

# codebeamer: Agile ALM für smarte Produktentwicklung

Category: Online-Marketing

geschrieben von Tobias Hager | 5. Februar 2026



# codebeamer: Agile ALM für smarte Produktentwicklung

Du entwickelst smarte Produkte, jonglierst mit Anforderungen, Bugs, Releases und Compliance-Vorgaben – und dein Tooling fühlt sich trotzdem an wie Windows 95 auf einem Touchscreen? Dann wird's Zeit, dass du codebeamer kennenzulernen. Das Agile ALM-Werkzeug, das dir nicht nur den Rücken freihält, sondern dir endlich ein System gibt, das mitdenkt, mitwächst und mitentwickelt. Schluss mit Excel-Hölle, SharePoint-Monstern und Jira-Workarounds – jetzt wird's ernst.

- Was codebeamer eigentlich ist – und warum es mehr als nur ein ALM-Tool ist
- Wie Agile ALM deine Produktentwicklung auf das nächste Level hebt
- Warum Requirements Management, Testmanagement und DevOps in einem Tool

Sinn machen

- Wie codebeamer regulatorische Anforderungen in sicherheitskritischen Branchen abdeckt
- Warum klassische Tools wie Jira, Confluence und Excel einfach nicht mehr reichen
- Wie du mit codebeamer durchgängige Traceability – von der Anforderung bis zum Release – realisierst
- Was codebeamer von klassischen PLM- oder PPM-Systemen unterscheidet
- Welche Integrationen, APIs und Schnittstellen codebeamer wirklich smart machen
- Wie du codebeamer in bestehende DevOps- und CI/CD-Pipelines einbindest
- Warum codebeamer nicht nur für Techies, sondern auch für Auditoren ein Traum ist

# Was ist codebeamer? Agile ALM neu gedacht

codebeamer ist ein Application Lifecycle Management (ALM) System – aber nicht irgendeins. Es ist die agile Antwort auf die überfrachteten, rückwärtsgewandten Toolchains, die viele Produktentwickler heute noch mit sich herumschleppen. codebeamer vereint Requirements Management, Testmanagement, Risikoanalyse, Qualitätsmanagement, DevOps-Integration und Compliance-Tracking in einer einzigen Plattform. Nicht als Flickenteppich, sondern als durchgängiges System mit echter End-to-End-Traceability.

Das Tool wurde ursprünglich für hochregulierte Branchen wie Automotive, Medizintechnik und Luftfahrt entwickelt – also genau dort, wo Nachverfolgbarkeit, Validierung und Auditierbarkeit nicht nur Buzzwords, sondern gesetzliche Anforderungen sind. Damit ist codebeamer nicht nur agil, sondern auch normkonform. ISO 26262, IEC 62304, ASPICE, FDA CFR 21 Part 11 – pick your poison. codebeamer kann sie alle.

Im Kern basiert codebeamer auf einer vollständig modellierbaren Datenstruktur. Work Items, Artefakte, Relationen und Prozesse lassen sich granular konfigurieren, versionieren und auditieren. Das bedeutet: Du kannst deine gesamte Produktentwicklung – von der ersten Anforderung bis zum ausgelieferten Release – nicht nur abbilden, sondern auch mit einem Klick zurückverfolgen. Klingt nach Science Fiction? Willkommen im Jahr 2025.

Anders als klassische ALM-Systeme verfolgt codebeamer einen API-zentrierten, offenen Ansatz. Es gibt keine proprietären Lock-ins oder abgeschotteten Dateninseln. Stattdessen bekommst du eine RESTful API, ein flexibles Datenmodell, BPMN-Prozesseditor und Integrationen zu gängigen DevOps-Tools wie Jenkins, Git, Azure DevOps, Jira, Polarion oder SAP Solution Manager. Du arbeitest hybrid? codebeamer auch.

Und ja, die UI ist nicht nur nutzbar, sondern tatsächlich angenehm – was man von vielen Enterprise-Tools nicht behaupten kann. Moderne Webtechnologie, responsive Design, rollenbasierte Dashboards und ein Collaboration-Modul, das

nicht wie ein Slack-Klon aus 2012 wirkt. Kurz: codebeamer ist ein ALM, das man freiwillig benutzt.

# Agile ALM: Warum Produktentwicklung mehr braucht als Jira-Tickets

Agile Produktentwicklung ist mehr als Daily Stand-ups und Backlog Grooming. Es geht um inkrementelle Wertschöpfung, frühes Feedback, permanente Anpassung – und das bei gleichzeitig wachsender Komplexität, kürzeren Time-to-Markets und steigenden regulatorischen Anforderungen. Wer das mit Jira, Excel und E-Mail managegen will, landet früher oder später im Chaos.

Agile ALM heißt: Du brauchst ein System, das nicht nur deine Tasks verwaltet, sondern deine gesamte Entwicklungslogik abbildet. Anforderungen, Use Cases, Tests, Bugs, Releases, Risikoanalysen, Dokumentation, Validierung – alles in einem konsistenten Modell, verbunden durch nachvollziehbare Relationen. Nur dann kannst du wirklich agil UND auditierbar arbeiten.

codebeamer liefert genau das. Es kombiniert agile Methoden wie Scrum, Kanban oder SAFe mit normkonformen Prozessen und auditierbaren Artefakten. Du kannst User Stories genauso modellieren wie Sicherheitsanforderungen nach ISO 26262 oder Testfälle nach GAMP 5. Und du kannst sie alle miteinander verknüpfen – automatisch, bidirektional und versioniert.

Zudem bietet codebeamer eine leistungsfähige Prozess-Engine auf Basis von BPMN 2.0. Damit kannst du deine Workflows nicht nur dokumentieren, sondern tatsächlich ausführen. Genehmigungsprozesse, Change Requests, CAPA-Prozesse, Risk Assessments – alles modellierbar, auditierbar und durchsetzbar. Agile Governance auf Enterprise-Niveau.

Und weil codebeamer rollenbasiert arbeitet, sehen Entwickler, Tester, Produktmanager und Auditoren jeweils genau das, was sie brauchen – nicht mehr, nicht weniger. Das erhöht nicht nur die Effizienz, sondern senkt auch die Fehlerquote. Denn ein Prozess, der nicht gelebt wird, ist kein Prozess. Punkt.

# Traceability in codebeamer: Von der Anforderung bis zum Bugfix

Traceability ist das Buzzword, das in jedem Auditbericht auftaucht – und das in der Praxis oft gnadenlos scheitert. In codebeamer dagegen ist Traceability kein Add-on, sondern Grundprinzip. Jede Anforderung, jede Änderung, jeder

Test und jede Release-Aktivität ist mit allem anderen verknüpfbar – automatisch, regelbasiert und versioniert.

Das beginnt mit dem Requirements Management. Du definierst Anforderungen in hierarchischen Strukturen, versiehst sie mit Attributen und Relationen, verlinkst sie mit User Stories, Epics, Testfällen oder Code Commits. Jede Änderung wird historisiert, jede Version dokumentiert. Und das alles, ohne ein Excel-Massaker zu hinterlassen.

Im Testmanagement kannst du Testfälle definieren, parametrisieren, ausführen und mit Anforderungen und Fehlerberichten verknüpfen. Die Testergebnisse werden automatisch den jeweiligen Anforderungen zugeordnet – und du bekommst einen Live-Traceability-Report auf Knopfdruck. Keine manuelle Nachbearbeitung, kein Copy-Paste-Wahnsinn.

Auch für Risiko- und Fehleranalysen bietet codebeamer spezialisierte Module. FMEA, Hazard Analysis, Fault Tree Analysis – alles direkt im System, direkt verknüpft mit deinen Anforderungen, Tests und Maßnahmen. Und sobald du ein Item änderst, wird die gesamte Trace-Kette aktualisiert. Du musst nichts nachpflegen. Es passiert einfach.

Das Highlight: Der Traceability Matrix Editor. Ein interaktives Grid, in dem du die Relationen zwischen Anforderungen, Tests und Artefakten visuell modellieren kannst. Drag & Drop, Filter, Versionierung – inklusive Audit-Log. Damit überzeugst du nicht nur deinen QA-Manager, sondern auch jeden externen Auditor. Garantiert.

## Compliance, Normen, Audits: codebeamer in regulierten Branchen

Smarte Produktentwicklung endet nicht beim MVP. Spätestens wenn du in regulierten Märkten unterwegs bist – Automotive, Medizintechnik, Luftfahrt, Bahn, Verteidigung – brauchst du mehr als Agile Dogma. Du brauchst Nachweise. Und zwar in Form von Prozessdefinitionen, Audit-Trails, Validierungsdokumentation und konformer Toolunterstützung. codebeamer liefert all das – out of the box.

Das System ist validierungsfähig nach GAMP 5, unterstützt elektronische Signaturen gemäß FDA CFR 21 Part 11 und erfüllt Anforderungen aus ISO 13485, ISO 26262, IEC 61508, IEC 62304 oder ASPIKE. Du kannst Audit-Trails, Signaturen, Genehmigungsworflows und Versionsstände vollständig dokumentieren – mit einem Klick.

Besonders mächtig ist das Dokumentations-Framework. Mit sogenannten „Baselines“ frierst du Artefaktstände ein – inklusive aller Relationen, Anhänge und Metadaten. Diese Baselines kannst du versionieren, vergleichen und als PDF oder Word exportieren. Damit generierst du Prüfberichte, Release-

Dokumentationen oder Design History Files direkt aus dem System – ohne Copy-Paste und ohne Toolwechsel.

Zusätzlich unterstützt codebeamer sogenannte Review-Workflows. Hier werden Items wie Anforderungen oder Testfälle durch definierte Prüfer validiert, mit eSignaturen versehen und freigegeben. Jeder Schritt wird protokolliert, jede Entscheidung dokumentiert. Kein Wildwuchs, keine Excel-Workarounds, keine Audit-Schweißausbrüche.

Und falls du mit Lifecycle-Modellen wie V-Modell, SPICE oder FDA-Waterfall arbeitest – kein Problem. codebeamer lässt sich entsprechend konfigurieren und bietet Templates für gängige Normen und Modelle. Du musst also nicht bei null anfangen, sondern kannst direkt mit einem validierten Setup loslegen.

## API, DevOps, CI/CD: codebeamer und die smarte Toolchain

Ein gutes ALM-System muss nicht alles können – aber sich überall einklinken. Und genau das macht codebeamer so stark. Es bietet eine umfassende REST API, Webhooks, Custom Connectors und eine offene Architektur, mit der du das Tool nahtlos in deine bestehende DevOps-Umgebung integrierst.

Du nutzt Jenkins für Continuous Integration? GitLab für Source Code Management? JIRA für Ticketing? Kein Problem. codebeamer lässt sich mit diesen Systemen synchronisieren – bidirektional, eventbasiert und kontrolliert. Du definierst, welche Artefakte wie gemappt werden, und codebeamer kümmert sich um die Synchronisation.

Auch für CI/CD-Pipelines bietet das System Hooks, Trigger und Statusinformationen. Du kannst Builds, Testläufe oder Deployments mit Artefakten in codebeamer verknüpfen – inklusive Statusanzeigen, Logs und Rückverlinkungen. Damit wird deine Toolchain nicht nur transparent, sondern auch steuerbar.

Für Entwickler gibt's SDKs, Command Line Interfaces (CLI) und sogar ein Plugin-System für eigene Erweiterungen. Damit kannst du codebeamer exakt auf deine Prozesse anpassen – ohne auf Updates oder Roadmaps des Herstellers zu warten. Du brauchst ein spezielles Review-Modul? Bau es dir. Du willst automatisch Artefakte aus Code-Commits erzeugen? Mach es.

Und weil das System mandantenfähig ist, kannst du mehrere Projekte, Teams oder Abteilungen parallel betreiben – mit individuellen Konfigurationen und gemeinsamen Plattformstandards. Das ist nicht nur flexibel, sondern auch skalierbar. Gerade für Konzerne oder OEM-Zulieferer ein echtes Argument.

# Fazit: codebeamer ist kein Tool – es ist ein Gamechanger

Wer heute smarte Produkte entwickelt, braucht mehr als ein Taskboard. Er braucht ein System, das den gesamten Lifecycle abbildet – von der Idee bis zum Audit. codebeamer ist genau dieses System. Es kombiniert Agile Methoden mit normkonformer Governance, tracebarer Dokumentation und technischer Offenheit. Kein Flickenteppich, kein Tool-Zoo, kein Dublettenwahnsinn – sondern ein integriertes ALM, das tatsächlich funktioniert.

Ob du in der Medizintechnik arbeitest, Automotive-Projekte koordinierst oder Embedded Systems entwickelst: Wenn du ernsthaft agil UND compliant sein willst, führt an codebeamer kein Weg vorbei. Denn das Tool liefert nicht nur Funktionen, sondern ein Framework, das deine Produktentwicklung auf das nächste Level hebt. Willkommen im echten ALM-Zeitalter.