Manifesto

Information architecture for agile projects

We are uncovering better ways of managing information in agile projects by doing it and helping others do it.

Author: Reiner Schindler
Organisation: MORe® Framework / projektkram
Version: v1.1 · August 2026
Source: more-framework.org/framework

Through this work we have come to value:

- **Collaboration with stakeholders** **over** formal documentation
- **Continuous refinement of requirements** **over** one-off specifications
- **Just-in-time analysis** **over** complete upfront definition
- **Shared understanding in the team** **over** individual responsibility
- **Usable value** **over** perfect requirements

That is, while there is value in the items on the right, we value the items on the left more.

MORe applies this agile mindset to the information architecture of modern projects.

> All values serve one common goal: to structure project knowledge traceably and to keep it available over time.

MORe® and existing standards

MORe® does not replace existing standards such as IREB, arc42 or Model Cards. MORe® connects their artefacts through leading locations, semantic relationships and traceable baselines. This is where traceability emerges – not at the end through reconstruction, but during the work.

The Requirement Cycle sets the rhythm. The information graph preserves the domain context.

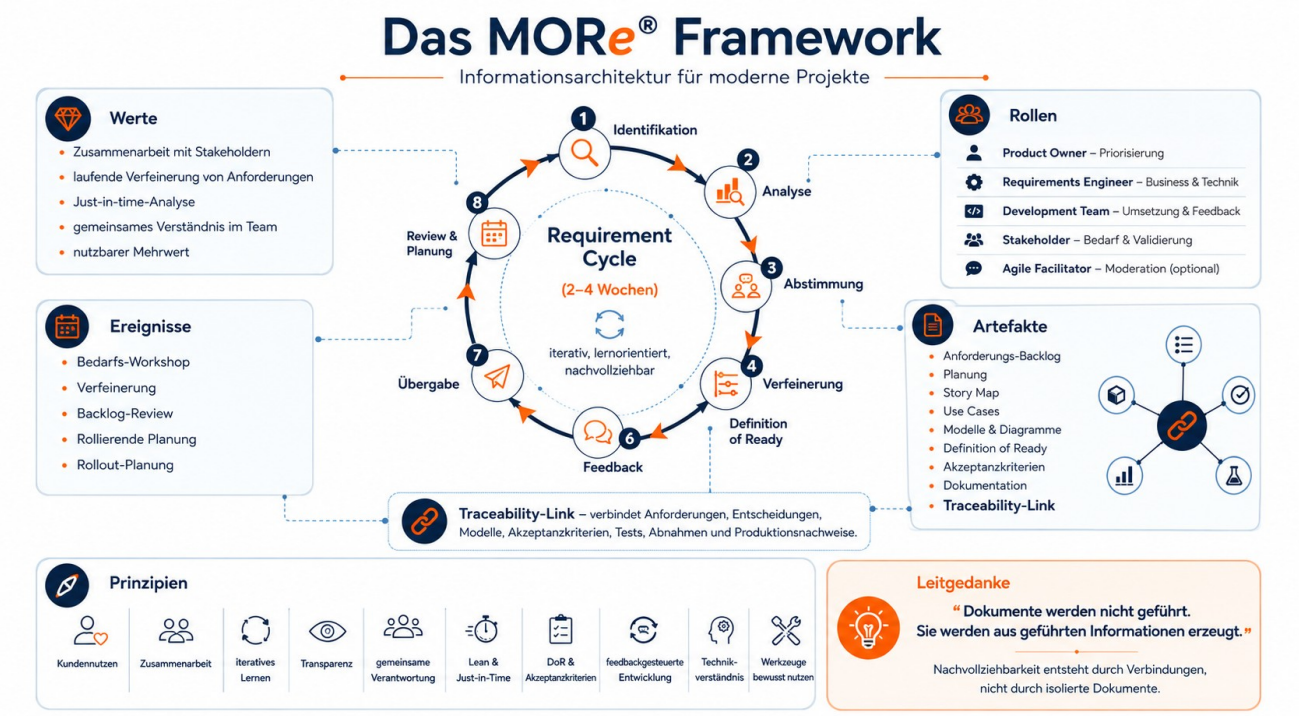

Figure 1: The MORe® Framework at a glance.

Principles

1. Focus on customer value

Requirements are prioritised according to business value and reviewed regularly.

2. Early and continuous collaboration

Stakeholders are actively involved in defining, validating and refining requirements.

3. Iterative learning

Requirements emerge, mature and change during the process. Change is understood as an opportunity, not as a disruption.

4. Transparency

Requirements are documented so that they are understandable, visible and traceable for everyone involved in the project.

5. Shared responsibility

Business departments, product owners, developers and requirements engineers contribute to understanding together.

6. Lean and just-in-time

Requirements are elaborated at the right level of detail and at the right time. Not earlier, not later.

7. Definition of Ready and acceptance criteria

Requirements are ready for implementation only once clear, verifiable criteria have been defined.

8. Feedback-driven development

Requirements are regularly reviewed and adjusted on the basis of results, feedback and data.

9. Foster technical understanding

Requirements engineers understand technical context in order to formulate requirements that are feasible and testable.

10. Use tools deliberately

Tools such as backlogs, story maps, use cases or modelling support the conversation. They do not replace it.

Roles

MORe distinguishes roles in order to create clarity about responsibilities in the requirements process.

Product Owner

Prioritisation and stakeholder perspective

Responsible for prioritising requirements and assessing their business value. Contributes the perspective of users and stakeholders.

Requirements Engineer

Link between business and technology

Translates business needs into understandable, testable and technically feasible requirements. Mediates between business and technology.

Development Team

Implementation feedback and refinement

Provides feedback on feasibility and ensures a shared understanding of the requirements. Supports refinement and implementation.

Stakeholders

Contributing and validating requirements

Business departments, customers or users who contribute requirements, validate them and take acceptance decisions.

Agile Facilitator

Facilitation, optional

Supports the structuring and facilitation of MORe activities. The role can be taken on by the Scrum Master, for example.

Events

These events help to identify and refine requirements continuously and to implement them iteratively.

Needs Workshop

Identify new requirements

The goal is to identify new requirements, put them into context and prioritise them roughly. Stakeholders and product owner are in dialogue. It is held as needed, for example monthly or at the start of a project.

Refinement

Break down, discuss, enrich

The goal is to make requirements concrete, break them down, add acceptance criteria and discuss them technically. Refinement takes place iteratively and regularly, for example weekly.

Backlog Review

Review and re-prioritise

The goal is to review the existing requirements in the backlog. Requirements can be re-prioritised, added, sharpened or archived. The focus is on relevance and maturity.

Rolling Planning

Regular adjustment

The goal is to adjust the planning regularly to new insights, changed priorities and the actual state of progress.

Rolling planning looks not only at the next implementation step, but at the ongoing need for analysis, refinement, decision, implementation and evidence. It creates transparency about which requirements are to be worked on in the coming requirement cycles, sprints, releases or milestones.

The plan is not fixed once and for all, but continuously carried forward. New requirements are taken in, existing requirements are re-prioritised, and dependencies, risks or changes in capacity are taken into account.

Rolling planning connects strategic objectives with the operational work on requirements.

Rollout Planning

Prepare requirements for implementation

The goal is the coordinated preparation of requirements that go into implementation or delivery. Rollout planning is the connecting point between requirements management and development.

Requirement Cycle

The requirement cycle is a recurring cycle of two to four weeks for structured work on requirements.

The steps

- Identification
- Analysis
- Alignment
- Refinement
- Definition of Ready
- Feedback
- Handover
- Closing with a review and an update of the planning

The closing reflects on which requirements have been completed, which insights have been gained and how priorities change for the next cycle.

Goals of the requirement cycle

- Identify, analyse and prioritise requirements
- Ensure alignment with stakeholders
- Refine requirements, for example as user stories, use cases or models
- Reach implementation readiness through the Definition of Ready
- Obtain feedback, for example through prototyping, reviews or workshops
- Hand requirements over to development or release them for implementation

Every cycle ends with a review and planning. It reflects on what is fully prepared and what will be handled in the next cycle.

Benefits of the requirement cycle

- Encourages iterative, learning-oriented work on requirements
- Connects strategic thinking with operational implementation
- Creates rhythm, structure and transparency
- Allows a flexible response to new insights or changes

Requirement Cycle and Scrum

The Requirement Cycle takes place before and alongside development. It identifies, analyses, refines and clarifies requirements until implementable work can emerge from them.

Scrum controls development. MORe® maintains the domain context. The user story is not the specification. It refers to the leading specification, the relevant decisions, acceptance criteria, tests and evidence.

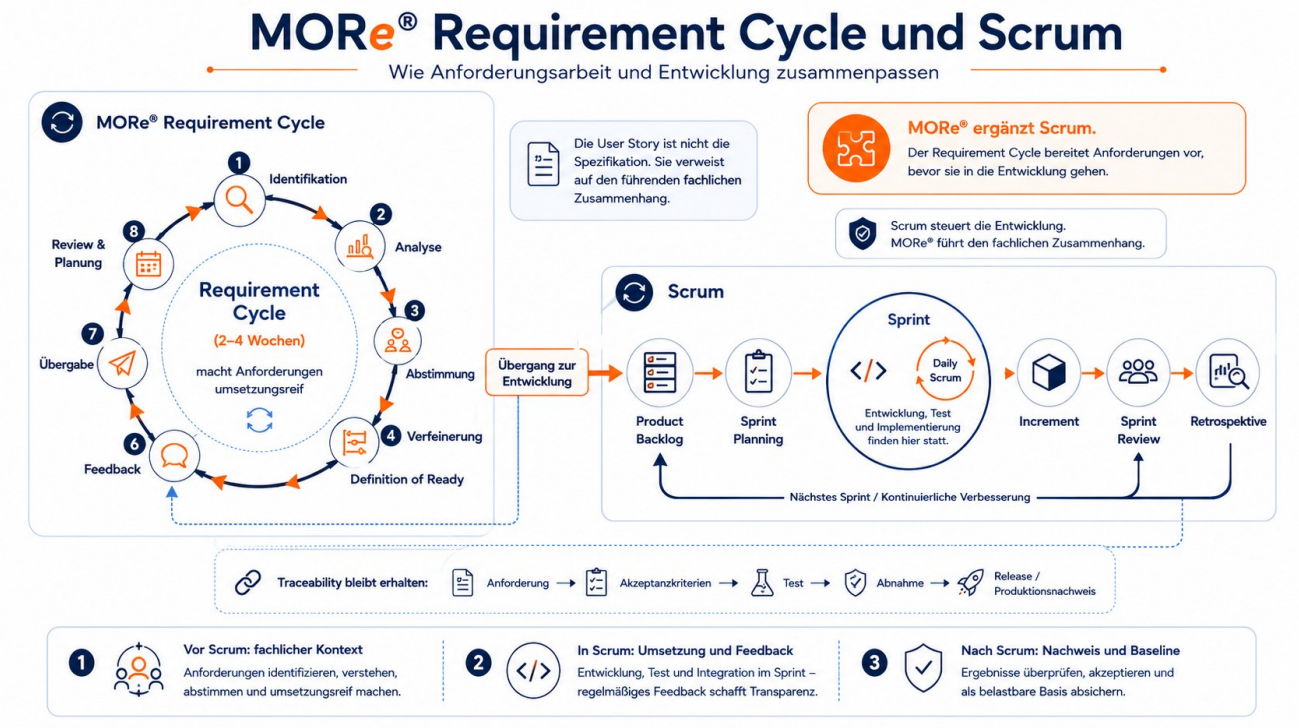

Figure 2: How the MORe® Requirement Cycle and Scrum work together.

Artefacts

MORe works with few but robust artefacts. What matters is not the amount of documentation, but the traceability of the connections.

In MORe, it is not only the artefact that is relevant, but its connection to the other artefacts.

Documents are not maintained. They are generated from maintained information.

A generated document is not the artefact. It is merely a view of the artefact – a form of publication at a given point in time, but not the object of work itself.

Requirements Backlog

The requirements backlog collects and prioritises all known requirements. It creates transparency about which requirements exist, what status they have and which will be worked on next.

Planning

Planning connects requirements with their implementation over time. It makes transparent which requirements are to be worked on in which requirement cycle, sprint, release or milestone, and it is continuously adjusted to new insights. In MORe, planning is not a one-off project event but an ongoing process.

Story Map

The story map arranges requirements along user journeys, business processes or domain workflows. It helps to recognise connections and to avoid looking at requirements in isolation.

Use Cases

Use cases describe how users, systems or roles interact with a function. They make domain workflows understandable and help to derive requirements from concrete situations of use.

Models and diagrams

Models and diagrams visualise processes, data flows, interfaces or domain structures. They reduce misunderstandings and support a shared understanding between the business department, requirements engineering and development.

Definition of Ready

MORe® deliberately uses a lean Definition of Ready as the transition point to development. It does not ensure that a user story becomes the specification. It ensures that a user story refers to the leading specification.

This prevents the common misunderstanding that a ticket or a user story replaces the domain context. The ticket controls implementation. The domain context remains maintained in the information architecture.

Acceptance criteria

Acceptance criteria make it verifiable when a requirement has been fulfilled from a business perspective. They connect the description of a requirement with implementation, test and acceptance.

Documentation

Documentation records business decisions, assumptions, clarifications, changes and evidence. It emerges while the requirement is being worked on, not afterwards as a reconstruction.

Traceability link

The traceability link is the connecting artefact in MORe. It links requirements with the associated decisions, models, acceptance criteria, tests, acceptances, changes and production evidence.

Traceability therefore does not arise from individual documents, but from the robust connection between them. The link makes visible how a business need became an implemented, tested, accepted change that can be evidenced in production.

Information graph and baseline

The page tree is navigation. The domain structure uses hierarchy for orientation, but traceability is created through semantic relationships. The information graph connects use cases and specifications with rules, decisions, acceptance criteria, tests, baselines and – in AI-supported systems – with data sources, models, prompt/guardrails and corpus/embedding.

The baseline holds together the approved state across business artefacts, technical implementation and production evidence. It makes visible which version applies from a business perspective, which decision approved it, how it was verified and on what basis it remains evidential in production.

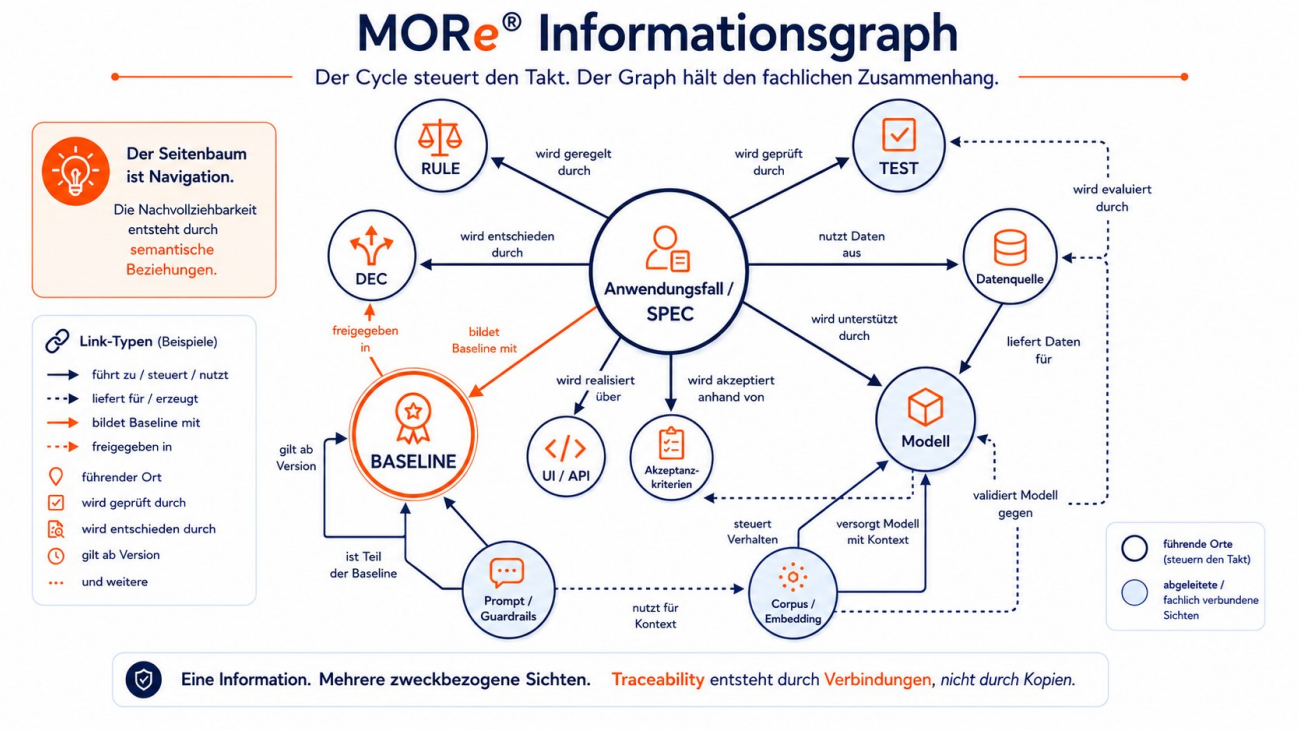

Figure 3: MORe® information graph with semantic relationships, baseline and AI-specific artefacts.

Further information

www.more-framework.org · mail@more-framework.org
www.projektkram.de · hallo@projektkram.de