

Informationsarchitektur für moderne Projekte

Der fehlende Baustein zwischen Scrum, Projektmanagement und Wissensmanagement – warum moderne Projekte ihr Wissen verlieren und wie das MORE Framework Informationen statt Dokumente organisiert.

Autor	Reiner Schindler
Organisation	MORE® - © more-framework.org
Datum	Juli 2026
Version	V1.3

MORE organisiert nicht Dokumente. Nicht Tickets. Nicht Dateien.

MORE liefert die Informationsarchitektur moderner Projekte

Das Projekt im 21. Jahrhundert

MORE organisiert nicht Dokumente. Nicht Tickets. Nicht Dateien.

MORE liefert die Informationsarchitektur moderner Projekte.

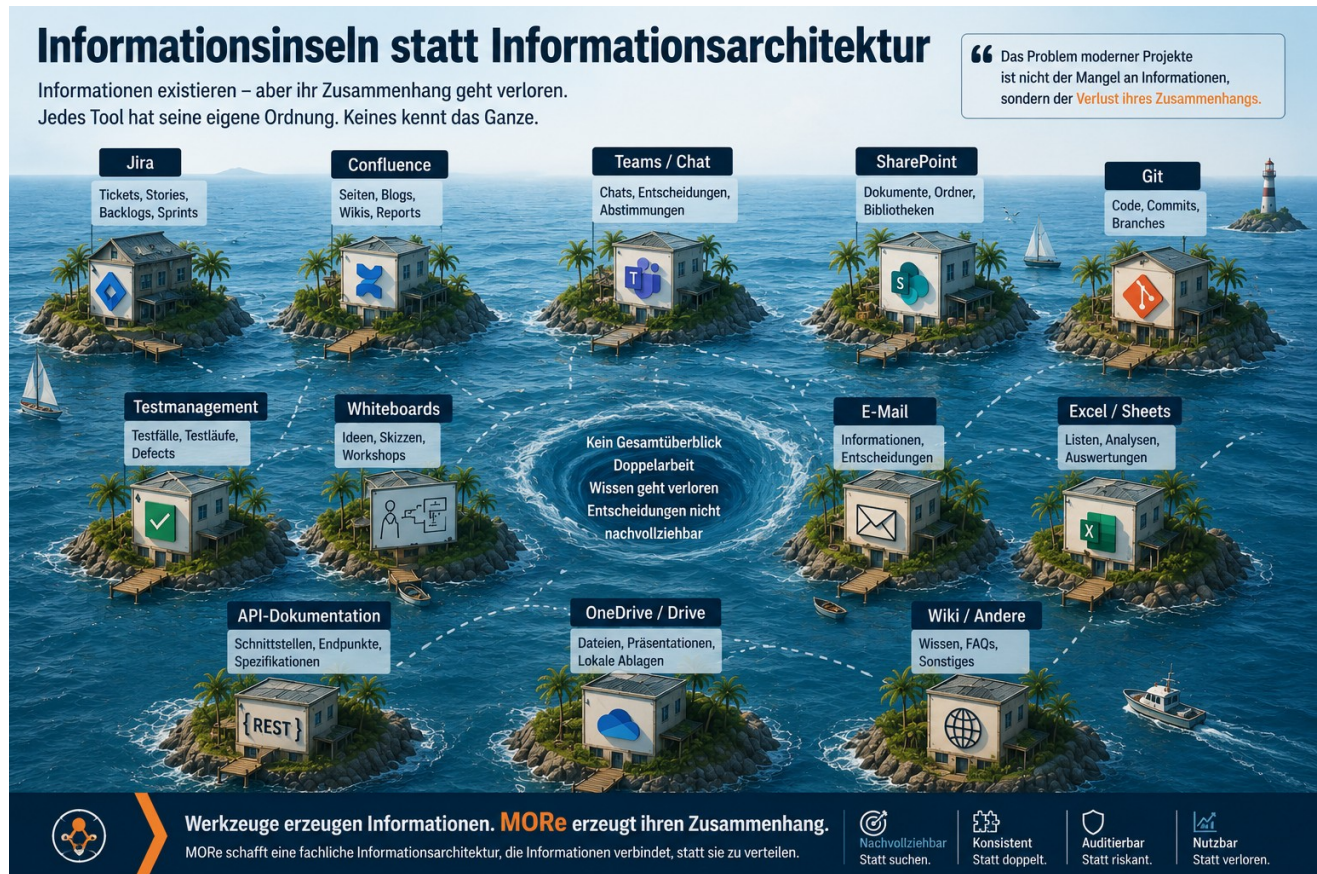


Abbildung 1 Moderne Projekte verfügen heute über eine Vielzahl spezialisierter Werkzeuge. Jedes Werkzeug organisiert Informationen nach seinen eigenen Regeln und Datenstrukturen. Der fachliche Zusammenhang zwischen den Informationen geht dabei häufig verloren. Es entstehen Informationsinseln.

Moderne Projekte scheitern nicht an fehlenden Informationen. Sie scheitern am Verlust ihres Zusammenhangs.

1 Management Summary

Moderne Projekte verfügen über mehr Informationen als jemals zuvor – und verlieren gleichzeitig den Überblick darüber. Anforderungen liegen in Jira, Fachkonzepte in Confluence, Entscheidungen in Chats und E-Mails, Dateien in Teams oder SharePoint, Testfälle in Testwerkzeugen und Quellcode in Git. Jede einzelne Information ist vorhanden. Ihr fachlicher Zusammenhang geht jedoch verloren.

Dieses Whitepaper vertritt eine These, die auf den ersten Blick unbequem ist:

Das Kernproblem moderner Projekte ist weder die Methode noch das Werkzeug. Es ist die Informationsfragmentierung. Werkzeuge strukturieren Dokumente, Dateien und Tickets – aber nicht das Projektwissen entlang seiner fachlichen Zusammenhänge. Dadurch entstehen Informationsinseln, zwischen denen der rote Faden reißt.

Die Antwort darauf ist keine weitere Methode und kein weiteres Tool, sondern eine Informationsarchitektur: eine fachliche Struktur, die festlegt, welche Information an welchem Ort entsteht, welche Frage sie beantwortet und wie sie mit den übrigen Informationen des Projekts verbunden ist.

Das MORE Framework liefert diese Informationsarchitektur. MORE strukturiert Projektinformationen nicht nach Dokumentarten oder Werkzeugen, sondern nach fachlichen Domänen und Anwendungsfällen. Jede Ebene der entstehenden Informationshierarchie beantwortet genau eine fachliche Frage.

Die Dokumentation wird dabei nicht nachträglich erstellt – sie entsteht während der Arbeit an der Anforderung, weil jede Information dort dokumentiert wird, wo sie fachlich entsteht.

Die zentralen Aussagen dieses Whitepapers im Überblick:

- **Informationsarchitektur statt Dokumentenarchitektur:** Nicht Dokumente werden organisiert, sondern Informationen und ihre fachlichen Zusammenhänge.
- **Dokumentation entsteht während der Bearbeitung:** Nicht als nachgelagerte Pflichtaufgabe, sondern als Bestandteil der Anforderungsarbeit.
- **Fachliche Hierarchie statt Dokumenthierarchie:** Die Struktur folgt Fachgebieten und Anwendungsfällen, nicht Ordern und Dateiformaten.
- **Projektwissen statt Dokumentenablage:** Am Ende des Projekts existiert eine vollständige, gewachsene Systemdokumentation – nicht ein Laufwerk voller Fragmente.
- **Jede Ebene beantwortet genau eine Frage:** Vom Anwendungsfall über Prozess, Fachmodell, Oberfläche und Schnittstelle bis zu Testfällen und Abnahme.

| MORE organisiert keine Dokumente. MORE organisiert Projektwissen.



2 Einleitung: Mehr Werkzeuge, weniger Überblick

Die Digitalisierung hat die Projektarbeit grundlegend verändert. Cloud-Plattformen machen Informationen überall verfügbar, agile Vorgehensmodelle haben starre Phasenpläne abgelöst, und für nahezu jede Aufgabe existiert heute ein spezialisiertes Werkzeug: Backlog-Management, Wissensdatenbanken, Kollaborationsplattformen, Testmanagement, Versionsverwaltung, digitale Whiteboards.

Jedes dieser Werkzeuge löst ein reales Problem – und jedes löst es gut.

Jira macht Arbeit sichtbar und steuerbar.

Confluence macht Texte gemeinsam bearbeitbar.

Git macht Quellcode nachvollziehbar.

Teams und Chat machen Abstimmung schnell.

Genau darin liegt das Paradox moderner Projekte:

Je besser die einzelnen Werkzeuge werden, desto stärker verteilt sich das Wissen des Projekts auf sie.

Das Ergebnis kennt jeder, der in einem größeren Projekt gearbeitet hat. Auf die einfache Frage „Warum verhält sich das System an dieser Stelle so?“ gibt es keine einfache Antwort mehr. Die Anforderung steht in einem Ticket, das seit zwei Jahren geschlossen ist. Die fachliche Diskussion fand in einem Chat statt. Die Entscheidung wurde in einem Meeting getroffen, dessen Protokoll auf einem Laufwerk liegt. Die Schnittstellenbeschreibung existiert in drei Versionen, von denen keine als verbindlich markiert ist.

Bemerkenswert ist: Keine dieser Informationen fehlt. Sie alle existieren.

Was fehlt, ist ihr Zusammenhang. Und dieser Zusammenhang ist keine technische Eigenschaft, die ein Werkzeug automatisch herstellen könnte – er ist eine fachliche Struktur, die bewusst gestaltet werden muss.

Dieses Whitepaper beschreibt, warum genau an dieser Stelle die Dokumentations- und Wissensprobleme moderner Projekte entstehen, warum Dokumente und Projektlaufwerke keine Lösung sind, und wie eine fachliche Informationsarchitektur – konkret: das MORE Framework – den Unterschied macht.



3 Das eigentliche Problem: Informationsfragmentierung

Wenn in Projekten über fehlende Dokumentation geklagt wird, richtet sich die Kritik meist an die falschen Adressaten. Es liegt nicht an den Dokumenten – es wurden nie so viele geschrieben wie heute. Es liegt nicht an Scrum – agile Teams liefern zuverlässiger als die meisten klassischen Projekte. Und es liegt nicht an Jira – Tickets bilden Arbeit präziser ab als jeder Statusbericht.

Das eigentliche Problem ist ein anderes: **Informationsfragmentierung**. Die größte Herausforderung moderner Projekte ist nicht der Mangel an Informationen, sondern deren Verteilung auf Systeme, die nichts voneinander wissen.

Jedes Werkzeug bringt seine eigene Ordnung mit. Jira ordnet nach Projekten, Epics und Sprints. Confluence ordnet nach Bereichen und Seitenbäumen. SharePoint ordnet nach Bibliotheken und Ordnern. Git ordnet nach Repositories und Branches. Jede dieser Ordnungen ist in sich schlüssig – aber keine davon ist die fachliche Ordnung des Projekts.

Keine beantwortet die Frage: Welche Informationen gehören fachlich zu dieser Anforderung?

So entstehen Informationsinseln: in sich konsistente Wissensbestände, die untereinander nur durch das Gedächtnis der Beteiligten verbunden sind. Solange das Team stabil ist und das Projekt läuft, funktioniert diese Verbindung erstaunlich gut. Sie zerfällt in dem Moment, in dem Wissensträger das Projekt verlassen, das Projekt in den Betrieb übergeht oder – im regulierten Umfeld – ein Prüfer nachvollziehen will, wie aus einem fachlichen Bedarf ein produktives System wurde.

Fragmentierung ist deshalb kein Ordnungsproblem, sondern ein Risikofaktor: für die Steuerung, für Test und Abnahme, für Revision und Audit und für den späteren Betrieb. Wer den Zusammenhang seiner Informationen verliert, verliert die Fähigkeit, sein eigenes System zu erklären.

Kernaussage

Nicht der Mangel an Informationen ist das Problem moderner Projekte, sondern ihre Fragmentierung. Werkzeuge strukturieren Dokumente, Dateien und Tickets – sie strukturieren nicht das Projektwissen entlang fachlicher Zusammenhänge.

4 Die heutige Projektlandschaft

Ein Blick auf die Werkzeuglandschaft eines typischen IT-Projekts macht das Ausmaß der Verteilung sichtbar. Die folgende Übersicht ist keine Übertreibung, sondern der Normalfall in mittleren und großen Organisationen:

Werkzeug	Was es strukturiert	Was es nicht strukturiert
Jira / Azure DevOps	Arbeitspakete, Tickets, Sprints, Status	Den fachlichen Zusammenhang zwischen Anforderungen
Confluence / Wiki	Seiten, Bereiche, Seitenbäume	Die Verbindung von Konzept, Umsetzung und Test
Teams / Slack / E-Mail	Kommunikation, Abstimmung	Entscheidungen und ihre Begründungen
SharePoint / OneDrive	Dateien, Ordner, Versionen	Welche Datei wozu gehört und was verbindlich ist
Git	Quellcode, Änderungen, Branches	Warum eine Änderung fachlich erfolgte
Testwerkzeuge	Testfälle, Testläufe, Ergebnisse	Welche Anforderung ein Test tatsächlich abdeckt
Whiteboards (Miro u. a.)	Skizzen, Workshops, Ideen	Den Übergang von der Idee zur Anforderung
Excel	Listen, Auswertungen, Planungen	Fast alles – und wird trotzdem überall eingesetzt

Jede Zeile dieser Tabelle beschreibt ein Werkzeug, das seinen Zweck erfüllt. Die rechte Spalte beschreibt, was zwischen den Werkzeugen verloren geht: der Zusammenhang. Eine Anforderung durchläuft im Laufe ihres Lebens fast alle diese Systeme – als Idee auf dem Whiteboard, als Ticket in Jira, als Konzept in Confluence, als Diskussion im Chat, als Code in Git, als Testfall im Testwerkzeug, als Abnahmeprotokoll auf dem Laufwerk. In jedem System hinterlässt sie eine Spur.

Kein System kennt die ganze Geschichte.

Der übliche Reflex auf dieses Problem lautet: ein weiteres Werkzeug einführen, das alles integriert. Die Erfahrung zeigt, dass genau das die Fragmentierung verschärft – die Projektlandschaft wächst um eine weitere Insel. Ein Werkzeug kann Verknüpfungen speichern, aber es ersetzt keine fachliche Struktur. Ein Tool entscheidet nicht, ob eine Anforderung vollständig verstanden wurde, und es erkennt nicht von selbst, ob die Verbindung zwischen Anforderung, Test und Abnahme inhaltlich sinnvoll ist.

Die Struktur muss vor dem Werkzeug kommen.



5 Warum Dokumentation fast immer fehlt

Am Ende vieler Projekte steht dieselbe Erkenntnis: Die Dokumentation fehlt, ist veraltet oder unvollständig. Das ist so verbreitet, dass es fast als Naturgesetz akzeptiert wird. Dabei hat es klar benennbare Ursachen – und keine davon heißt „die Beteiligten waren zu faul“.

5.1 Dokumentation als nachgelagerte Aufgabe

In vielen Vorgehensmodellen wird Dokumentation als eigenständige, nachgelagerte Aufgabe behandelt: Erst wird entwickelt und getestet, dann – „wenn Zeit ist“ – wird dokumentiert. Unter Termin- und Kostendruck ist die Reihenfolge damit bereits entschieden. Funktionierende Software ist sichtbar und wird abgenommen; Dokumentation ist unsichtbar und wird verschoben. Was verschoben wird, wird gekürzt. Was gekürzt wird, entfällt.

5.2 Nachträgliche Dokumentation ist Rekonstruktion

Wenn Dokumentation erst am Ende entsteht, ist sie keine Dokumentation mehr, sondern eine Rekonstruktion. Kurz vor Abnahme, Audit oder Go-Live wird versucht, nachträglich zusammenzuführen, was im Projektverlauf nicht verbunden wurde: Tickets werden durchsucht, Entscheidungen aus E-Mails rekonstruiert, Testnachweise zusammengestellt, Abnahmen nachgetragen. Das Ergebnis ist zwangsläufig lückenhaft – nicht, weil die Beteiligten nachlässig wären, sondern weil das Wissen über das „Warum“ zu diesem Zeitpunkt bereits verloren ist. Was erst am Ende dokumentiert wird, ist meistens zu spät, zu lückenhaft oder zu schwach.

5.3 Der fehlende Ort

Die dritte Ursache ist die unscheinbarste und zugleich wichtigste: Selbst wer während der Arbeit dokumentieren will, findet oft keinen Ort dafür. Wohin gehört eine fachliche Entscheidung? Ins Ticket, das nach dem Sprint geschlossen wird? Ins Wiki, in dem sie niemand sucht? In das Protokoll, das auf dem Laufwerk liegt? Wenn es keinen definierten Ort für eine Information gibt, wird sie entweder gar nicht oder an mehreren Orten gleichzeitig festgehalten – und beides führt zum selben Ergebnis: Sie ist später nicht auffindbar oder nicht verbindlich.



5.4 Der Perspektivwechsel von MORE

MORE verfolgt einen anderen Ansatz und setzt an allen drei Ursachen gleichzeitig an. Dokumentation wird nicht als nachgelagerte Aufgabe behandelt, sondern als Bestandteil der Arbeit an der Anforderung. Jede Information wird dann dokumentiert, wenn daran gearbeitet wird – nicht Wochen später, nicht nachträglich aus Tickets rekonstruiert. Und für jede Information gibt es genau einen fachlich definierten Ort. Damit verschwindet die Dokumentation als eigener Arbeitsschritt: Sie ist das Nebenprodukt sauberer fachlicher Arbeit – oder genauer: Sie ist die fachliche Arbeit, sichtbar gemacht.

Das MORE-Prinzip

Dokumentation ist Teil der Arbeit – nicht ihr nachträgliches Nebenprodukt. Die Dokumentation entsteht während der Bearbeitung der Anforderung, weil jede Information dort dokumentiert wird, wo sie fachlich entsteht.

6 Dokumente sind keine Informationsarchitektur

Der klassische Lösungsversuch für das Dokumentationsproblem ist das strukturierte Projektlaufwerk: eine sorgfältig angelegte Ordnerstruktur mit Namenskonventionen, Vorlagen und Ablageregeln. Dieser Ansatz scheitert regelmäßig – und zwar nicht an mangelnder Disziplin, sondern an einem Konstruktionsfehler.

Ein Dokument ist ein Container. Es bündelt Informationen, die zu einem bestimmten Zeitpunkt für einen bestimmten Zweck zusammengestellt wurden: ein Fachkonzept, ein Testhandbuch, eine Schnittstellenspezifikation. Das Problem: Dieselbe Information wird für verschiedene Zwecke gebraucht und landet deshalb in mehreren Containern. Die Beschreibung eines Geschäftsprozesses steht im Fachkonzept, in der Schulungsunterlage, im Testkonzept und in der Betriebsdokumentation – viermal, in vier Versionen, mit vier Pflegezyklen. Spätestens nach der zweiten Änderung widersprechen sich die Container.

Ordnerstrukturen verschärfen das Problem, statt es zu lösen. Ein Ordner kann eine Datei nur an genau einer Stelle einsortieren, aber jede fachliche Information hat mehrere legitime Zugehörigkeiten: Die Schnittstellenbeschreibung gehört zum Anwendungsfall, zur technischen Architektur und zum Testumfang gleichzeitig. Welcher Ordner ist der richtige? Jede Antwort ist falsch – und deshalb entstehen Kopien, Verweise und schließlich das bekannte Suchen an drei Orten.

Hinzu kommt die Zeitdimension: Dokumente sind Momentaufnahmen. Ein Fachkonzept beschreibt den Stand zur Freigabe – drei Monate später ist das System weiter, das Dokument nicht. Ein Projektlaufwerk ist deshalb keine Wissensbasis, sondern ein Archiv von Momentaufnahmen unterschiedlichen Alters, aus dem niemand mehr ablesen kann, was aktuell gilt.

Die Konsequenz ist grundsätzlich: Das Problem lässt sich nicht durch bessere Dokumente lösen, sondern nur durch einen Wechsel der Ordnungseinheit.

Nicht das Dokument ist die sinnvolle Einheit der Projektorganisation, sondern die Information.

Eine Informationsarchitektur legt fest, welche Informationen es gibt, welche Frage jede Information beantwortet, wo sie gepflegt wird – genau einmal – und wie sie mit anderen Informationen zusammenhängt. Dokumente können daraus jederzeit erzeugt werden: als Sichten auf den aktuellen Informationsstand, zusammengestellt für einen Zweck. Der umgekehrte Weg – aus Dokumenten eine Informationsarchitektur zu machen – funktioniert nicht.

Merksatz

So viel Dokumentation wie nötig – aber jede Information genau einmal und am richtigen Ort. Dokumente sind Sichten auf Informationen, nicht ihr Speicherort.

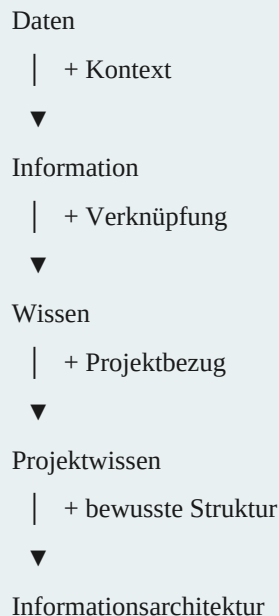
7 Vom Datum zum Projektwissen: der Informationsbegriff

Bevor der Kern von MORE beschrieben wird, lohnt ein kurzer Blick auf die Begriffe. Denn „Informationsarchitektur“ ist mehr als ein Schlagwort – der Begriff beschreibt präzise, worum es geht. Daten sind Rohmaterial: ein Feldwert, ein Zeitstempel, ein Ticketstatus. Zu Informationen werden Daten erst durch Kontext – wenn erkennbar ist, worauf sie sich beziehen und was sie bedeuten. „Status: Done“ ist ein Datum. „Die Anforderung UC-042 wurde am 12. Mai fachlich abgenommen“ ist eine Information.

Wissen entsteht, wenn Informationen verknüpft werden und Zusammenhänge erkennbar machen: Warum wurde diese Anforderung gestellt? Welche Entscheidung liegt ihr zugrunde? Wie hängt sie mit dem Geschäftsprozess zusammen, den sie verändert? Wissen ist damit immer relational – es lebt nicht in einzelnen Informationen, sondern in ihren Verbindungen.

Projektwissen schließlich ist das auf ein Vorhaben bezogene Wissen: die Gesamtheit der fachlichen Zusammenhänge, Entscheidungen, Strukturen und Nachweise, die ein Projekt ausmachen. Genau dieses Projektwissen ist es, das in fragmentierten Projektlandschaften verloren geht – nicht die einzelnen Informationen, sondern ihre Verbindungen.

Eine Informationsarchitektur ist die bewusst gestaltete Struktur dieses Projektwissens: Sie definiert die Informationsarten, ihre Orte, ihre Verantwortlichkeiten und – entscheidend – ihre Beziehungen untereinander. Sie ist damit das Gegenstück zur gewachsenen Werkzeuglandschaft: nicht technisch, sondern fachlich; nicht nach Systemen organisiert, sondern nach Zusammenhängen.



8 Der Kern von MORE: die Informationshierarchie

MORE – More Organized Requirements engineering – ist als Framework für reguliertes Anforderungsmanagement entstanden. Es verbindet fachliche Anforderungen, Entscheidungen, Umsetzung, Test, Abnahme und Produktionsnachweis in einem durchgängigen Arbeitsmodell. In der praktischen Anwendung hat sich gezeigt: Der Kern von MORE ist mehr als Anforderungsmanagement. Er ist eine Informationsarchitektur für das gesamte Projekt.

MORE strukturiert Projektinformationen nicht nach Dokumentarten oder Werkzeugen, sondern nach fachlichen Domänen und Anwendungsfällen. Die oberste Ordnungsebene ist das Fachgebiet – eine fachliche Domäne wie Vertrieb, Vertragsverwaltung oder Meldewesen. Innerhalb eines Fachgebiets ist der Anwendungsfall (Use Case) die zentrale Ordnungseinheit: eine in sich geschlossene fachliche Aufgabe, die ein Nutzer oder ein System erledigen will.

An jedem Anwendungsfall hängen die Informationen, die ihn vollständig beschreiben – jede genau einmal, jede an ihrem definierten Ort:



Drei Eigenschaften dieser Hierarchie sind entscheidend.

Erstens: Sie ist fachlich, nicht technisch. Die Struktur folgt der Frage „Wozu gehört diese Information fachlich?“

– nicht der Frage „In welchem Werkzeug liegt sie?“. Ob der Prozess in Confluence beschrieben, das Fachmodell

in einem Modellierungswerkzeug gepflegt und die Testfälle in einem Testtool verwaltet werden, ist zweitrangig. Die Hierarchie ist die logische Klammer über den Werkzeugen.

Zweitens: Jedes Artefakt hat einen klaren Zweck. Kein Artefakt existiert aus methodischen Gründen oder weil eine Vorlage es vorsieht. Was für einen Anwendungsfall nicht relevant ist, entfällt ersatzlos – eine reine Datenmigration braucht keine Oberfläche, ein interner Batchlauf keine Abnahme durch Endanwender. So viel Dokumentation wie nötig, nicht so viel wie möglich.

Drittens: Entscheidungen hängen am Anwendungsfall. Fachliche Entscheidungen werden nicht in Protokollen begraben, sondern dort festgehalten, wo sie wirken: am Anwendungsfall, den sie betreffen. Wer später fragt „Warum ist das so gelöst?“, findet die Antwort am Ort der Lösung – nicht in einem Meetingprotokoll vom vorletzten Jahr.

Oberhalb der Anwendungsfälle bildet eine zusammenfassende Seite je Fachgebiet die Klammer: Sie gibt den Überblick über die Domäne, ihre Anwendungsfälle und ihren Zusammenhang. So strukturiert ist das Fachgebiet nichts anderes als die Prozessbeschreibung des Projekts – nicht als eigenständiges Dokument, sondern als gewachsene Struktur.

8.1 Lesbarkeit im Großen

Die kompakten Kürzel und kurzen Artefaktnamen sind kein kosmetisches Detail. Wenn ein Projekt 300 Artefakte umfasst, entscheidet die Benennung darüber, ob die Struktur lesbar bleibt. „UC-042 Meeting aus Kontakt anlegen“ erfasst man beim Überfliegen; „GPB-037 Geschäftsprozessbeschreibung“ nicht. Die Namenskonvention aus Kürzel, Nummer und knappem Klartext hält die Hierarchie auch in großen Projekten navigierbar – vom kleinen Zoho-Erweiterungsprojekt bis zum Großvorhaben mit Dutzenden Fachgebieten.

9 Jede Ebene beantwortet genau eine Frage

Das tragende Prinzip der MORE-Informationshierarchie lässt sich in einem Satz zusammenfassen: Jede Ebene beantwortet genau eine fachliche Frage – und ergänzt den vorherigen Informationsstand, ohne ihn zu wiederholen.

Dieses Prinzip klingt unscheinbar, hat aber weitreichende Konsequenzen. Es beendet die Redundanz, weil jede Information genau einen Ort hat. Es beendet die Vollständigkeitsdebatten, weil für jede Ebene klar ist, wann sie fertig ist: wenn ihre Frage beantwortet ist. Und es macht Lücken sichtbar, weil eine unbeantwortete Frage sofort auffällt.

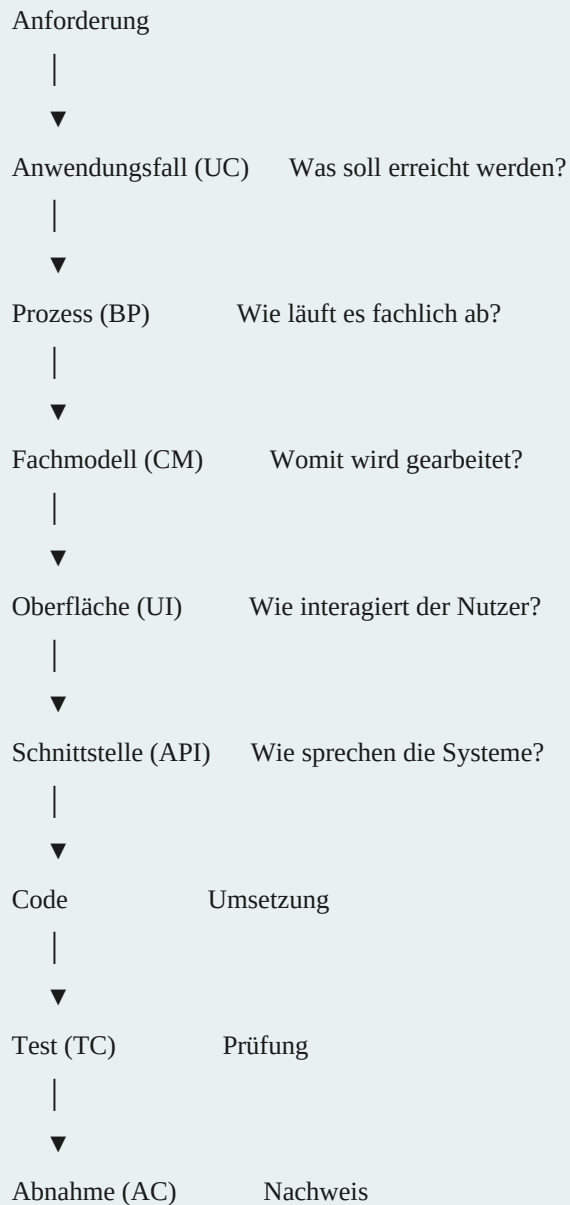
Kürzel	Ebene	Die eine Frage	Typische Inhalte
UC	Anwendungsfall	Was will der Nutzer erreichen?	Ziel, Akteure, Auslöser, Ergebnis, Abgrenzung
BP	Prozess	Wie läuft es fachlich ab?	Ablauf, Varianten, Sonderfälle, fachliche Regeln
CM	Fachmodell	Mit welchen Begriffen und Daten wird gearbeitet?	Fachklassen, Attribute, Beziehungen, Zustände
UI	Oberfläche	Wie interagiert der Nutzer mit dem System?	Masken, Felder, Validierungen, Bedienlogik
API	Schnittstelle	Wie kommunizieren die Systeme?	Endpunkte, Datenformate, Fehlerverhalten
TC	Testfälle	Wie wird die Umsetzung geprüft?	Testszenarien, Testdaten, erwartete Ergebnisse
AC	Abnahme	Wann ist die Anforderung fachlich erfüllt?	Akzeptanzkriterien, Abnahmeentscheidung, Nachweis
DEC	Entscheidungen	Warum ist es so gelöst?	Festlegung, Alternativen, Begründung, Datum, Entscheider

Die Reihenfolge der Ebenen ist dabei kein Formalismus, sondern bildet den natürlichen Erkenntnisweg einer Anforderung ab: Erst wird klar, was erreicht werden soll (UC). Dann, wie der fachliche Ablauf aussieht (BP) und mit welchen Begriffen gearbeitet wird (CM). Daraus ergeben sich Oberfläche (UI) und Schnittstellen (API). Und erst wer weiß, was gebaut wird, kann festlegen, wie es geprüft (TC) und wann es akzeptiert wird (AC).



9.1 Der Weg einer Anforderung

Damit zeigt sich der eigentliche Charakter von MORE: Es ist kein Dokumentationsframework. Die Hierarchie beschreibt den Weg einer Anforderung – von der fachlichen Idee bis zur abgenommenen, produktiven Lösung. Jede Ebene ist eine Station dieses Weges, und die Dokumentation ist schlicht die Spur, die die Anforderung auf ihrem Weg hinterlässt:





Und genau deshalb funktioniert Traceability in MORE ganz natürlich. Nachvollziehbarkeit muss nicht nachträglich als Reporting-Übung hergestellt werden – sie entsteht als Ergebnis sauberer Arbeit an der Anforderung, weil jede Station des Weges ihre Information am definierten Ort hinterlässt. Der Zusammenhang zwischen Bedarf, Entscheidung, Umsetzung, Test, Abnahme und Produktion ist keine zusätzliche Verwaltungsleistung, sondern die Struktur selbst.

More traceability. Less reconstruction. – Mehr Nachweisfähigkeit, weniger nachträgliche Rekonstruktion.

10 Beispiel: Der Meeting-Button in Zoho CRM

Wie sich die Informationshierarchie in der Praxis anfühlt, zeigt ein bewusst kleines Beispiel: eine Erweiterung in Zoho CRM. Der Vertrieb wünscht sich einen Button in der Kontaktansicht, mit dem direkt aus einem Kontakt ein Meeting angelegt werden kann – vorbelegt mit den Daten des Kontakts. Eine Anforderung, wie sie in jedem Projekt vorkommt: klein genug für ein einzelnes Ticket, groß genug, um alle typischen Informationsarten zu erzeugen.

In der klassischen Welt würde diese Anforderung als Ticket erfasst, im Chat diskutiert, umgesetzt, kurz getestet und geschlossen. Zwei Jahre später weiß niemand mehr, warum der Button bestimmte Felder vorbelegt, warum ein Meeting nicht angelegt wird, wenn der Kontakt kein E-Mail-Feld hat, und ob das Verhalten so abgenommen wurde. In MORE sieht dieselbe Anforderung so aus:

Artefakt	Inhalt (Auszug)
UC-042 Meeting aus Kontakt anlegen	Ziel: Vertriebsmitarbeiter legt aus der Kontaktansicht mit einem Klick ein Meeting an. Akteur: Vertrieb. Auslöser: Terminvereinbarung im Kundengespräch. Ergebnis: Meeting im Kalender, verknüpft mit dem Kontakt.
BP-042 Ablauf	Kontakt öffnen → Button „Meeting anlegen“ → Formular mit vorbelegten Daten → Datum/Uhrzeit wählen → Speichern → Meeting erscheint in Kontakt-Historie und Kalender. Variante: Kontakt ohne E-Mail-Adresse → Hinweis, keine Einladung.
CM-042 Fachmodell	Fachklassen Kontakt und Meeting, Beziehung 1:n, übernommene Attribute (Name, Firma, E-Mail), Pflichtfelder des Meetings.
UI-042 Oberfläche	Position des Buttons in der Kontaktansicht, Formularfelder mit Vorbelegung, Validierungen, Verhalten bei Abbruch.
API-042 Schnittstelle	Aufruf der Zoho-Meeting-API: Endpunkt, übergebene Felder, Fehlerverhalten bei nicht erreichbarem Kalenderdienst.
TC-042 Testfälle	Standardfall, Kontakt ohne E-Mail, Abbruch im Formular, Kalenderdienst nicht erreichbar, Berechtigungen.
AC-042 Abnahme	Akzeptanzkriterien je Testfall, fachliche Abnahme durch Vertriebsleitung am 12.05., Nachweis verlinkt.
DEC-042 Entscheidungen	Festlegung: keine Einladung ohne E-Mail-Adresse statt Pflichtfeld-Zwang – Begründung, Alternativen, Entscheider, Datum.

Der Aufwand für diese Struktur ist minimal – jede dieser Informationen musste ohnehin geklärt werden, um die Anforderung umzusetzen. MORE hat nichts hinzugefügt außer dem Ort, an dem die Klärung festgehalten wird. Der Unterschied zeigt sich später: Die Frage „Warum bekommt der Kunde



keine Einladung?“ beantwortet DEC-042 in dreißig Sekunden. Die Frage „Ist das Verhalten so abgenommen?“ beantwortet AC-042. Und wenn der Anwendungsfall erweitert wird, ist der vollständige Informationsstand der Ausgangspunkt – nicht eine Archäologie-Übung in geschlossenen Tickets.

Dasselbe Muster trägt vom kleinen Zoho-Projekt bis zum Großvorhaben. Es skaliert, weil es nicht auf Vollständigkeit von Dokumenten setzt, sondern auf die Beantwortung von Fragen – und die Fragen sind in jedem Projekt dieselben.

11 MORE und Scrum

Die häufigste Rückfrage zu MORE lautet: „Wir arbeiten agil – passt das zusammen?“ Die Antwort ist ein klares Ja, und der Grund liegt in einer sauberen Arbeitsteilung: Scrum und MORE beantworten unterschiedliche Fragen.

11.1 Was Scrum beschreibt

Scrum beschreibt die Zusammenarbeit von Teams: Rollen, Ereignisse, Artefakte der Teamorganisation. Es beantwortet die Fragen, wie Arbeit priorisiert wird, in welchem Rhythmus geliefert wird, wie das Team Transparenz herstellt und sich verbessert. Darin ist Scrum stark – und daran ändert MORE nichts.

11.2 Was Scrum bewusst nicht beschreibt

Scrum sagt bewusst nichts darüber, wie Anforderungen fachlich strukturiert, dokumentiert und nachvollziehbar geführt werden. Der Scrum Guide definiert kein Anforderungsformat, keine Dokumentationsstruktur und keine Nachweisführung – das ist keine Lücke im Sinne eines Fehlers, sondern eine bewusste Auslassung: Scrum ist ein Rahmenwerk für Teamarbeit, kein Informationsmodell. Ein Ticket kann erledigt, eine User Story abgeschlossen, ein Inkrement geliefert sein – und trotzdem bleibt offen, ob der fachliche Ursprung, die Entscheidung, der Test und die Abnahme belastbar miteinander verbunden sind.

11.3 Warum beide zusammenpassen

Genau hier greifen die beiden ineinander: Scrum organisiert die Arbeit, MORE organisiert das Wissen, das dabei entsteht. Das Product Backlog bleibt das Steuerungsinstrument des Product Owners; die MORE-Hierarchie ist der Ort, an dem der fachliche Gehalt der Backlog-Einträge lebt. Eine User Story verweist auf ihren Anwendungsfall; Refinement füllt Prozess, Fachmodell und Akzeptanzkriterien; die Definition of Ready ist erreicht, wenn die Fragen der relevanten Ebenen beantwortet sind. Sprint für Sprint wächst so nicht nur das Produkt, sondern auch sein dokumentiertes Wissen – im selben Takt, ohne eigenen Dokumentationssprint am Ende.

MORE ersetzt damit weder Scrum noch Kanban, LeSS oder SAFe. Es ergänzt sie um die Perspektive, die sie bewusst offenlassen: die Struktur des Projektwissens und – im regulierten Umfeld – die Nachweisfähigkeit. Agile Vorgehensmodelle beschreiben die Zusammenarbeit von Teams. MORE beschreibt die Struktur des Projektwissens.

12 MORE und Projektmanagement

Die Informationshierarchie von MORE ist am Beispiel fachlicher Anforderungen entstanden. Ihr Grundprinzip – fachliche Domänen, darunter Anwendungsfälle, daran die Informationen mit je einer klaren Frage – ist jedoch nicht auf Anforderungen beschränkt. Es trägt für das gesamte Projektmanagement.

Ein Projekt lässt sich als eigenes Fachgebiet mit vier (Fach-)Domänen strukturieren – angelehnt an die klassischen Disziplinen, wie sie auch das V-Modell kennt:

Domäne	Typische Anwendungsfälle	Beispiele für Informationen
Projektmanagement (PM)	Projekt planen, steuern, berichten, Risiken managen	Projektauftrag, Statusberichte, Entscheidungen Planung, Risikoliste,
Qualitätsmanagement (QM)	Qualität planen, prüfen, nachweisen	QS-Plan, Prüfkriterien, Prüfprotokolle Reviews,
Systemerstellung (SE)	Anforderungen umsetzen (Kern der MORE-Hierarchie)	Fachgebiete, Anwendungsfälle, BP/CM/UI/API/TC/AC
Konfigurationsmanagement (KM)	Stände verwalten, Änderungen steuern, ausliefern	Versionen, Änderungsanträge, Release-Nachweise

So strukturiert, ist das Projektmanagement nichts anderes als die Prozessbeschreibung des Projekts selbst: Auch das Projekt hat Anwendungsfälle („Statusbericht erstellen“, „Risiko bewerten“, „Release ausliefern“), auch diese haben Abläufe, Verantwortliche, Entscheidungen und Nachweise. Dieselbe Hierarchie, dieselben Prinzipien – jede Information genau einmal, am richtigen Ort, mit klarer Frage. Für Projektleitungen hat das einen praktischen Effekt: Steuerungsinformationen und Fachinformationen leben in derselben Architektur. Der Statusbericht verweist auf die Anwendungsfälle, um die es geht; die Risikoliste verweist auf die betroffenen Fachgebiete; die Entscheidung im Lenkungsausschuss hängt an dem Anwendungsfall, den sie betrifft. Das Projekt erklärt sich aus seiner eigenen Struktur – und zwar auch noch nach seinem Ende, wenn Revision oder Betrieb Fragen stellen.

13 MORE und Wissensmanagement

Auf den ersten Blick liegt der Einwand nahe: „Das haben wir doch – wir haben ein Wiki.“ Tatsächlich unterscheidet sich MORE von einem Wiki in einem grundsätzlichen Punkt: Ein Wiki ist ein Werkzeug, MORE ist eine Architektur.

Ein Wiki stellt Seiten bereit, die beliebig angelegt, verlinkt und strukturiert werden können. Genau diese Freiheit ist seine Schwäche: Ohne verbindliche Struktur wächst ein Wiki organisch – jeder Bereich nach eigener Logik, jede Seite nach dem Geschmack ihres Autors. Nach zwei Jahren ist das Wiki selbst eine Informationsinsel: voll mit Wissen, dessen Aktualität, Verbindlichkeit und Zusammenhang niemand mehr beurteilen kann. Dieselbe Information existiert dreimal in unterschiedlichen Versionen; die Suche liefert alle drei.

MORE setzt vor dem Werkzeug an. Die Informationshierarchie legt fest, welche Seiten es gibt, welche Frage jede Seite beantwortet und wie die Seiten zusammenhängen. Ein Wiki kann dann durchaus das Trägersystem sein – aber als bewirtschaftete Struktur, nicht als freie Fläche. Der Unterschied ist derselbe wie zwischen einem Lagerhaus mit Regalsystem und einem Lagerhaus, in dem jeder ablegt, wo Platz ist.

Auch zur klassischen Wissensmanagement-Perspektive besteht ein wichtiger Unterschied: Wissensmanagement wird oft als eigene Disziplin mit eigenen Prozessen betrieben – Wissen wird „gesammelt“, „aufbereitet“ und „geteilt“, typischerweise nach der eigentlichen Arbeit. MORE behandelt Wissen nicht als Sammelobjekt, sondern als Nebenprodukt strukturierter Arbeit: Das Projektwissen entsteht dort, wo gearbeitet wird, in dem Moment, in dem gearbeitet wird. Es muss nicht eingesammelt werden, weil es nie verstreut war.

Am deutlichsten wird der Unterschied am Projektende. Klassisch beginnt dann die Übergabedokumentation: Wissen wird rekonstruiert, kondensiert und in Betriebshandbücher gegossen. In MORE existiert die vollständige Systemdokumentation zu diesem Zeitpunkt bereits – sie ist während des Projekts gewachsen, Anwendungsfall für Anwendungsfall. Die Übergabe an den Betrieb ist keine Dokumentationsphase, sondern eine Übergabe von Strukturzugängen.



14 Verwandte Ansätze und Abgrenzung

Die Frage, wie Projekte ihr Wissen strukturieren, dokumentieren und nachvollziehbar halten, ist nicht neu. Mehrere Disziplinen haben Antworten entwickelt – jede aus ihrer eigenen Perspektive und für ihren eigenen Geltungsbereich. Dieses Kapitel ordnet MORE in diese Landschaft ein. Es zeigt, welche Gedanken bereits publiziert sind, wo MORE anschlussfähig ist und worin der eigentliche Unterschied liegt. Denn ein Framework, das seine Nachbarn kennt, ist glaubwürdiger als eines, das behauptet, im leeren Raum entstanden zu sein.

14.1 Informationsarchitektur im Web- und Content-Umfeld

Der Begriff Informationsarchitektur ist seit den 1990er-Jahren durch das Web- und Usability-Umfeld geprägt. Das Standardwerk von Rosenfeld, Morville und Arango [1] versteht darunter die Strukturierung von Websites und Informationsräumen aus Nutzersicht: Navigation, Benennung, Suche und Auffindbarkeit, entwickelt mit Methoden wie Card Sorting und Tree Testing. Daneben hat sich im Enterprise-Content-Management eine zweite Lesart etabliert: Informationsarchitektur als unternehmensweite Ordnung von Inhalten, Metadaten und Governance – zuletzt verstärkt diskutiert als Voraussetzung für den produktiven Einsatz von KI.

Beide Lesarten teilen mit MORE die Grundüberzeugung, dass Informationen eine bewusst gestaltete Struktur brauchen. Sie beziehen sich jedoch auf Informationsräume für Endnutzer beziehungsweise auf Unternehmensinhalte insgesamt – nicht auf das fachliche Wissen eines Projekts entlang seiner Anforderungen. MORE verwendet den Begriff deshalb bewusst in einer dritten, eigenständigen Bedeutung: Informationsarchitektur als fachliche Struktur des Projektwissens.

14.2 Agile und lebende Dokumentation

Der Gedanke, dass Dokumentation während der Arbeit entstehen soll statt nachträglich, hat in der Softwareentwicklung eine eigene Traditionslinie. Rüping [3] beschreibt bereits 2003 in Musterform, wie schlanke, leichtgewichtige Projektdokumentation entsteht: nur was gebraucht wird, gut organisiert und zugänglich. Ambler [4] formuliert im Agile Modeling ähnliche Prinzipien, darunter die Forderung, Informationen an genau einer Stelle zu pflegen. Adzic [5] zeigt mit Specification by Example, wie ausführbare Spezifikationen zu lebender Dokumentation werden. Am weitesten geht Martraire [2]: Living Documentation entwickelt sich im gleichen Takt wie Design und Entwicklung – Wissen wird dort abgeholt, wo es ohnehin entsteht, und wo möglich automatisiert aufbereitet.

Das **MORE-Prinzip „Dokumentation entsteht während der Bearbeitung“** steht damit in guter Gesellschaft und beansprucht hier keine Erstautorschaft. Der Unterschied liegt im Geltungsbereich: Die genannten Ansätze sind entwickler- und codezentriert. Sie beantworten, wie Entwicklungsteams



ihr technisches Wissen lebendig halten. MORE wendet **dasselbe Prinzip auf das gesamte Projektwissen an** – einschließlich fachlicher Entscheidungen, Abnahmen, Nachweisen und den Informationen des Projektmanagements selbst.

14.3 Single Sourcing und topic-basierte Strukturen

Auch das Prinzip „jede Information genau einmal“ ist etabliert – in der Technischen Redaktion. Topic-basiertes Schreiben und Single Sourcing, standardisiert unter anderem in der Darwin Information Typing Architecture (DITA) [6], zerlegen Inhalte in eigenständige, wiederverwendbare Einheiten, die aus einem Pool in unterschiedliche Zieldokumente zusammengestellt werden. Dokumente sind dort ausdrücklich Sichten auf Inhalte, nicht deren Speicherort – **exakt die Umkehrung der Ordnungseinheit, die auch MORE vollzieht.**

Der Unterschied liegt erneut im Gegenstand: Die Technische Redaktion strukturiert Produkt- und Anwenderdokumentation für die Zeit nach der Entwicklung. MORE strukturiert das Wissen, das **während** der Entwicklung entsteht – vom fachlichen Bedarf bis zum Produktionsnachweis. Beide Welten ergänzen sich: **Eine nach MORE geführte Informationshierarchie ist eine ideale Quelle für topic-basierte Anwenderdokumentation.**

14.4 Strukturvorlagen für technische Dokumentation

Das Ordnungsprinzip „jede Ebene beantwortet genau eine Frage“ hat prominente Verwandte. arc42 [7] beschreibt seine zwölf Gliederungspunkte selbst als Schubladen eines Schrankes, von denen jede eine bestimmte Art von Architekturinformation aufnimmt. Das C4-Modell [8] staffelt Architekturdarstellungen über vier hierarchische Sichten. Diátaxis [9] ordnet technische Dokumentation nach vier Nutzerbedürfnissen – Tutorial, Anleitung, Referenz, Erklärung – und leitet daraus Struktur, Inhalt und Stil ab. Die Norm ISO/IEC/IEEE 42010 [10] formalisiert denselben Gedanken mit Viewpoints und Views für Architekturbeschreibungen.

Alle diese Vorlagen belegen, wie tragfähig das Prinzip klar geschnittener Informationsorte ist. Ihr Geltungsbereich ist jedoch jeweils ein Ausschnitt: die Softwarearchitektur (arc42, C4, ISO 42010) oder die Produktdokumentation (Diátaxis). MORE überträgt das Prinzip auf die Ebene darüber – die fachliche Struktur des gesamten Projekts – und kann die genannten Vorlagen innerhalb einzelner Artefakte problemlos aufnehmen.

14.5 Requirements Engineering, ALM und Traceability

Durchgängige Nachvollziehbarkeit von der Anforderung bis zum Test ist der Kern einer ganzen Werkzeugkategorie: Requirements-Management- und ALM-Plattformen wie IBM DOORS Next, Siemens Polarion, PTC Codebeamer oder Jama Connect [17] positionieren sich als Single Source of Truth und verketten Anforderungen mit Design, Code und Testfällen – insbesondere für regulierte Branchen. Jama Connect liefert mit den Traceability Information Models sogar vorkonfigurierte Artefakttypen und Beziehungsgeflechte je Compliance-Standard. Auf der methodischen Seite definieren die Norm ISO/IEC/IEEE 29148 [11], das IREB-Curriculum [12] und Volere [13] Anforderungsarten, Attribute und Prozesse; Vorgehensmodelle wie das V-Modell XT [14] und PRINCE2 [15] arbeiten mit produktorientierten Ergebnisstrukturen, und DIN 69901 [16] beschreibt die Projektdokumentation als Zusammenstellung wesentlicher Projektinformationen.

Hier ist die Verwandtschaft zu MORE am größten – und zugleich der Unterschied am deutlichsten. Die Werkzeuge liefern Verknüpfungstechnik, aber keine fachliche Struktur; welche Informationen es geben muss und welche Frage jede beantwortet, muss der Anwender selbst festlegen. Die Standards wiederum definieren Anforderungsarten und Prozesse, aber keine gewachsene, anwendungsfallzentrierte Informationshierarchie, in der auch Entscheidungen, Abnahmen und Projektmanagement-Informationen ihren Ort haben. MORE ist werkzeugneutral und setzt genau an dieser Stelle an: als die fachliche Struktur, die ein Werkzeug erst sinnvoll nutzbar macht.

14.6 Einordnung: die Lücke, die MORE schließt

Zusammengefasst: Die Bausteine von MORE haben publizierte Verwandte. Dokumentation, die während der Arbeit entsteht, kennt die agile Literatur [2][3][4][5]. Jede Information genau einmal kennt die Technische Redaktion [6]. Klar geschnittene Informationsorte kennen arc42, Diátaxis und die Architekturnormen [7][9][10]. Durchgängige Traceability kennt die ALM-Welt [17].

Was es bisher nicht gibt, ist die Synthese: ein werkzeugunabhängiges Framework, das eine fachliche Hierarchie aus Fachgebieten und Anwendungsfällen über das gesamte Projektwissen spannt – einschließlich Entscheidungen, Test, Abnahme und der Domänen des Projektmanagements –, in dem jede Ebene genau eine Frage beantwortet und die Dokumentation als Nebenprodukt der Anforderungsarbeit entsteht, bis hin zur regulatorischen Nachweisfähigkeit. Genau diese Verbindung ist der Beitrag von MORE. Es konkurriert nicht mit den genannten Ansätzen, sondern gibt ihnen den gemeinsamen fachlichen Rahmen, der bisher fehlte.



15 MORE im regulierten Umfeld

Seinen Ursprung hat MORE im regulierten Anforderungsmanagement – und dort zeigt die Informationsarchitektur ihren größten Hebel. Denn im regulierten Umfeld reicht „fertig“ nicht aus. Entscheidend ist, ob nachvollziehbar bleibt: was beauftragt wurde, warum es beauftragt wurde, wer es entschieden hat, wie es umgesetzt und geprüft wurde, wer es abgenommen hat und wann es produktiv wurde. Ein Ticketstatus „Done“ beantwortet keine dieser Fragen. Das ist der Unterschied zwischen erledigter Arbeit und nachweisbarer Arbeit.

15.1 Die regulatorische Landschaft: Nachweispflichten nehmen zu

Wer heute in Banken, Versicherungen oder Kritischen Infrastrukturen IT-Vorhaben verantwortet, arbeitet in einem Umfeld, in dem die Nachweispflichten in kurzer Folge deutlich verschärft wurden:

- **Banken und Finanzdienstleister:** Die MaRisk der BaFin verlangen seit jeher ein nachvollziehbares Management von IT-Veränderungen. Seit dem 17. Januar 2025 gilt zusätzlich unmittelbar die EU-Verordnung DORA (Digital Operational Resilience Act): IKT-Risikomanagement, dokumentiertes Änderungsmanagement, Steuerung von IKT-Drittparteien und ein Informationsregister über alle vertraglichen IKT-Abhängigkeiten – jeweils mit der Pflicht, Umsetzung und Wirksamkeit gegenüber der Aufsicht belegen zu können. Die bisherigen BAIT sind in diesem Regime aufgegangen.
- **Versicherungen:** Für sie gilt DORA gleichermaßen, ergänzt um die Anforderungen aus VAG und Solvency II an ein wirksames Governance-System. Auch die VAIT wurden durch das europäische Regime abgelöst.
- **KRITIS-Betreiber und NIS2-Einrichtungen:** Mit dem NIS2-Umsetzungsgesetz, in Kraft seit dem 6. Dezember 2025, wächst der Kreis der beaufsichtigten Unternehmen in Deutschland von rund 4.500 auf etwa 29.500. Sie müssen Risikomanagementmaßnahmen umsetzen und dokumentieren, erhebliche Vorfälle in gestuften Fristen melden (24 Stunden, 72 Stunden, ein Monat) und sich beim BSI registrieren. Verstöße kosten bis zu 10 Millionen Euro oder 2 Prozent des weltweiten Umsatzes – und die Geschäftsleitung haftet persönlich. Seit dem 17. März 2026 ergänzt das KRITIS-Dachgesetz die digitale um die physische Resilienz.
- **Managementsysteme:** ISO/IEC 27001 verlangt „dokumentierte Information“ als gelenkten Bestandteil des ISMS – aktuell, auffindbar, versioniert; ISO 9001 stellt dieselbe Anforderung an das Qualitätsmanagement. Zertifizierungsaudits prüfen nicht, ob Dokumente existieren, sondern ob sie gelebt werden und zusammenpassen.

So unterschiedlich diese Regelwerke sind – ihr gemeinsamer Nenner ist derselbe: Sie verlangen keine Papierberge, sondern belastbare Nachvollziehbarkeit. Gefordert ist der durchgängige Zusammenhang zwischen Anforderung, Entscheidung, Umsetzung, Test, Abnahme und Betrieb. Genau dieser Zusammenhang ist es, der in fragmentierten Projektlandschaften zuerst verloren geht.



15.2 Was Prüfer wirklich fragen

Ob interne Revision, Wirtschaftsprüfer, Zertifizierungsauditor oder Sonderprüfung der Aufsicht – die Prüflogik ist immer dieselbe: Ausgehend von einer Veränderung im System wird der Weg rückwärts verfolgt. Die typischen Fragen lauten:

- **„Wer hat diese Änderung beauftragt, und warum?“** – In MORE beantwortet das der Anwendungsfall (UC) mit seinem fachlichen Ziel und Auslöser.
- **„Welche Entscheidung liegt zugrunde, wer hat sie getroffen?“** – Die Entscheidungen (DEC) hängen am Anwendungsfall: Festlegung, Alternativen, Begründung, Entscheider, Datum.
- **„Wie wurde die Änderung umgesetzt, welche Systeme und Schnittstellen sind betroffen?“** – Prozess (BP), Fachmodell (CM), Oberfläche (UI) und Schnittstellen (API) beschreiben genau das.
- **„Wie wurde getestet, und deckt der Test die Anforderung ab?“** – Die Testfälle (TC) sind je Anwendungsfall geführt; die Abdeckung ist sichtbar statt behauptet.
- **„Wer hat fachlich abgenommen, und wann ging es produktiv?“** – Abnahme (AC) mit Akzeptanzkriterien, Abnahmeentscheidung und Produktionsnachweis.

In der klassischen Projektlandschaft löst jede dieser Fragen eine Suche aus – durch geschlossene Tickets, Chatverläufe, Protokolle und Laufwerke. In MORE ist jede Frage eine Ebene der Informationshierarchie. Der Prüfpfad muss nicht hergestellt werden; er ist die Struktur selbst.

15.3 Wie MORE Nachweisfähigkeit strukturell herstellt

Der entscheidende Punkt: MORE erfüllt regulatorische Anforderungen nicht durch zusätzliche Compliance-Arbeit, sondern durch die Art, wie ohnehin gearbeitet wird.

- **Änderungsmanagement (MaRisk, DORA):** Der Weg einer Anforderung vom Bedarf bis zur Produktion ist der dokumentierte Änderungsprozess – je Anforderung, lückenlos, ohne separates Formularwesen.
- **Dokumentierte Information (ISO 27001, ISO 9001):** Jede Information existiert genau einmal an einem definierten Ort. Lenkung, Aktualität und Auffindbarkeit sind Eigenschaften der Struktur, nicht Ergebnis von Disziplin.
- **Risikomanagement-Dokumentation (NIS2):** Eine aktuelle, fachlich strukturierte Systemdokumentation ist die Voraussetzung jeder belastbaren Risikoanalyse – in MORE existiert sie laufend, nicht erst nach einer Bestandsaufnahme.
- **Drittparteien und Abhängigkeiten (DORA):** Schnittstellen (API) sind je Anwendungsfall beschrieben; welche externen Dienste ein fachlicher Prozess nutzt, ist ablesbar statt rekonstruierbar.

- **Verantwortung der Geschäftsleitung (NIS2, DORA):** Wer persönlich haftet, braucht Transparenz über den Stand der Nachweise. Die MORE-Hierarchie zeigt je Anwendungsfall, welche Ebenen beantwortet sind – und wo Lücken bestehen.

MORE ersetzt dabei weder ein ISMS noch die juristische Bewertung der eigenen Betroffenheit. Es liefert die Informationsarchitektur, in der die geforderten Nachweise als Nebenprodukt der Projektarbeit entstehen – statt als Sonderaufwand vor dem Audit.

15.4 Audit ohne Projektarchäologie

Was das praktisch bedeutet, zeigt der Vergleich. Der klassische Ablauf: Eine Prüfung wird angekündigt, eine Taskforce gebildet. Wochenlang werden Tickets durchsucht, Entscheidungen aus E-Mails rekonstruiert, Testnachweise eingesammelt, Abnahmen nachgetragen – gebundene Kapazität, angreifbare Lücken, und am Ende Feststellungen, die weniger mit der tatsächlichen Arbeit als mit ihrer fehlenden Dokumentation zu tun haben.

Mit MORE kehrt sich das Bild um: Die Prüfung beginnt bei der Informationshierarchie. Jede Stichprobe – „zeigen Sie mir den Weg dieser Anforderung“ – führt entlang der Ebenen vom Anwendungsfall bis zum Produktionsnachweis. Was der Prüfer sieht, ist keine für ihn erstellte Dokumentation, sondern der Arbeitsstand des Projekts. Das ist der Zustand, den man auditfähig by design nennen kann: Die Organisation ist jederzeit prüfbar, weil Nachvollziehbarkeit nicht hergestellt, sondern mitgeführt wird.

Für Banken, Versicherungen und KRITIS-Betreiber ist das mehr als Komfort. Es senkt Prüfungsaufwand und Feststellungsrisiko, entlastet die Fachbereiche vor Audits – und es verwandelt eine Pflicht in einen Standortvorteil: Wer Nachweisfähigkeit strukturell besitzt, kann regulatorische Veränderungen aufnehmen, ohne jedes Mal von vorn zu beginnen.

16 Der Nutzen im Überblick

Eine Informationsarchitektur ist kein Selbstzweck. Ihr Nutzen zeigt sich daran, dass jede Rolle im Projekt schneller und verlässlicher an die Informationen kommt, die sie braucht – und dass Informationen nicht mehrfach erarbeitet werden müssen.

Zusammengefasst bewirkt MORE:

- Projektwissen bleibt nachvollziehbar – über Teamwechsel, Projektphasen und das Projektende hinaus.
- Informationen sind fachlich statt technisch organisiert – die Struktur folgt dem Projekt, nicht den Werkzeugen.
- Dokumentation wächst kontinuierlich mit dem Projekt – ohne eigene Dokumentationsphase.
- API-, UI- und Testbeschreibungen können unabhängig gepflegt und projektspezifisch zusammengestellt werden.
- Am Projektende existiert bereits eine vollständige Systemdokumentation.

Rolle	Konkreter Nutzen
Entwicklerinnen und Entwickler	Vollständiger fachlicher Kontext am Anwendungsfall statt Ticket-Archäologie; Entscheidungen und Begründungen am Ort der Umsetzung.
Product Owner	Fachlicher Zusammenhang der Backlog-Einträge bleibt sichtbar; Refinement füllt eine dauerhafte Struktur statt vergänglicher Tickets.
Projektleitung	Steuerungs- und Fachinformationen in einer Architektur; Risiken und Entscheidungen mit direktem Bezug zu den betroffenen Anwendungsfällen.
Qualitätssicherung / Test	Klare Zuordnung: Welche Anforderung wird durch welche Tests abgedeckt – Testabdeckung wird sichtbar statt behauptet.
Revision / Audit / Compliance	Durchgängiger Prüfpfad vom Bedarf über Entscheidung, Umsetzung und Test bis zur Produktion – ohne nachträgliche Rekonstruktion.
Betrieb	Gewachsene, aktuelle Systemdokumentation zur Übernahme; das „Warum“ hinter dem Systemverhalten bleibt auffindbar.
Neue Teammitglieder	Einarbeitung entlang der fachlichen Struktur statt mündlicher Überlieferung – das Projekt erklärt sich selbst.

17 Fazit und Ausblick

Moderne Projekte scheitern selten an fehlenden Informationen. Sie leiden darunter, dass ihr Wissen fragmentiert ist – verteilt auf Werkzeuge, die jeweils ihre eigene Ordnung mitbringen, aber keine gemeinsame fachliche Struktur kennen. Mehr Werkzeuge, bessere Vorlagen und diszipliniertere Ablage lösen dieses Problem nicht, weil sie an der falschen Einheit ansetzen: am Dokument statt an der Information.

Die Antwort ist eine Informationsarchitektur: eine fachliche Struktur, in der jede Information genau einmal existiert, genau eine Frage beantwortet und mit den Informationen verbunden ist, zu denen sie fachlich gehört. Das MORE Framework liefert diese Architektur – als Hierarchie aus Fachgebieten und Anwendungsfällen, an denen Prozess, Fachmodell, Oberfläche, Schnittstellen, Testfälle, Abnahme und Entscheidungen hängen.

Dabei ist MORE bewusst kein Ersatz für Bestehendes: Scrum, Kanban oder klassische Projektsteuerung organisieren weiterhin die Zusammenarbeit; Jira, Confluence und Git bleiben die Werkzeuge. MORE liefert die Informationsarchitektur, innerhalb derer diese Werkzeuge eingesetzt werden – die fachliche Struktur, die ein Werkzeug erst sinnvoll nutzbar macht.

Vielleicht ist das die wichtigste Erkenntnis dieses Whitepapers: MORE ist kein Dokumentationsframework. Die Hierarchie beschreibt den Weg einer Anforderung von der Idee bis zum Nachweis – und die Dokumentation ist die Spur, die dieser Weg hinterlässt. Traceability, vollständige Systemdokumentation und Prüffähigkeit sind keine Zusatzleistungen, sondern Nebenprodukte sauberer, strukturierter Arbeit.

MORE wird weiterentwickelt: um ausgearbeitete Beispiele, Templates für gängige Werkzeuglandschaften und Leitfäden für die Einführung in bestehende Projekte. Aktuelle Informationen, Ressourcen und Kontaktmöglichkeiten finden sich unter more-framework.org.

18 Quellen und weiterführende Literatur

- [1] Rosenfeld, L.; Morville, P.; Arango, J.: Information Architecture. For the Web and Beyond. 4. Auflage, O'Reilly, 2015.
- [2] Martraire, C.: Living Documentation. Continuous Knowledge Sharing by Design. Addison-Wesley, 2019.
- [3] Rüping, A.: Agile Documentation. A Pattern Guide to Producing Lightweight Documents for Software Projects. Wiley, 2003.
- [4] Ambler, S. W.: Agile Modeling. Effective Practices for eXtreme Programming and the Unified Process. Wiley, 2002; agilemodeling.com.
- [5] Adzic, G.: Specification by Example. How Successful Teams Deliver the Right Software. Manning, 2011.
- [6] OASIS: Darwin Information Typing Architecture (DITA). OASIS Standard; siehe auch Topic-based Authoring und Single Sourcing in der Technischen Redaktion.
- [7] Starke, G.; Hruschka, P.: arc42 – Ressourcen für Architekturdokumentation. arc42.org.
- [8] Brown, S.: The C4 Model for Visualising Software Architecture. c4model.com.
- [9] Procida, D.: Diátaxis. A Systematic Approach to Technical Documentation Authoring. diataxis.fr.
- [10] ISO/IEC/IEEE 42010: Software, Systems and Enterprise – Architecture Description.
- [11] ISO/IEC/IEEE 29148: Systems and Software Engineering – Life Cycle Processes – Requirements Engineering.
- [12] IREB e. V.: Certified Professional for Requirements Engineering (CPRE) – Lehrplan Foundation Level. ireb.org.
- [13] Robertson, S.; Robertson, J.: Mastering the Requirements Process. Getting Requirements Right (Volere). 3. Auflage, Addison-Wesley, 2012.
- [14] V-Modell XT. Der Beauftragte der Bundesregierung für Informationstechnik (Hrsg.).
- [15] AXELOS: Managing Successful Projects with PRINCE2 (Product-based Planning). 7. Auflage, 2023.
- [16] DIN 69901: Projektmanagement – Projektmanagementsysteme. Beuth Verlag.
- [17] Produktdokumentation der Hersteller: IBM DOORS Next; Siemens Polarion ALM; PTC Codebeamer; Jama Connect (Traceability Information Models).
- [18] MORe Framework: Reguliertes Anforderungsmanagement von der Idee bis zur Produktion. more-framework.org.
- [19] Verordnung (EU) 2022/2554 über die digitale operationale Resilienz im Finanzsektor (DORA); anwendbar seit 17. Januar 2025.



- [20] Richtlinie (EU) 2022/2555 (NIS-2-Richtlinie); umgesetzt durch das NIS2-Umsetzungsgesetz (NIS2UmsuCG), in Kraft seit 6. Dezember 2025.
- [21] KRITIS-Dachgesetz (Umsetzung der Richtlinie (EU) 2022/2557 – CER), in Kraft seit 17. März 2026.
- [22] BaFin: Mindestanforderungen an das Risikomanagement (MaRisk), Rundschreiben in der jeweils gültigen Fassung.
- [23] ISO/IEC 27001: Informationssicherheits-Managementsysteme – Anforderungen (insbesondere „dokumentierte Information“).
- [24] Bundesamt für Sicherheit in der Informationstechnik (BSI): BSI-Gesetz in der Fassung des NIS2-Umsetzungsgesetzes; Informationen zu Registrierung, Melde- und Nachweispflichten. [bsi.bund.de](https://www.bsi.bund.de).

MORe organisiert nicht Dokumente. Nicht Tickets. Nicht Dateien.

MORe liefert die Informationsarchitektur moderner Projekte.