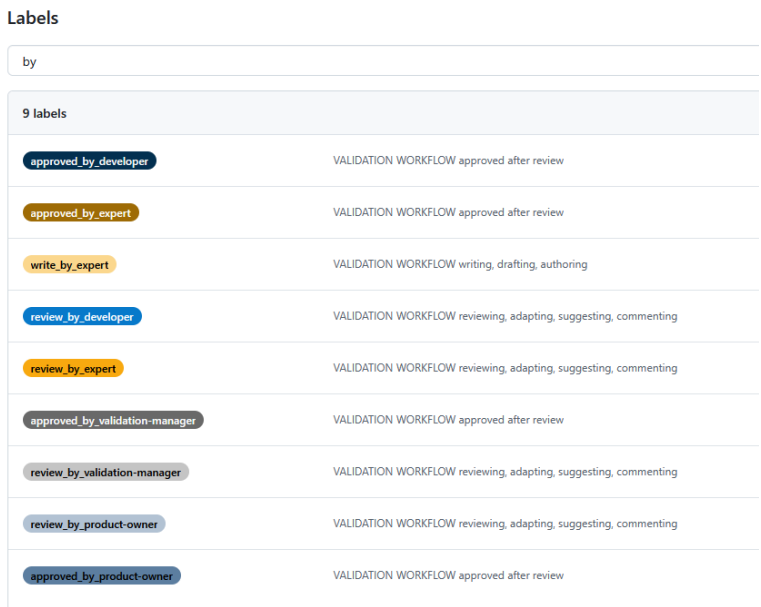

Figure 4. Labels used across examples.

**REQUIREMENTS / FUNCTIONAL SPECIFICATIONS**

Overview of Review Workflow for this example of Functional Specifications (FS).

Experts draft, developers review and suggest a few clarifications. When experts and developers have found agreement on the draft, the product owner approves and the validation manager sets the status to Done.

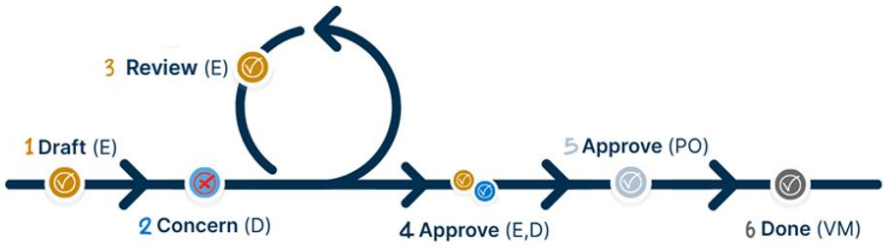


Figure 5. FS Overview of review workflow example

FS Step 1 Draft by Expert

The experts create the first draft of the Specifications. They write two requirements, Req1: “Produce excel with two columns from comments in R code”. Req2: “Limit values to 100 characters.”

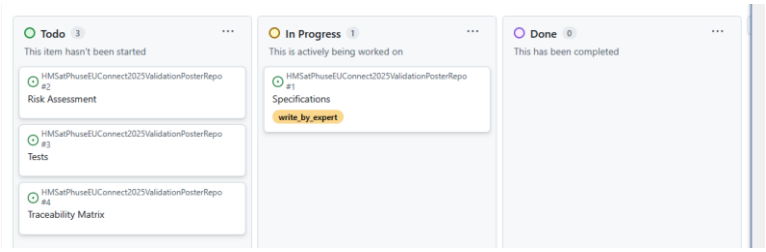


Figure 6. FS Step 1 Board view.

FS Step 2 Concern by Developer

The developers have some questions, such as “What should the names of the columns be?” and “What should happen if the value is longer than the threshold?”. They put comments in the document and assign the ticket back to be reviewed by the experts.

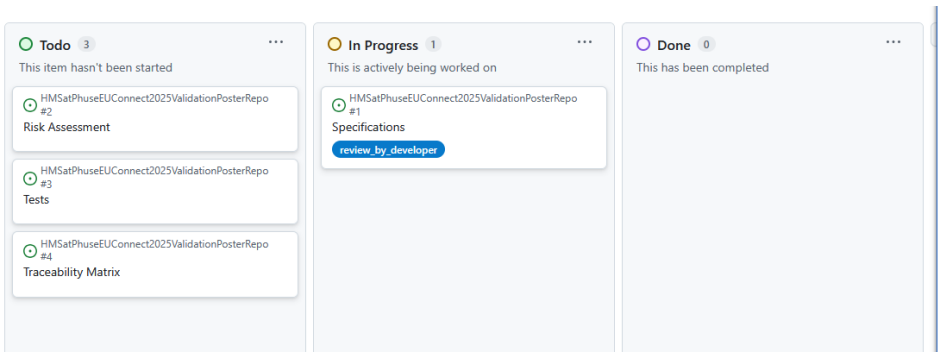


Figure 7. FS Step 2 Board view.

FS Step 3 Review by Expert

The experts work on the document, rephrasing the requirements as Req1: “Produce excel with two columns ‘Test_ID’ and ‘Test_Description’ from test_that functions”. Req2: “Limit values to 100 characters. If exceeded, cut off and add suffix [cutoff].” Then they assign it back to be reviewed by the developers.

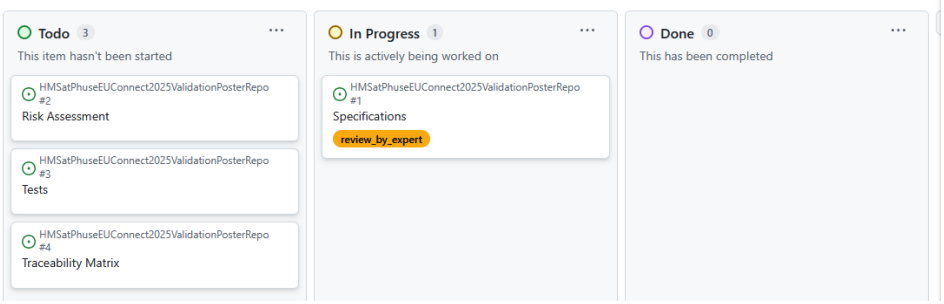


Figure 8. FS Step 3 Board view.

FS Step 4 Approval by Expert and Developer

The experts and developers approve.

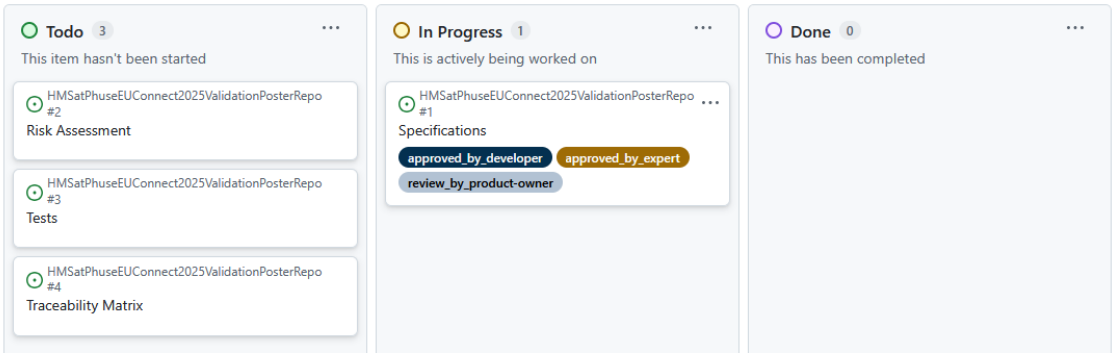


Figure 9. FS Step 4 Board view.

FS Step 5 Approval by Product Owner

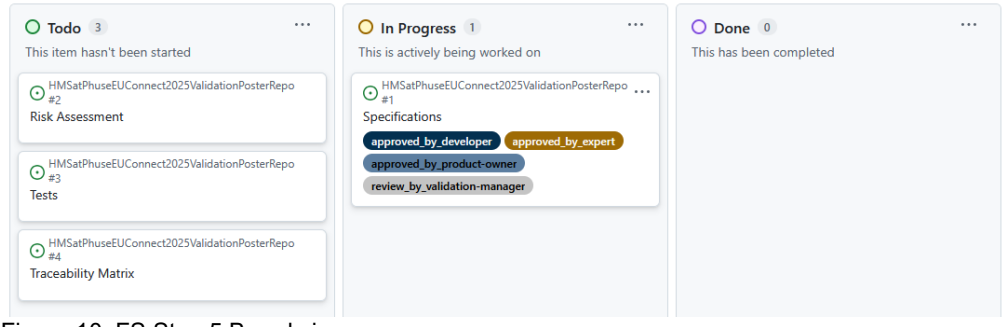


Figure 10. FS Step 5 Board view.

FS Step 6 Approval by Validation Manager

Finally, all stakeholders have approved, and the ticket status is set to done by the validation manager and the formal round of signatures could be started in the dedicated system.
Essentially, this chain of approval mimics the formal signatures that take place in a document management system, but support the work in progress review process that comes before the formal round of reviews. The cycle in the software development platform could be repeated for small chunks of scope that are released together in a formal validation cycle at a later point in time.

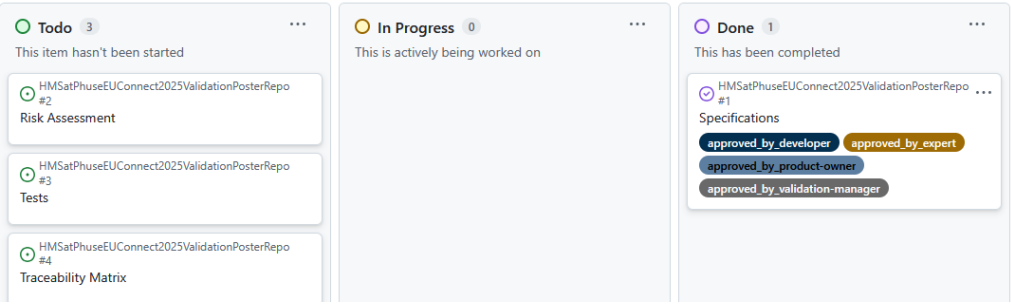

Figure 11. FS Step 6 Board view.

**RISK ASSESSMENT EXAMPLE ITERATIVE REVIEW WORKFLOW**

Overview of Review Workflow for this Example Risk Assessment (RA)

Experts draft, developers review, the product owner may raise concerns, and approvals can be revoked and repeated. Finally, the validation manager sets the status to Done.

Risk assessments are drafted, reviewed, and refined after concerns. Approvals may be revoked and the cycle restarted until all stakeholders agree. Finally, the validation manager sets the status to Done. This ensures transparency and avoids late surprises before formal validation.

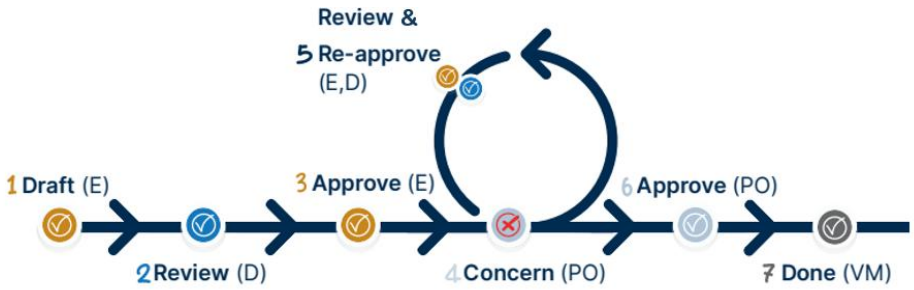


Figure 12. RA Overview of review workflow example

RA Step 1 Draft by Expert

The experts write the first draft of the risk assessment associated with the functionality. A hazard if the first requirement is not correctly implemented could be “Values are jumbled leading to wrong mapping of IDs to descriptions.” They estimate the severity to be high, the probability of occurrence medium and the probability of detection medium, resulting in an overall risk priority score of high. If the second requirement is not fulfilled, the consequence could be “Too long values render the excel table not readable”. The related severity would be medium, probability of occurrence also medium, but highly probable detection, leading to a risk priority score of low.

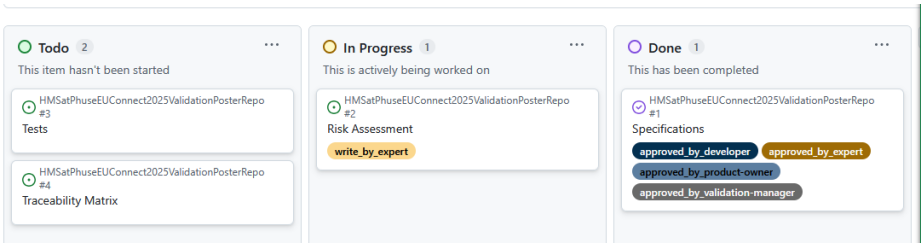


Figure 13. RA Step 1 Board view.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Functional_risk_assessment_ID | Functional_risk_assessment_Description | Severity | Occurrence | Detection | Risk_Priority |
| 2 | FRA1 | Values are jumbled leading to wrong mapping of IDs to descriptions. | high | medium | medium | high |
| 3 | FRA2 | Too long values render the excel table not readable. | medium | medium | high | low |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

Figure 14. RA Step 1 Document view.

RA Step 2 Review by Developer

They assign the risk assessment to be ready for review by the developers, who approve.

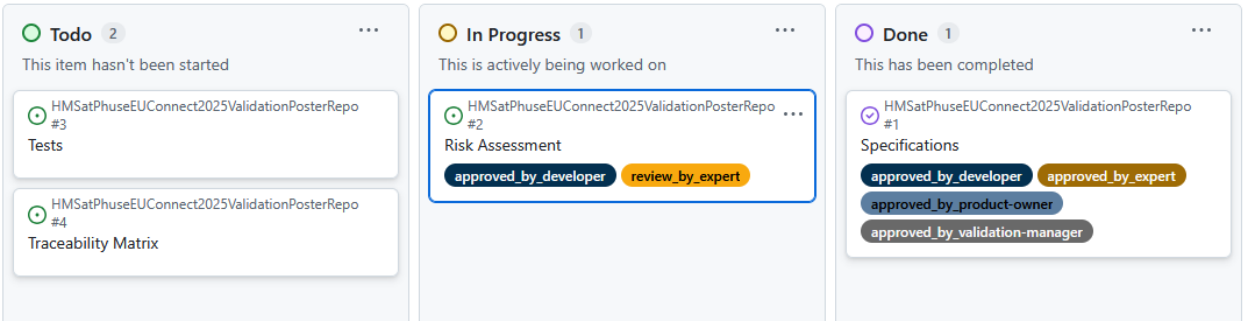


Figure 15. RA Step 2 Board view.

RA Step 3 Approval by Expert

The experts also approve and set it to be reviewed by the product owner.

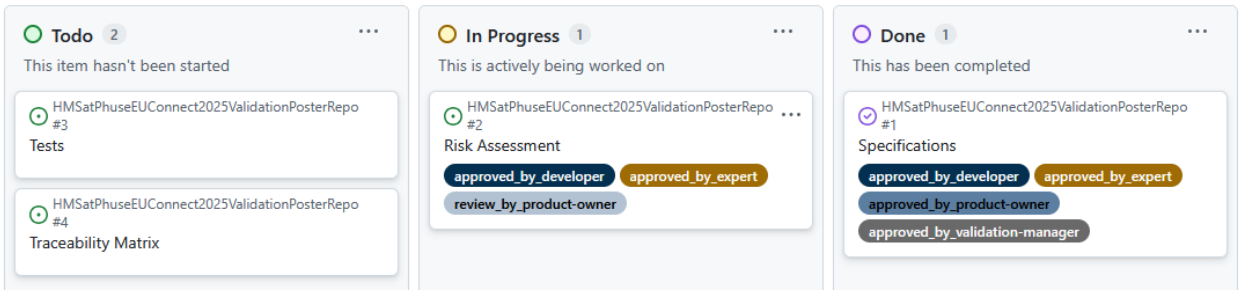


Figure 16. RA Step 3 Board view.

RA Step 4 Concern by Product Owner

The product owner is of the opinion that a lengthy description would impose not only medium but high severity and therefore changes the overall risk score from low to medium. The previous expert and developer approvals are removed by the product owner because they need to review his changes.

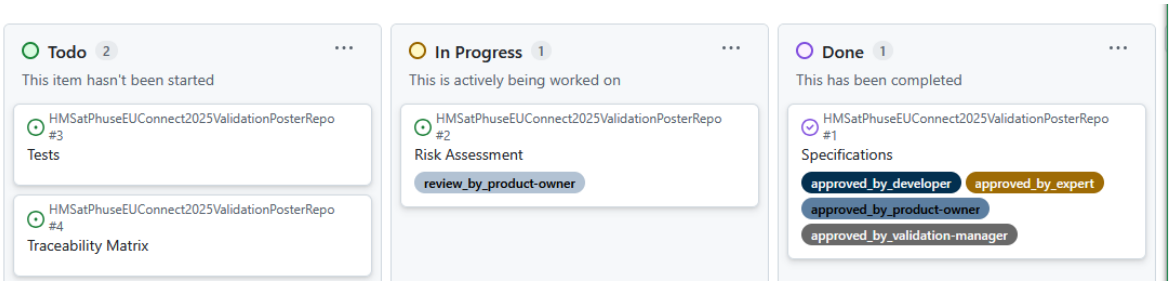


Figure 17. RA Step 4 Board view.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | **Functional_risk_assessment_ID** | **Functional_risk_assessment_Description** | **Severity** | **Occurrence** | **Detection** | **Risk_Priority** |
| 2 | FRA1 | Values are jumbled leading to wrong mapping of IDs to descriptions. | high | medium | medium | high |
| 3 | FRA2 | Too long values render the excel table not readable. | high | medium | high | medium |
| 4 | | | | | | |
| 5 | | | | | | |

Figure 18. RA Step 4 Document view.

RA Step 5 Review and re-approval by Expert and Developer

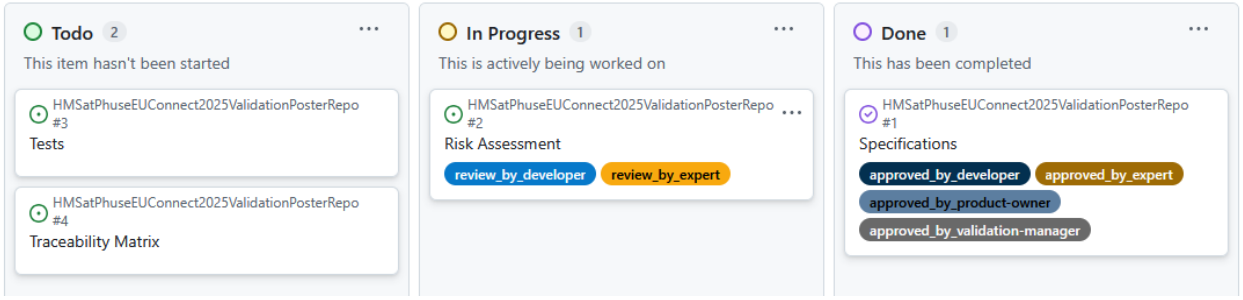


Figure 19. RA Step 5 Board view.

RA Step 6 Approval by Product Owner

Developers and experts approve, then the product owner also approves and sets it for review by the validation manager.

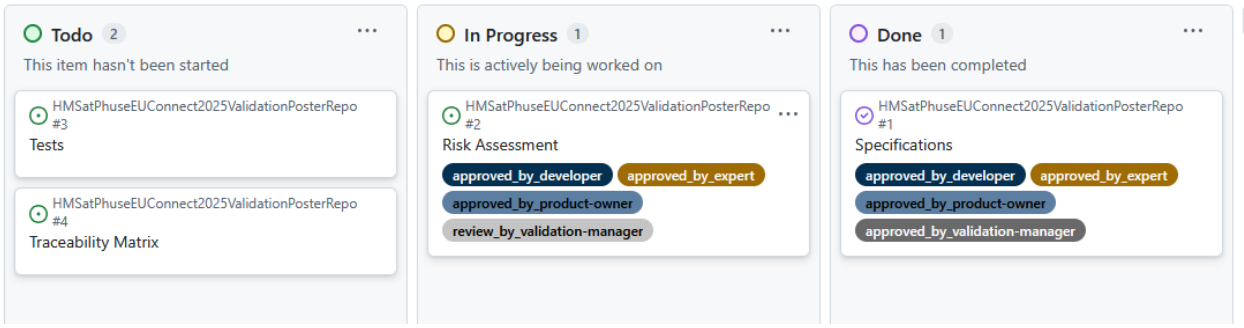


Figure 20. RA Step 6 Board view.

RA Step 7 Approval by Validation Manager

Finally, all stakeholders have approved the items in question and the status is set to done by the validation manager.

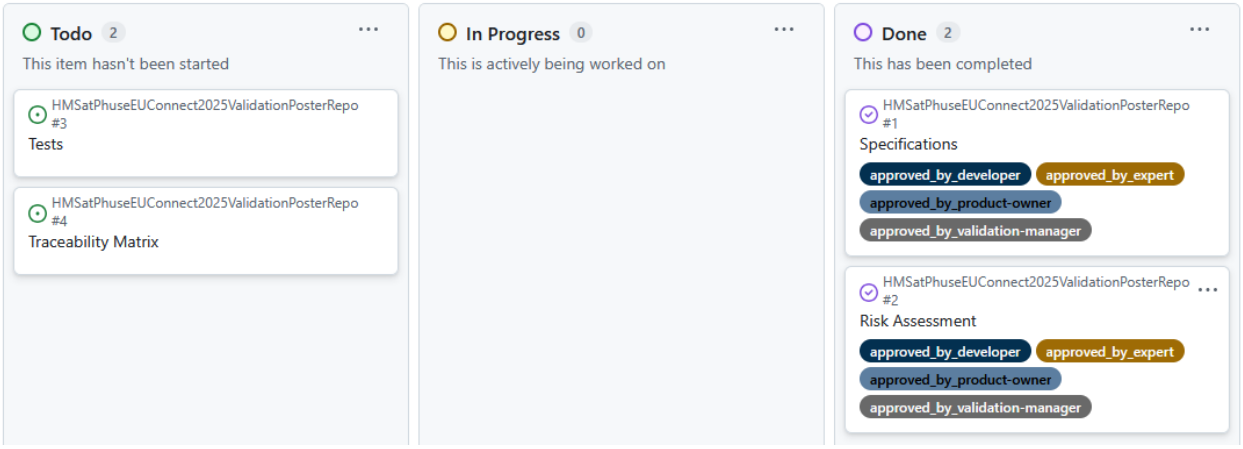

Figure 21. RA Step 7 Board view.

## CONCLUSION

This chain of approval mimics the formal signatures that take place in a document management system but supports the work in progress review process that comes before the formal round of reviews. The cycle in the software development platform could be repeated for small chunks of scope that are released together in a formal validation cycle at a later point in time. Keeping the validation documents in sync close to the development activities allows for the true advantage of agile workflows to enable fast iterations between demos and incorporation of feedback. Don't let the document review process hold you back from building what you truly want - invest in a good process for good products.

## REFERENCES


https://github.com/phuse-org/valtools, last accessed 2025-10-28

https://github.com/insightsengineering/thevalidatoR, last accessed 2025-10-28

https://testthat.r-lib.org/, last accessed 2025-10-28

https://docs.github.com/en/actions/tutorials/manage-your-work/add-labels-to-issues, last accessed 2025-10-28

International Society for Pharmaceutical Engineering (ISPE). (2022). GAMP® 5: A Risk-Based Approach to Compliant GxP Computerized Systems (2nd ed.), https://guidance-docs.ispe.org/doi/book/10.1002/9781946964571

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Author Name: Sophia Stahl-Toyota
Company: HMS Analytical Software GmbH
Address: Gruene Meile 29, 69115 Heidelberg, GERMANY
Work Phone: +49 6221 6051-530
Email: sophia.stahl-toyota@analytical-software.de
Website: https://www.analytical-software.de

Author Name: Maximilian Kreienbaum
Company: HMS Analytical Software GmbH
Address: Gruene Meile 29, 69115 Heidelberg, GERMANY
Work Phone: +49 6221 6051-518
Email: maximilian.kreienbaum@analytical-software.de

Author Name: Dr. Christoph Frank
Company: HMS Analytical Software GmbH
Address: Gruene Meile 29, 69115 Heidelberg, GERMANY
Work Phone: +49 6221 6051-134
Email: christoph.frank@analytical-software.de

Website: https://www.analytical-software.de