

Projektmanagement, Requirements Engineering & mehr



Application Lifecycle Management mit *objectiFRPM*[®]

Whitepaper

microTOOL
making IT better

Projektmanagement, Requirements Engineering und mehr

Application Lifecycle Management mit *objectiF RPM*

Vom Stakeholder-Bedarf zu IT-Systemen für erfolgreiches Business

Unternehmen und öffentliche Verwaltung erleben heute tiefgreifende Veränderungen, die sich auf Geschäftsmodelle, Geschäftsprozesse und Produkte auswirken. Die IT ist mehr denn je gefordert, innovative Lösungen zeitnah zu liefern. Dem stehen oft zwei Hindernisse im Weg: eine schwerfällige, heterogene Entwicklungsumgebung und fehlende methodische Ausdrucksmittel speziell für Geschäfts- und Anforderungsanalyse.

Wie kann man diese Hindernisse überwinden?

Unsere Antwort lautet: mit **Application Lifecycle Management** (ALM) unterstützt durch *objectiF RPM*.

Nachfolgend erfahren Sie, wie Application Lifecycle Management mit *objectiF RPM* Ihnen hilft,

- Projekte schnell einzurichten und agil, klassisch oder hybrid zu planen,
- Termine, Kosten und Ressourceneinsatz immer im Blick zu haben,
- Bedarf, Ziele und Anforderungen der Stakeholder zu ermitteln und in erwartungsgerechte Lösungen umzusetzen,
- effizient im Team zusammenzuarbeiten,
- Traceability vom Stakeholder-Bedarf bis zum Code zu nutzen,
- Revisionsicherheit und Compliance sicherzustellen – als Voraussetzung für eine hohe Prozess- und Lösungsqualität.

Viel Spaß beim Erkunden von *objectiF RPM*.

Inhalt

<i>objectiF RPM</i> im Überblick.....	3
Tour durch den Application Lifecycle mit <i>objectiF RPM</i>	4
Der Einstieg: Stakeholder und Kontext analysieren	4
Requirements Engineering: Herausforderungen und Lösungen	6
Anforderungen und ihre Beziehungen nachvollziehbar entwickeln	6
Änderungsauswirkungen sichtbar machen	8
Anforderungen prüfen.....	9
Lasten- oder Pflichtenheft schnell erstellen.....	10
Die unterstützten UML- und SysML-Diagramme.....	11
Design & Implementierung: Herausforderungen und Lösungen	12
Systementwurf aus Anforderungen entwickeln.....	12
Model-based Development „light“: Designmodelle zur Implementierung nutzen	14
Deployment: Systeme in der Cloud bereitstellen	15
Traceability bis zum Code sicherstellen.....	16
Testmanagement	18
Video-Links zu Design & Implementierung	18
Projektmanagement: Herausforderungen und Lösungen	19
Projekte anlegen, strukturieren, planen und skalieren	19
Aktivitätsorientiert oder agil planen	20
Änderungen der Projektplanung nachvollziehen	21
Mitarbeiterinsatz und Kosten planen	22
Projekt-Controlling	23
<i>objectiF RPM</i> auf Ihre Organisation zuschneiden	25
<i>objectiF RPM</i> anpassen	25
<i>objectiF RPM</i> erweitern	25
Zustandsbasierte Workflows definieren	26
Video-Links zum Erweitern von <i>objectiF RPM</i>	26
Technische Details	26
Infrastruktur	27
Einsatzszenarios	28
Schnittstellen.....	29
<i>objectiF RPM</i> kennenlernen	29

objectiF RPM im Überblick

objectiF RPM ist Software für Application Lifecycle Management (ALM). Das bedeutet: *objectiF RPM* unterstützt IT-Teams vom erkannten Bedarf der Stakeholder bis zur bereitgestellten Lösung.

Es bietet Funktionen für **Requirements Engineering, Design & Implementierung, Qualitätssicherung & Testmanagement, Projektmanagement** einschließlich Kostenplanung und **Ressourcenmanagement, Risikomanagement** sowie **Reporting & Dokumentation**.

objectiF RPM richtet sich damit an Projektmanager, Product Owner, Business Analysten, Requirements Engineers, Software- und Systementwickler, QS-Verantwortliche und Tester. Für sie alle stellt *objectiF RPM* eine gemeinsame Infrastruktur für Zusammenarbeit und Kommunikation zur Verfügung.

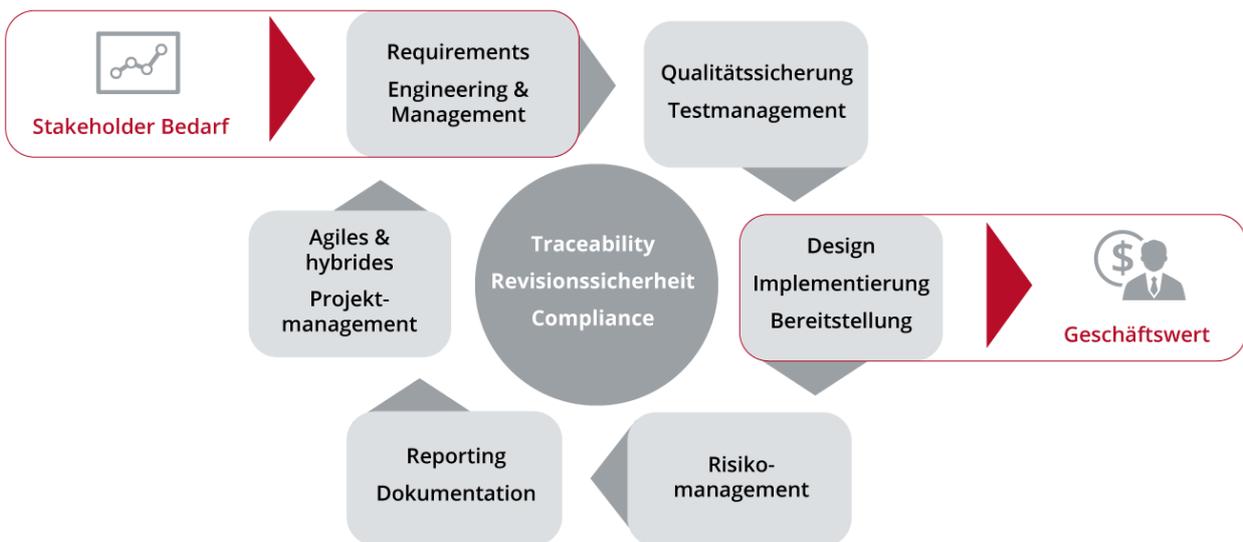


Abbildung 1: Stakeholder-Bedarf mithilfe von Software und Systemen in Geschäftswert umsetzen – das ist das Ziel von Application Lifecycle Management. *objectiF RPM* bietet Ihnen dafür Struktur und Mittel

Über Bilder kann man besser reden als über Text und Code. Deshalb finden Sie in *objectiF RPM* zahlreiche Diagramme für Requirements Engineering sowie Design & Implementierung. *objectiF RPM* basiert auf bewährten Standards und unterstützt damit ein einheitliches, methodisches Vorgehen.

Sie können *objectiF RPM* vor Ort im lokalen Netzwerk oder in der Cloud einsetzen. Sie können per Browser oder mit Windows Clients an Ihren Projekten arbeiten. Egal, für welches technische Szenario Sie sich entscheiden, *objectiF RPM* bietet Ihnen Mehrbenutzer-Zugriff auf die Echtzeit-Ergebnisse Ihrer Projekte. Für die sichere Verwaltung Ihrer Projektergebnisse übernimmt *objectiF RPM* das **Versions- & Konfigurationsmanagement**, das **Backlog-Management**, das **Dokumentenmanagement** sowie die **Code-Verwaltung** und die **Projektechzeitauswertung**.

Der Erfolg von ALM-Software hängt wesentlich davon ab, wie gut sie sich in die technische Infrastruktur integrieren sowie an die Organisationskultur und vorhandene Prozesse anpassen lässt. Hier unterscheidet sich *objectiF RPM* deutlich von vergleichbaren Lösungen: Sie können *objectiF RPM* für Ihre Organisation, Projekte und Workflows konfigurieren. Mehr noch: Sie können es bedarfsgerecht um neue Typen von Projektergebnissen mit eigener Benutzeroberfläche erweitern. Sie können selbst Funktionen entwickeln und sie in Menüs und Workflows integrieren. Und schließlich können Sie Schnittstellen zu vorhandenen Tools realisieren.

Drei Gründe für *objectiF RPM*



1. Mehr Nachvollziehbarkeit für bessere Planung

Stakeholder ändern ihre Anforderungen, ein Mitarbeiter fällt aus, Reviews und Tests dauern länger als erwartet – das sind alltägliche Projekt Ereignisse, die Arbeitsfortschritt, Termin-, Kosten- und Ressourcenplanung beeinflussen. Mit *objectiF RPM* behalten Sie den Überblick darüber, welche Folgen Anforderungsänderungen haben, an welchen Anforderungen gearbeitet wird, wie stark das Team ausgelastet ist, in welchem Zustand sich Anforderungen, Reviews, Testfälle, Test und Dokumente befinden und welche Kosten im Projekt aufgelaufen sind. Die Grundlage dafür ist die Nachvollziehbarkeit der Anforderungen bis zum Code sowie zu Projektaktivitäten in *objectiF RPM*.



2. Mehr Struktur für höhere Prozess- und Lösungsqualität

Strukturieren hilft, komplexe Projekte beherrschbar zu machen. Deshalb werden in *objectiF RPM* Projekte mithilfe von Vorlagen eingerichtet. Eine Projektvorlage legt die Struktur an, in der die Projektergebnisse abgelegt werden und erzeugt einen initialen Projektplan. Er bildet bereits die wichtigsten Aktivitäten Ihres Entwicklungsprozesses mit Meilensteinen und Kontrollflüssen ab. Eine Projektvorlage bringt außerdem Ergebnistypen, Backlogs, Workflows, Abfragen, Charts und vieles mehr mit. Sie können Projektvorlagen auf Ihren Prozess zuschneiden und kontinuierlich verbessern – für ein standardisiertes Vorgehen und eine einheitliche Qualität in Ihren Projekten.



3. Mehr Geschwindigkeit für größere Effizienz

Neue Projekte aufsetzen, die Planung fortschreiben und Projekte skalieren – mit *objectiF RPM* bedeutet das alles kaum Aufwand für Sie. Denn in jedem per Vorlage erzeugten Projekt finden Sie Planungsmuster vor, die Ihnen helfen, neue Releases, Sprints, Work-Packages und zusätzliche Teams mit wenigen Klicks anzulegen. Der Projektplan wird dabei automatisch fortgeschrieben. Backlogs, Planungssichten, Auswertungen und Dashboards – kurz alles, was Sie für die Planung und Steuerung Ihres Projekts brauchen – erzeugen die Muster gleich mit. Jedes Projekt enthält außerdem Dokumentenvorlagen z. B. für Lastenhefte. So erfinden Sie nicht in jedem Projekt das Rad neu und können ohne Rüstzeit starten. Wie sieht die Arbeit mit *objectiF RPM* konkret aus? Kommen Sie mit uns auf eine ...

Tour durch den Application Lifecycle mit *objectiF RPM*

Der Einstieg: Stakeholder und Kontext analysieren

Ein neues System muss vor allem eines: den Bedarf der Stakeholder erfüllen, indem es den erwarteten Geschäftswert schafft. Die erste Aufgabe im Application Lifecycle besteht deshalb darin, alle – wirklich alle – Stakeholder mit ihrem Bedarf, ihren Zielen und möglichen Konflikten zu identifizieren. Geschieht das nicht, besteht die Gefahr, dass wichtige Anforderungen an die neue Lösung gar nicht erst formuliert werden. Außerdem brauchen Sie eine klare Antwort auf die Frage nach dem Scope des zu entwickelnden

Systems. Sonst bekommen Sie Termine und Kosten nur schwer in den Griff. Für die Stakeholder-Analyse und Abgrenzung des Scopes bieten sich grafische Mittel an.

In *objectiF RPM* finden Sie deshalb:

- **Zieldiagramme**, die dabei helfen, die Stakeholder mit ihren Beziehungen, Zielen und Zielkonflikten sichtbar zu machen,
- **Kontextdiagramme**, die das System abgrenzen, indem sie Anwender, externe Schnittstellen und die dazwischen fließenden Informationen abbilden.

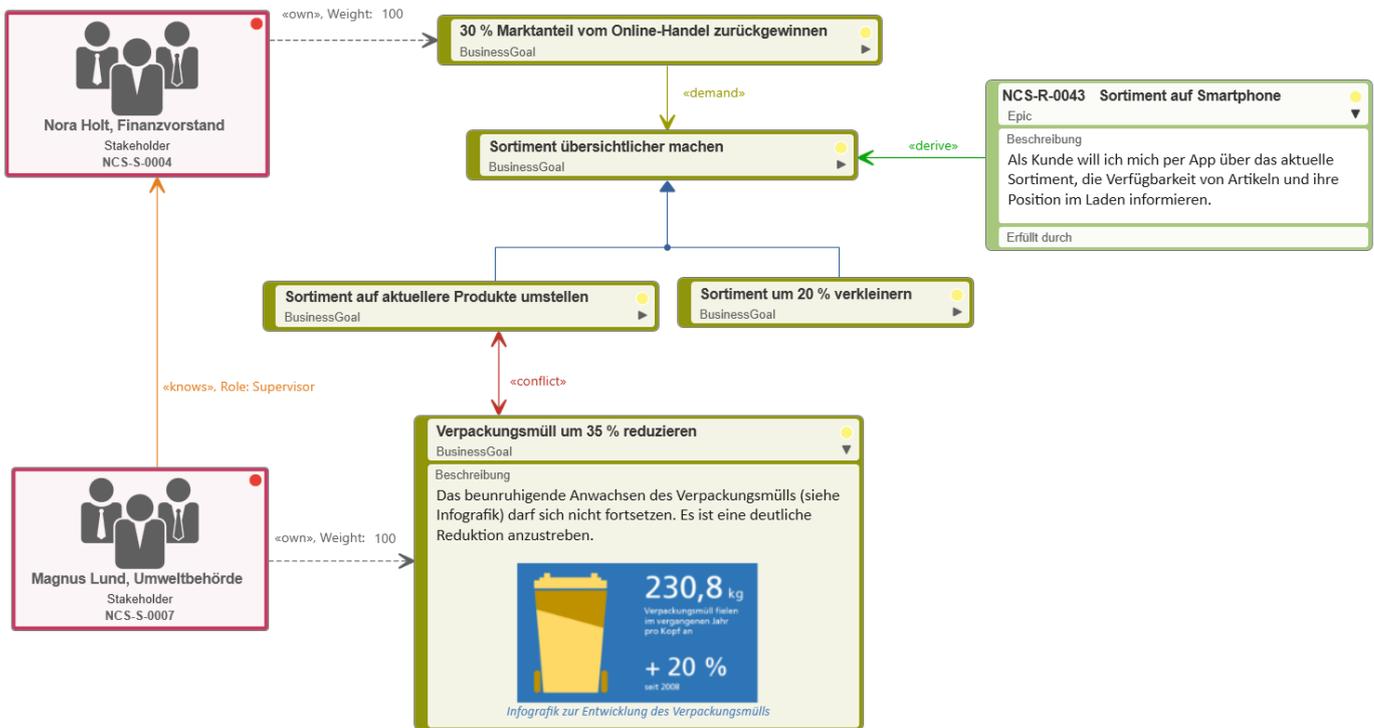


Abbildung 2: Zieldiagramm als ein visuelles Mittel für die Analyse der Stakeholder, ihrer Beziehungen und Ziele

Beschreibungsformulare für Stakeholder und Ziele sorgen für eine einheitliche Dokumentation. Auswertungen in Form von konfigurierbaren Ziel- und Stakeholder-Listen erleichtern das Abstimmen und Priorisieren der Ziele mit den Stakeholdern. Ziele ändern sich. Mit *objectiF RPM* machen Sie Änderungen nachvollziehbar, indem Sie Ziele und Zieldiagramme versionieren.

Weitere Funktionen für die Stakeholder-Analyse

- ✓ Export von Listen der Stakeholder und ihrer Ziele nach MS Excel
- ✓ Import von Zielen aus MS Excel
- ✓ Generieren von MS Word Dokumenten aus Stakeholder- und Zielbeschreibungen
- ✓ Teilen von Zielen per E-Mail zur Abstimmung mit Stakeholdern

Requirements Engineering: Herausforderungen und Lösungen

Die Ziele der Stakeholder sind ein Ausgangspunkt für die Ermittlung von Anforderungen. Die Anforderungen müssen dann schrittweise verfeinert werden, damit sie gemeinsam mit den Stakeholdern bewertet, priorisiert und in die Entwicklung eingeplant werden können.

Zentrale Herausforderungen dabei sind,

- die Anforderungen und ihre komplexen Zusammenhänge zu verstehen und für Änderungen nachvollziehbar zu machen,
- Qualität der Anforderungen sicherzustellen,
- Anforderungen mit möglichst geringem Aufwand im Lasten- oder Pflichtenheft zu dokumentieren und die Dokumentation aktuell zu halten,
- mit großen Mengen an Anforderungen umzugehen,
- und bei all dem das Big Picture der angestrebten Lösung nicht aus den Augen zu verlieren.

Anforderungen und ihre Beziehungen nachvollziehbar entwickeln

Auf dem Weg zu den Anforderungen an eine neue Lösung sind Artefakte wie Use Cases, Personas, Geschäftsprozessmodelle, Business Entitäten etc. hilfreich. Deshalb finden Sie in *objectiF RPM* visuelle Strukturierungsmittel aus UML und SysML, die Ihnen helfen, das Big Picture der Lösung festzuhalten. Dazu gehören Use Case Diagramme, Swimlane-Diagramme, Package-, Block- und Klassendiagramme. Vor allem aber können Sie Anforderungen mit ihren vielfältigen Beziehungen und Abhängigkeiten in **Anforderungsdiagrammen** modellieren.

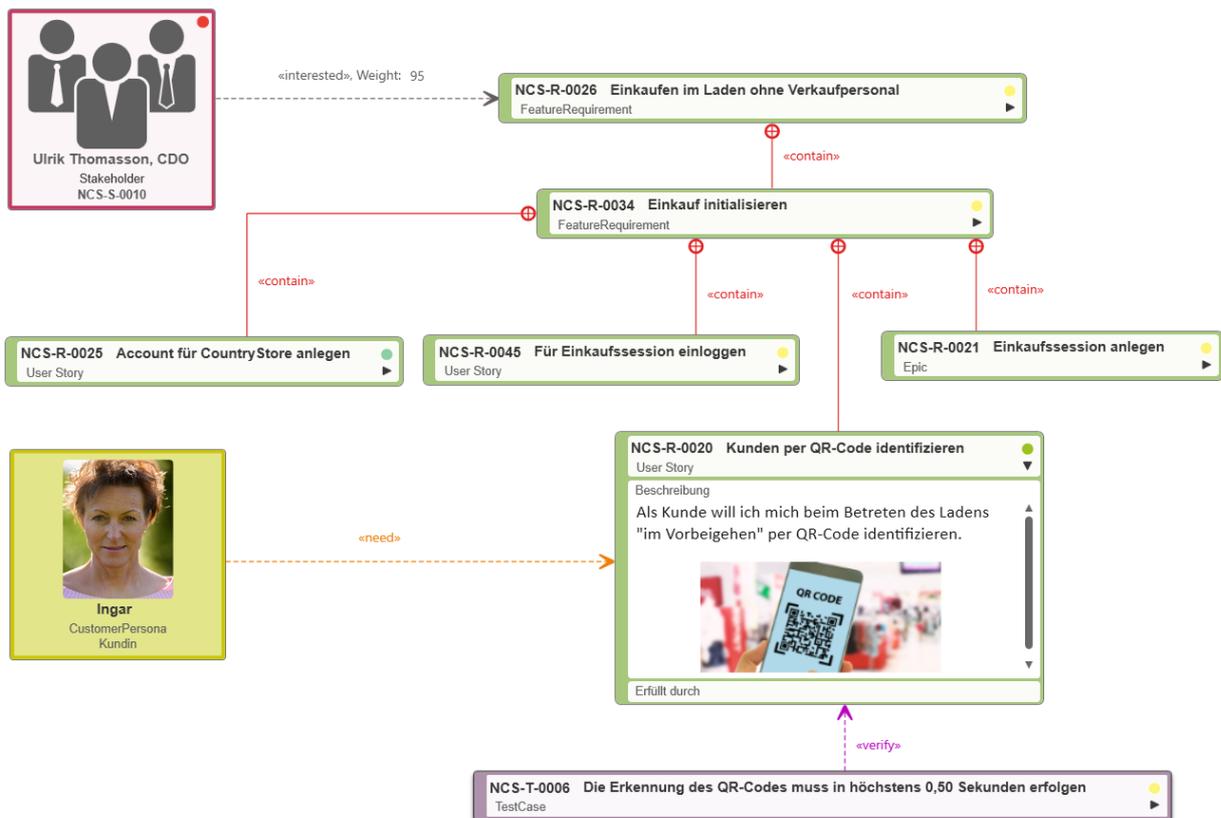


Abbildung 3: Ein Beispiel für das Anforderungsdiagramm mit einigen seiner Ausdrucksmittel

Der besondere Nutzen dieser Diagramme: Mit jeder darin dargestellten Beziehung von Anforderungen untereinander oder zu Stakeholdern, Testfällen, Lösungskomponenten etc. legen Sie auswertbare und navigierbare „Spuren“ für die Traceability.

Anforderungen zu modellieren ist eine von mehreren Optionen. Hier sind weitere:

- Sie können Anforderungen – wie alle anderen Ergebnisse des Requirements Engineering – per Bildschirmformular erfassen und bearbeiten. Auch die Beziehungen zu anderen Elementen können Sie im Formular herstellen. Wenn Sie Diagramme erstellt und darin Beziehungen zwischen Elementen abgebildet haben, dann finden Sie diese Beziehungen automatisch in den Formularen wieder. Alternativ zu den vorgegebenen Formularen können Sie auch eigene verwenden.
- Sie können Anforderungen aus fremden Quellen übernehmen, z.B. aus MS Excel- und MS Word-Dateien sowie aus Dateien im Requirements Interchange Format (ReqIF). Außerdem können Sie Anforderungen direkt aus E-Mails ableiten.

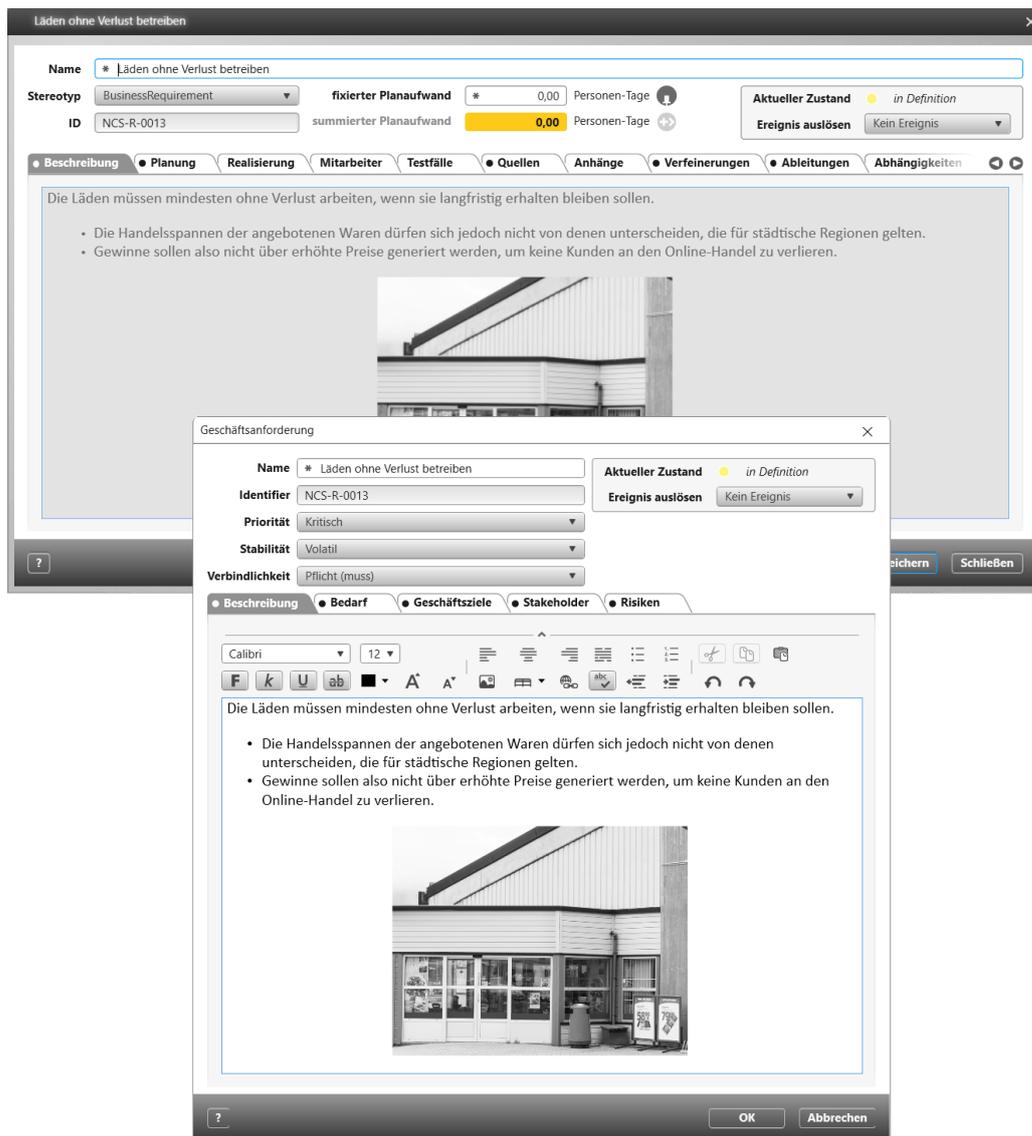


Abbildung 4: Ein Beispiel für das Standard-Anforderungsformular (oben) und dieselbe Anforderung in einem benutzereigenen Formular, hier speziell für Geschäftsanforderungen (im Vordergrund unten)

- Als Product Owner können Sie Anforderungen direkt im Product-Backlog anlegen, bis auf das Niveau von User Stories verfeinern und miteinander in Beziehung setzen.

Alle Anforderungen werden automatisch im **Product-Backlog** zusammengeführt – egal, ob Sie sie modelliert, direkt erfasst oder über eine der Schnittstellen importieren haben.

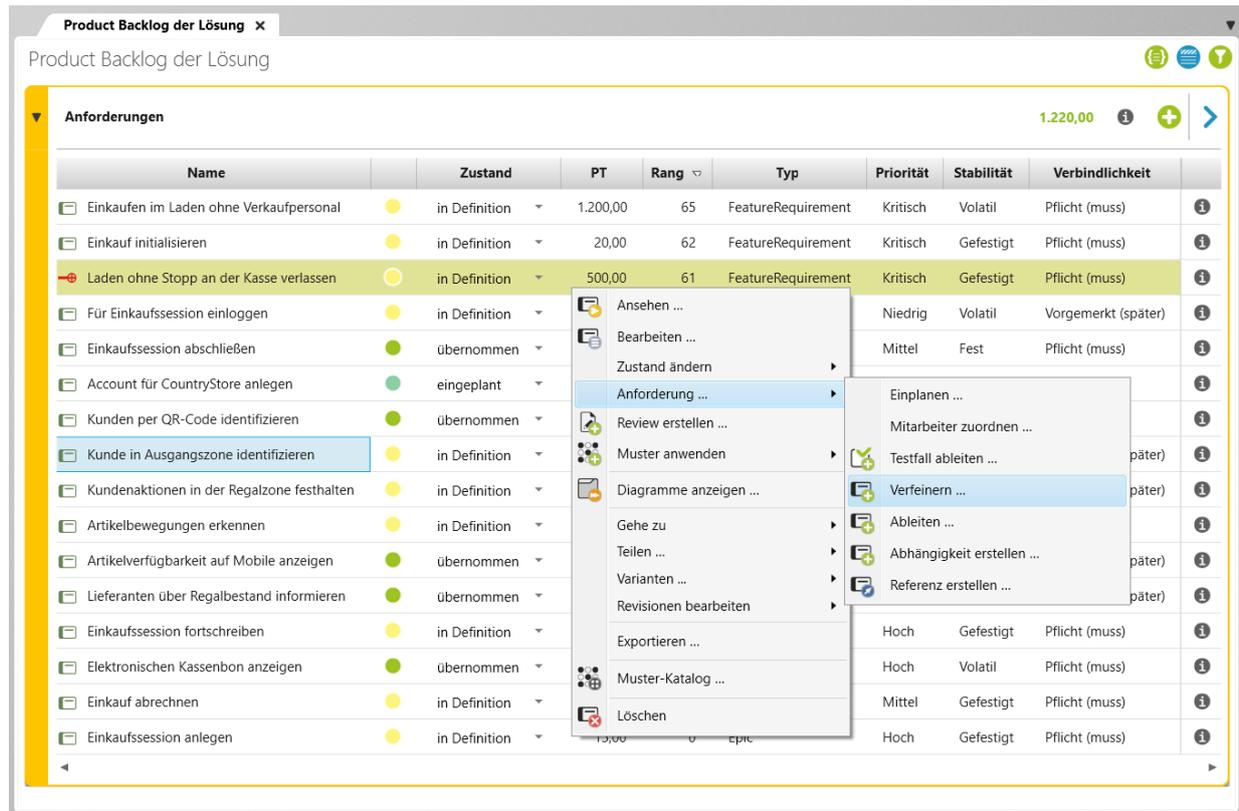


Abbildung 5: Anforderungen eingeben und bearbeiten, verfeinern, priorisieren, bewerten, schätzen und versionieren – dieses alles funktioniert auch direkt im Product-Backlog

Änderungsauswirkungen sichtbar machen

Große Projekte müssen mit tausenden von Anforderungen umgehen. Viele davon gehen auf Anwendungsfälle, Ziele und Stakeholder zurück. Zu den Anforderungen werden im Verlauf des Projekts Testfälle definiert und schließlich Komponenten entworfen und implementiert. Nehmen wir an, ein Stakeholder ändert seine Ziele. Dann brauchen Sie Mittel, um die davon betroffenen Anforderungen und alle damit in Beziehung stehenden Elemente zu identifizieren. In **objectiF RPM** finden Sie gleich mehrere Instrumente, die Ihnen helfen, Auswirkungen von Änderungen zu erkennen.

- Sie können direkt aus Formularen oder Diagrammen heraus zu abhängigen Elementen navigieren, also z.B. vom Stakeholder zu seinen Zielen, von den Zielen zu den daraus abgeleiteten Anforderungen, weiter zu den zugehörigen Testfällen sowie den entworfenen Komponenten und ggf. weiter bis zum Code.
- Sie können Elemente und ihre Abhängigkeiten in Dokumente generieren.
- Oder Sie verschaffen sich einen Überblick über die Zusammenhänge von Arbeitsergebnissen mit Echtzeitabfragen, die Sie individuell konfigurieren und filtern können.

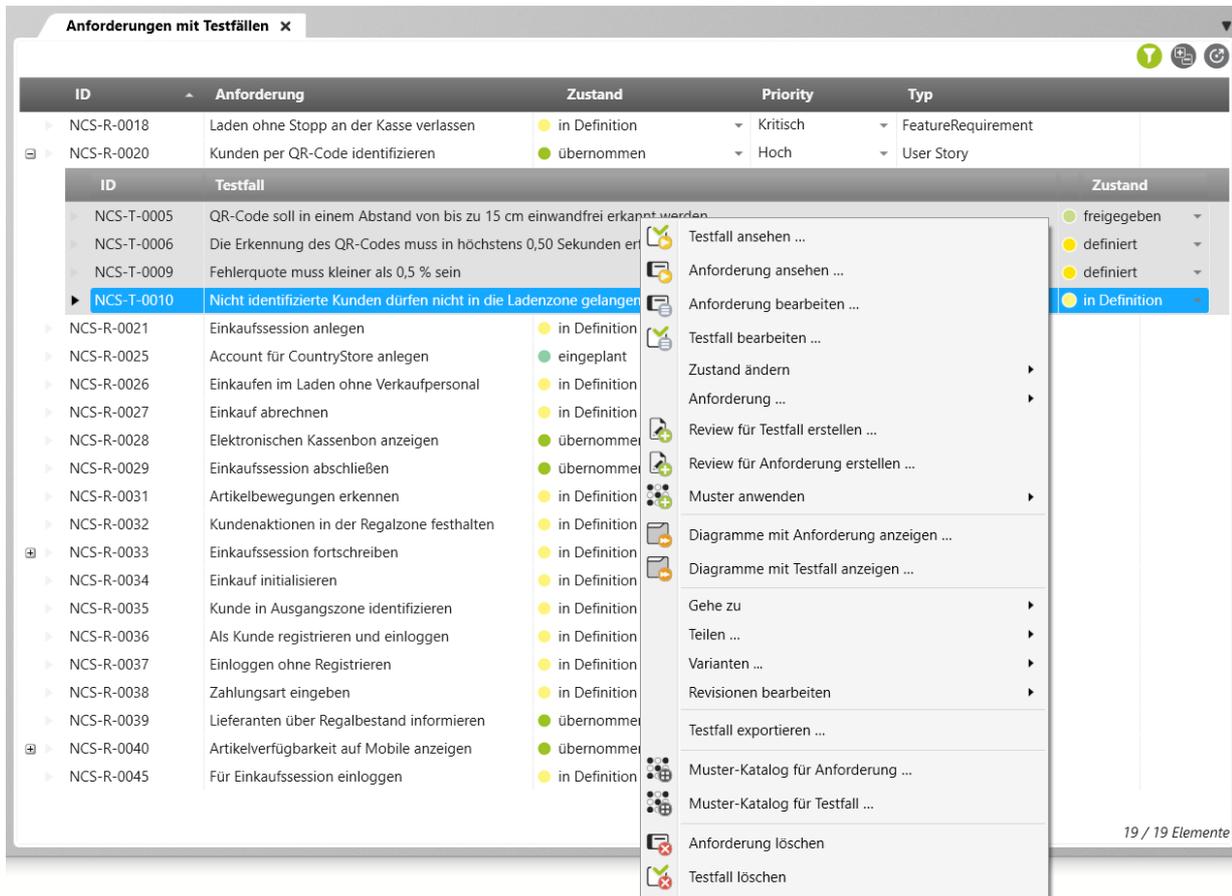


Abbildung 6: Die Echtzeit-Auswertung „Anforderungen mit zugehörigen Testfällen“: Auf jedem Listeneintrag sind Kontextmenüs für die weitere Bearbeitung und die Navigation verfügbar

Anforderungen prüfen

Sind die Anforderungen korrekt und vollständig beschrieben? Haben Sie die Stakeholder richtig verstanden? Wenn erst bei der Realisierung von Anforderungen auffällt, dass sie unklar formuliert sind, kann das teuer werden.

Deshalb ermöglicht **objectiF RPM** asynchrone **Reviews** in verteilten Teams von Reviewern. Jeder Reviewer kann Kommentare hinterlassen und Anforderungen akzeptieren oder ablehnen. Den Review-Workflow können Sie an spezielle Gegebenheiten in Ihrer Organisation ganz leicht anpassen.

Qualität bleibt mit **objectiF RPM** also nicht dem Zufall überlassen. Das gilt nicht nur für Anforderungen. Auch andere Ergebnisse wie Stakeholder-Ziele, Anwendungsfälle, Testfälle oder Dokumente können Sie auf die gleiche Weise im Team prüfen.

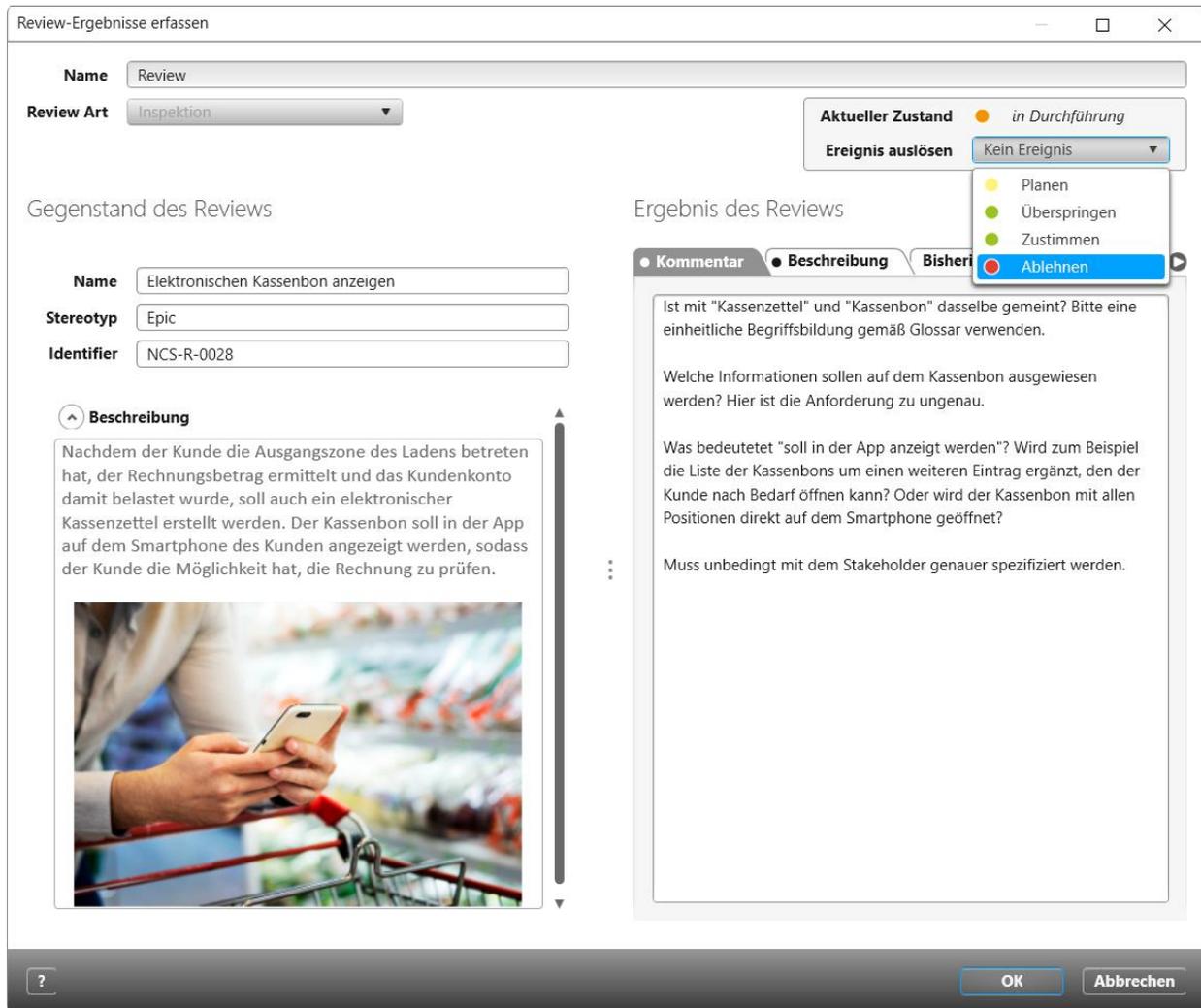


Abbildung 7: So gibt ein Reviewer seinen Kommentar zu einer Anforderung ein und löst ein entsprechendes Ereignis aus. Über diese Ereignisse und die daraus resultierenden Zustände wird der Review-Workflow gesteuert.

Lasten- oder Pflichtenheft schnell erstellen

Die Dokumentation von Anforderungen als Lasten- oder Pflichtenheft kostet in der Regel viel Zeit. Die Aktualität der Dokumente sicherzustellen, ist eine zusätzliche Herausforderung. Aber nicht bei der Arbeit mit **objectiF RPM**: Sie können **MS Word-Dokumente** oder PDF-Dateien mit nur wenigen Klicks aus Ihren Arbeitsergebnissen erzeugen. Auf der Basis von Dokumentvorlagen werden die gewünschten Anforderungen, beliebige Ergebnisse, Diagramme und Auswertungen in Ihre Dokumente generiert. Anforderungen haben sich geändert? Dann generieren Sie einfach einen neuen Stand Ihrer Dokumente. Ihre ergänzend eingegebenen Texte ändern sich dabei natürlich nicht.

Sie haben keine Zeit, an einer Dokumentvorlage für Ihr Lastenheft zu feilen? Dann verwenden Sie am besten die MS Word-Vorlage für ein Lastenheft nach ISO/IEC/IEEE 29148:2011. Sie ist standardmäßig in **objectiF RPM** enthalten.

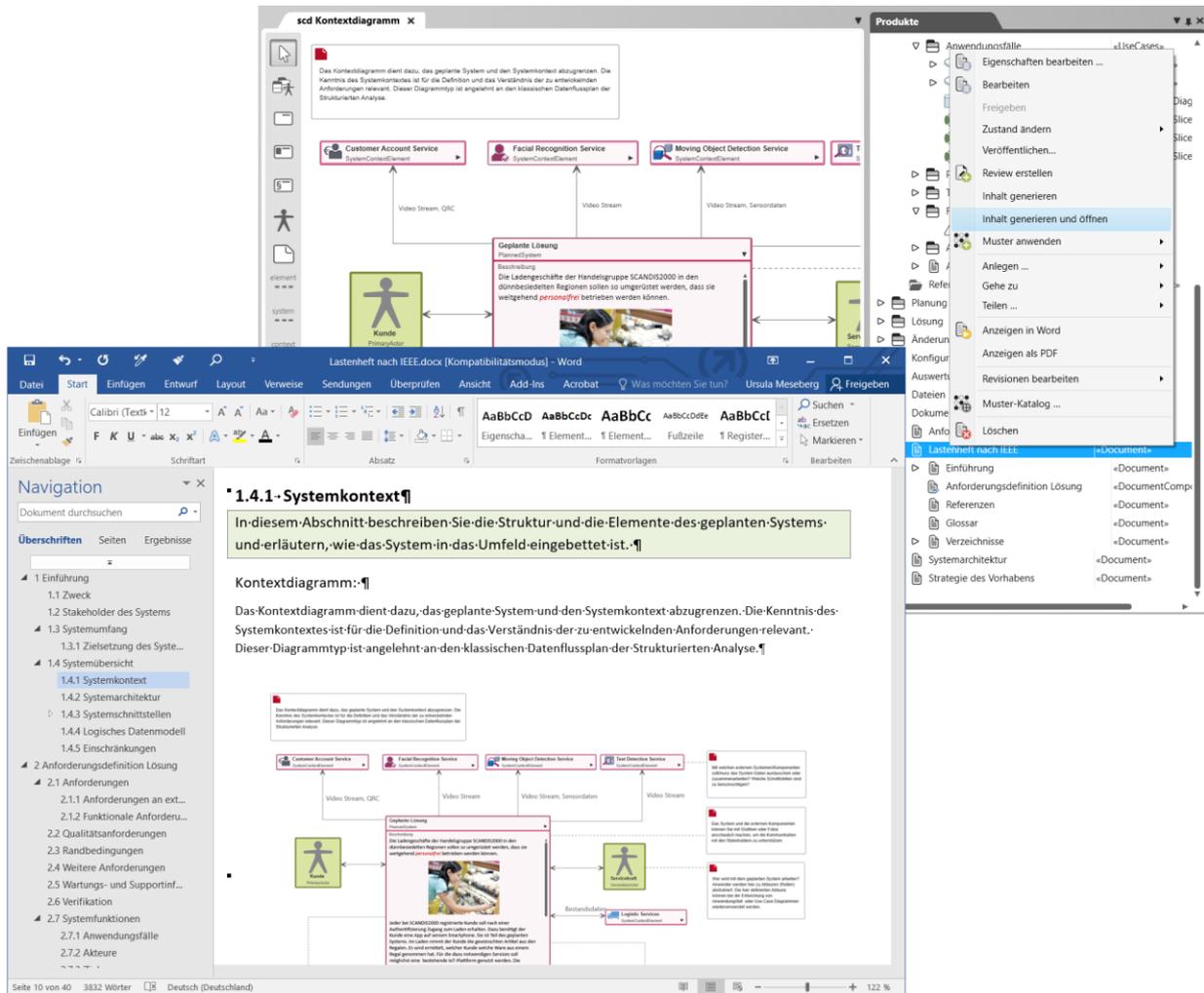


Abbildung 8: Lastenheft per Knopfdruck auf der Basis der Vorlage in objectiF RPM

Die unterstützten UML- und SysML-Diagramme

Diagramm	Anwendungszweck
 Aktivitätsdiagramm	Beschreibung und Spezifikation von Use Cases und Geschäftsprozessen
 Anforderungsdiagramm	Analyse der Anforderungen und ihrer Beziehungen zu Stakeholdern und Personas, Verfeinern von Anforderungen, Sicherstellen der Traceability
 Blockdiagramm	Gliederung der Systemarchitektur (z.B. in Hardware/Software Komponenten)
 Klassendiagramm	Fachliche und technische Datenmodellierung
 Packagediagramm	Spezifikation der Komponentengliederung des Systems
 Systemkontextdiagramm	Abgrenzen des Systemkontextes, Definition des Scope zur Vermeidung von Scope Creep
 Use Case Diagramm	Gestaltung der Anwender/Systeminteraktion
 Zieldiagramm	Analyse der Stakeholder und ihrer Ziele

Weitere Funktionen für das Requirements Engineering und Kollaboration:

- ✓ Versionsmanagement für Artefakte, Diagramme und Dokumente
- ✓ Variantenmanagement
- ✓ Konfigurationsmanagement mit Baselines
- ✓ Teilen von Diagrammen und einzelnen Ergebnissen per E-Mail mit Hyperlinks
- ✓ E-Mail-Benachrichtigung bei Zustandsänderungen von Ergebnissen
- ✓ Import und Export von JIRA-Issues
- ✓ Erstellung eines Projekt-Glossars

Design & Implementierung: Herausforderungen und Lösungen

Eine zentrale Herausforderung bei Design und Implementierung besteht darin, den Weg von den Anforderungen bis zum Code sichtbar zu machen, sodass die Architekturentscheidungen nachvollziehbar bleiben. Nur so können Sie die Konsequenzen von neuen und veränderten Anforderungen auf die Implementierung sicher einschätzen, Änderungsbedarf lokalisieren und die Anforderungen schnell umsetzen.

Wenn Sie speziell serviceorientierte Systeme entwickeln, sind Sie mit einer weiteren Herausforderung konfrontiert: der Komplexität serviceorientierter Architekturen.

In beiden Fällen setzt **objectiF RPM** auf die ordnende Wirkung von Modellen.

Systementwurf aus Anforderungen entwickeln

Um Architekturen sichtbar und Entwurfsentscheidungen nachvollziehbar zu machen, bietet **objectiF RPM** für Komponenten, Daten und das Verhalten eines Systems Modelle an. Die zentrale Frage nach dem Zusammenhang zwischen Anforderungen, Systemkomponenten und Code können Sie mit den Architekturmodellen jederzeit beantworten.

Der Systementwurf ist der Schlüssel zu langlebigen, erweiterbaren und gut wartbaren Systemen. Damit ist er auch in agilen Projekten unverzichtbar. Die Systemarchitektur zu entwerfen, bedeutet jedoch nicht zwangsläufig, grafische Designmodelle zu erstellen. Sie können Systemelemente auch einfach per Formular anlegen und mit Anforderungen in Beziehung setzen.

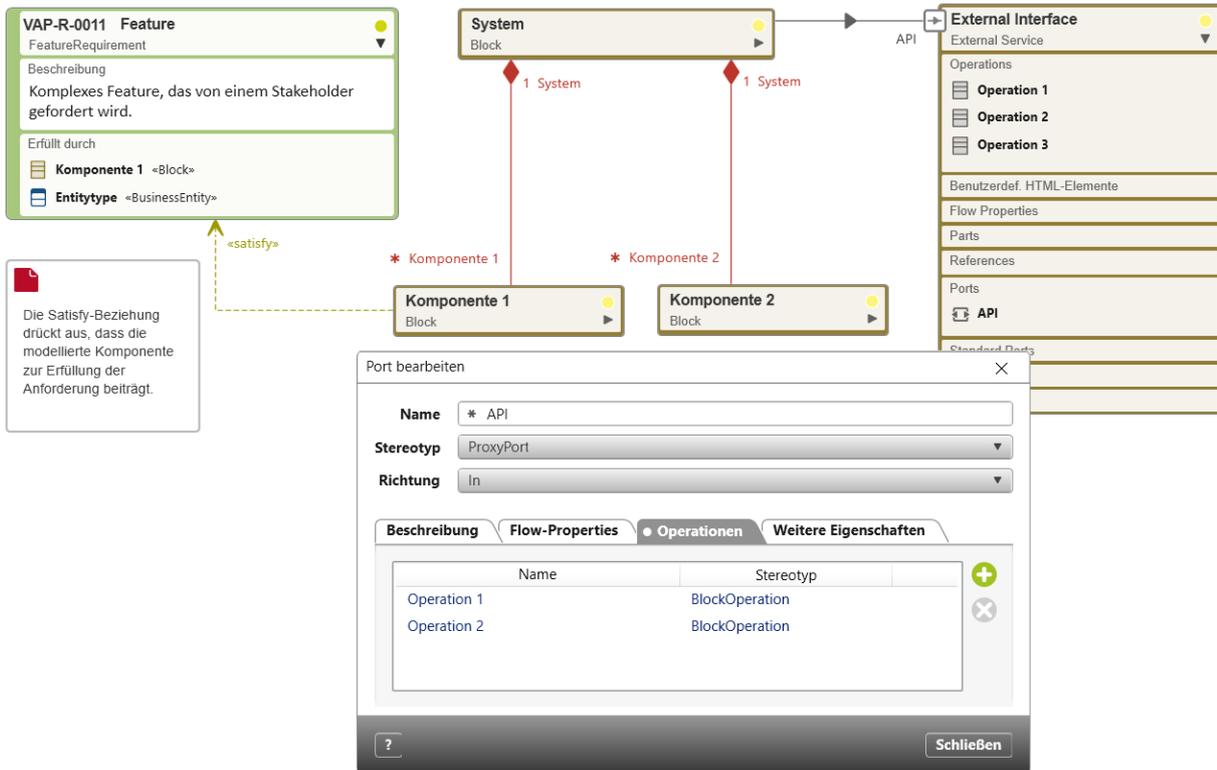


Abbildung 9: Systementwurf mit einem Blockdiagramm. Mit diesem Diagrammtyp können zum Beispiel die Komponenten, ihre Struktur und die von ihnen verwendeten APIs sichtbar gemacht werden. Außerdem können die Beziehungen zwischen Anforderungen und Komponenten grafisch veranschaulicht werden.

Identifizier	Anforderung	Stereotyp	Zustand									
▶ NCS-R-0037	Einloggen ohne Registrieren	UseCaseSlice	● in Definition									
▶ NCS-R-0038	Zahlungsart eingeben	UseCaseSlice	● in Definition									
▶ NCS-R-0018	Laden ohne Stopp an der Kasse verlassen	FeatureRequirement	● in Definition									
▶ NCS-R-0026	Einkaufen im Laden ohne Verkaufspersonal	FeatureRequirement	● in Definition									
▣ ▶ NCS-R-0031	Artikelbewegungen erkennen	FeatureRequirement	● in Definition									
<table border="1"> <thead> <tr> <th>Element</th> <th>Stereotyp</th> <th>Zustand</th> </tr> </thead> <tbody> <tr> <td>ArticleLabel</td> <td>Class</td> <td>● definiert</td> </tr> <tr> <td>RecognitionService</td> <td>Web Service API</td> <td>● definiert</td> </tr> </tbody> </table>				Element	Stereotyp	Zustand	ArticleLabel	Class	● definiert	RecognitionService	Web Service API	● definiert
Element	Stereotyp	Zustand										
ArticleLabel	Class	● definiert										
RecognitionService	Web Service API	● definiert										
⊕ ▶ NCS-R-0032	Kundenaktionen in der Regalzone festhalten	FeatureRequirement	● in Definition									
▶ NCS-R-0034	Einkauf initialisieren	FeatureRequirement	● in Definition									
▶ NCS-R-0035	Kunde in Ausgangszone identifizieren	FeatureRequirement	● in Definition									
⊕ ▶ NCS-R-0020	Kunden per QR-Code identifizieren	User Story	● eingeplant									
⊕ ▶ NCS-R-0025	Account für CountryStore anlegen	User Story	● eingeplant									
▶ NCS-R-0029	Einkaufssession abschließen	User Story	● in Realisierung									
▶ NCS-R-0045	Für Einkaufssession einloggen	User Story	● in Definition									

19 / 19 Elemente

Abbildung 10: Die Beziehungen zwischen Anforderungen und Architekturelementen können auch mit Hilfe von Auswertungen nachvollzogen werden

Entwurfsmuster zum Anlegen wiederkehrender Strukturen vereinfachen und beschleunigen die Entwurfsarbeit zusätzlich.

Ob per Muster, formularbasiert oder mit Diagrammen – in jedem Fall stellen Sie die **Traceability** von Anforderungen zum Systementwurf sicher.

Model-based Development „light“: Designmodelle zur Implementierung nutzen

Serviceorientierte Architekturen stehen heute hoch im Kurs. Sie bestehen aus kleinen entkoppelten Diensten, die über definierte APIs kommunizieren und von unabhängigen Teams entwickelt werden können. Sie verkürzen die Bereitstellungszyklen und vereinfachen die Skalierung von Anwendungen. Damit passen sie gut zu agilen Organisationen. Der Preis dafür ist eine hohe Komplexität. Sie resultiert aus der großen Zahl an Services und ihren Interaktionen.

Wie orientiert man sich unter Dutzenden oder Hunderten von Service-Komponenten? Wie verhindert man, dass das Rad von jedem Team neu erfunden wird? Wie vermeidet man in heterogenen Umgebungen einen Mangel an Entwurfsstandards?

Die Antwort, die *objectiF RPM* darauf gibt, lautet: mit einer leichtgewichtigen Form der **modellbasierten Entwicklung (Model-based Development)**, die speziell auf Anwendungen aus RESTful-Services und Microservices zugeschnitten ist. Konkret bedeutet das: *objectiF RPM* unterstützt Sie mit:

- Mustern für die einfache Gliederung der Fachlichkeit in Domänen,
- Backlogs pro Domäne und Team für die unabhängige Entwicklung von Services,
- Funktionen für den Entwurf von Services und Operationen sowie für die API-Definition,
- Funktionen für die Code-Generierung.

Damit genau das gebaut wird, was Sie entworfen haben, können Sie mit *objectiF RPM* aus den Designmodellen **Quellcode** erzeugen und mit einem Klick aus dem Kontext des Entwurfs in den Code-Editor wechseln.

objectiF RPM ist standardmäßig mit JavaScript, HTML Custom Elements, Node.js und Visual Studio Code zu einer durchgängigen Entwicklungsumgebung integriert. Damit wird insbesondere das Konzept der ganzheitlichen Entwicklung von client-seitigen User-Interfaces und server-seitiger Verarbeitungslogik unterstützt. Die Versionsverwaltung für den Quellcode übernimmt dabei ebenfalls *objectiF RPM*.

Sie arbeiten mit anderen Technologien? Kein Problem: Die Integration von *objectiF RPM* mit den genannten Technologiekomponenten wurde durch **Erweiterungsfunktionen** realisiert. Der skriptbasierte Quellcode der Erweiterungsfunktionen steht Ihnen als Vorlage für die Integration von *objectiF RPM* mit Ihrem spezifischen Technologie-Stack zur Verfügung.

Die Integration lohnt sich. Denn die leichtgewichtige Form der modellbasierten Entwicklung mit **Entwurfsmustern** und **automatisierten Implementierungsschritten** sichert die Einhaltung des gewählten Architekturstils über alle Releases eines Systems hinweg.

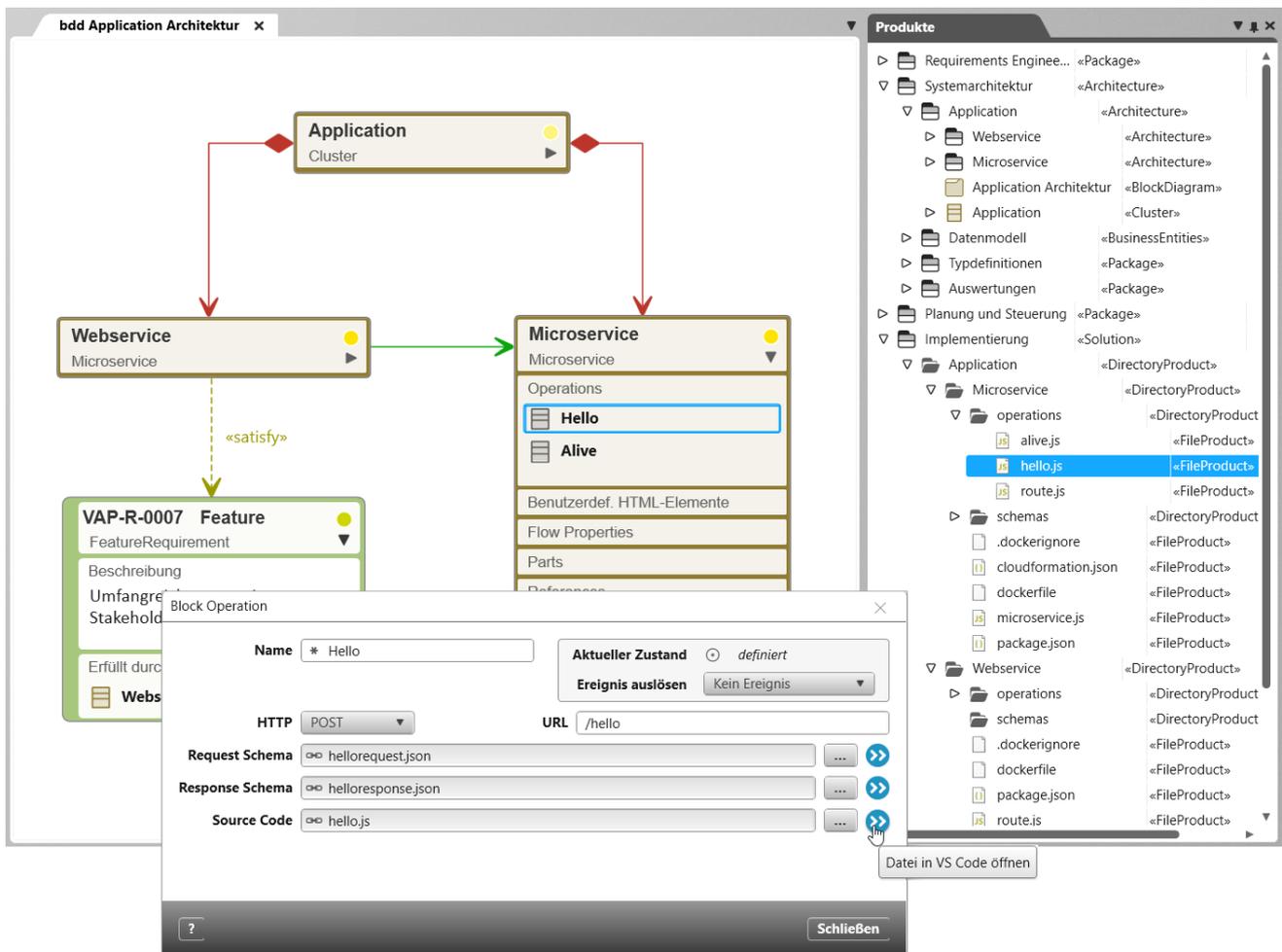


Abbildung 11: Mit einem Klick von einer Service-Operation in die generierte Code-Datei wechseln. Verzeichnisstruktur und Code.

Deployment: Systeme in der Cloud bereitstellen

Um hohe Agilität zu erreichen, entscheiden sich viele Unternehmen dafür, eine DevOps-Kultur zu etablieren. D.h., sie führen Entwicklungskompetenz und die Fähigkeit zur kontinuierlichen Bereitstellung von Lösungen in ihren Teams zusammen. Serviceorientierte Architekturen, die die Möglichkeit bieten, neue und verbesserte Services jederzeit – zum Beispiel in einer Cloud – für den Betrieb bereitzustellen, sind dafür besonders geeignet.

objectiF RPM bringt Deployment-Funktionen für einen Technologie-Stack aus JavaScript, Node.js, Visual Studio Code und Amazon Elastic Cloud mit. Die Deployment-Funktionen sind in Form von Erweiterungsfunktionen zu **objectiF RPM** implementiert. Der Quellcode dieser Funktionen steht Ihnen auch hier wieder als Vorlage zur Verfügung, wenn Sie **objectiF RPM** an Ihren spezifischen Technologie-Stack anpassen wollen.

Deployment mit **objectiF RPM** hilft, Innovation schneller verfügbar zu machen und Systeme leichter zu skalieren. Die Anpassung von **objectiF RPM** zahlt sich für Sie hier also garantiert aus.

Traceability bis zum Code sicherstellen

Im Projektalltag stellt die Traceability von den Anforderungen bis zum Code eine Herausforderung dar. Sie ist von zentraler Bedeutung, wenn es darum geht, herauszufinden, warum eine Funktion so entwickelt wurde, wie sie aktuell vorliegt, was aufgrund einer User Story am Code geändert wurde, vom wem und wann?

Mit **objectiF RPM** stellen Sie die Nachvollziehbarkeit von den Stakeholdern und ihren Zielen über die Anforderungen mit ihren Testfällen bis hin zu den Komponenten und den zugehörigen Code-Dateien sicher. Sie können von jeder Version einer Code-Datei zu dem Stand der Anforderung navigieren, zu dem die Version angelegt worden ist. Auch wenn sich die Anforderung mittlerweile geändert hat, lässt sich so nachvollziehen, wie sie und auch der zugehörige Code zu einem früheren Zeitpunkt ausgesehen haben.

Was Sie dafür tun müssen? Nur zwei Dinge: Überlassen Sie **objectiF RPM** einfach die Quellcode-Verwaltung. Halten Sie bei der Rückgabe einer neuen oder geänderten Code-Datei an **objectiF RPM** die Anforderungen bzw. User Stories fest, die der Grund für das Erstellen oder Ändern der Datei waren.

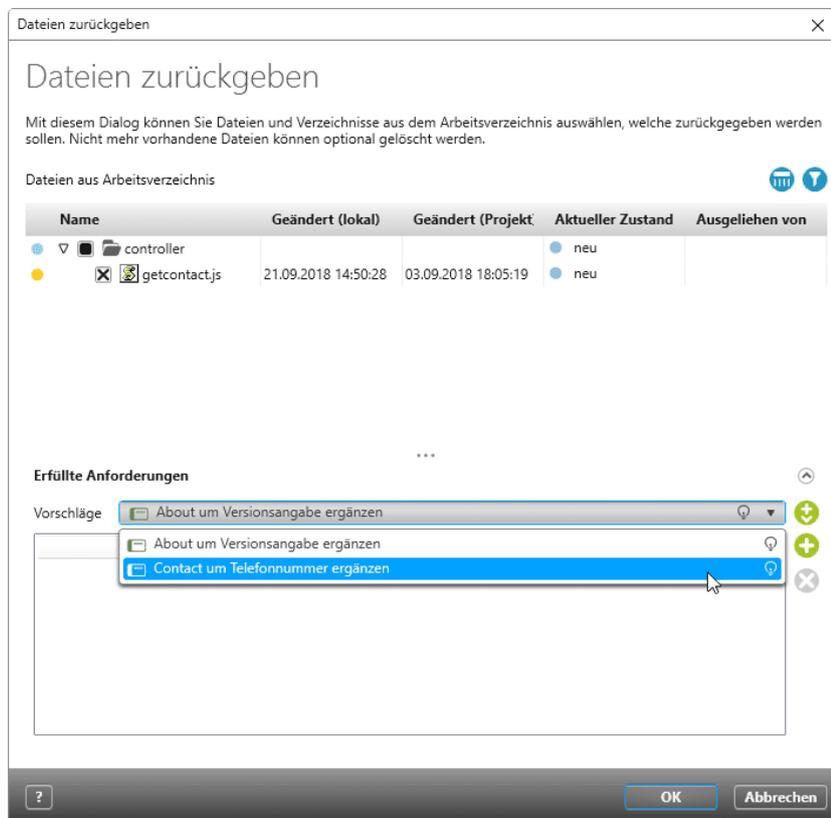


Abbildung 12: So stellen Sie die Traceability zwischen Code-Dateien und Anforderungen her:
Die Datei `getcontact.js` erfüllt die User Story „Contact um Telefonnummer ergänzen“.

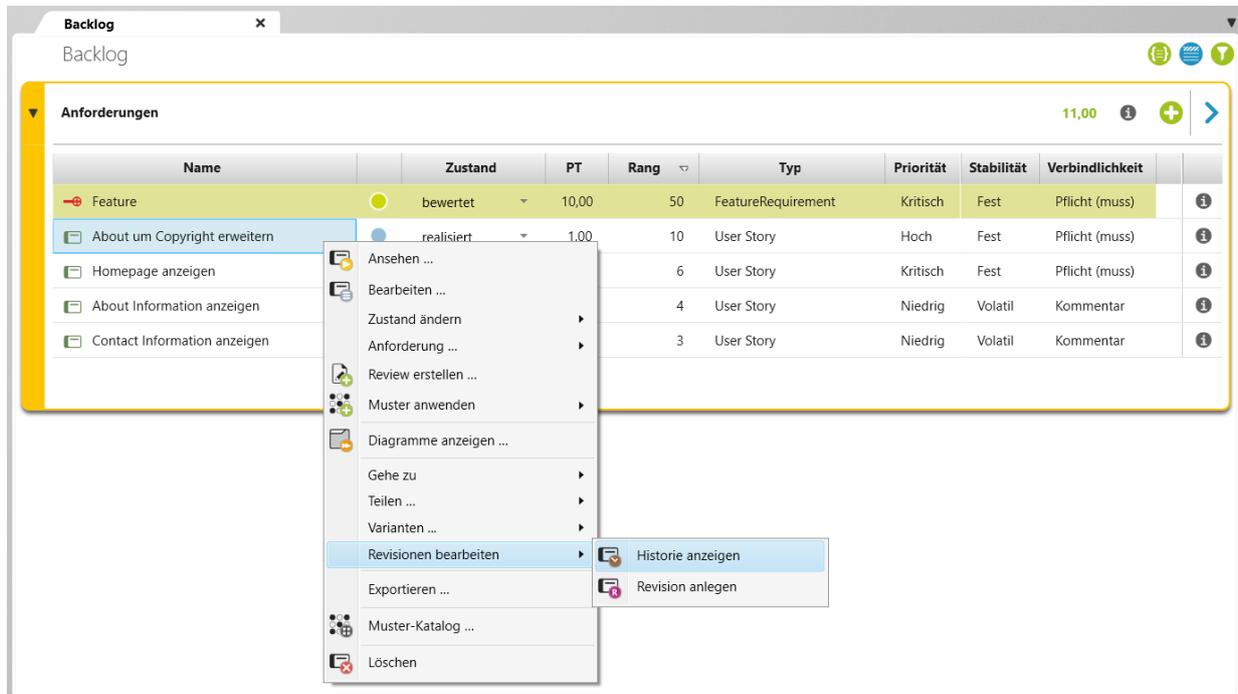


Abbildung 13: Die Historie einer Anforderung und speziell einer User Story gibt Auskunft darüber, wie die früheren Versionen im Detail aussahen...

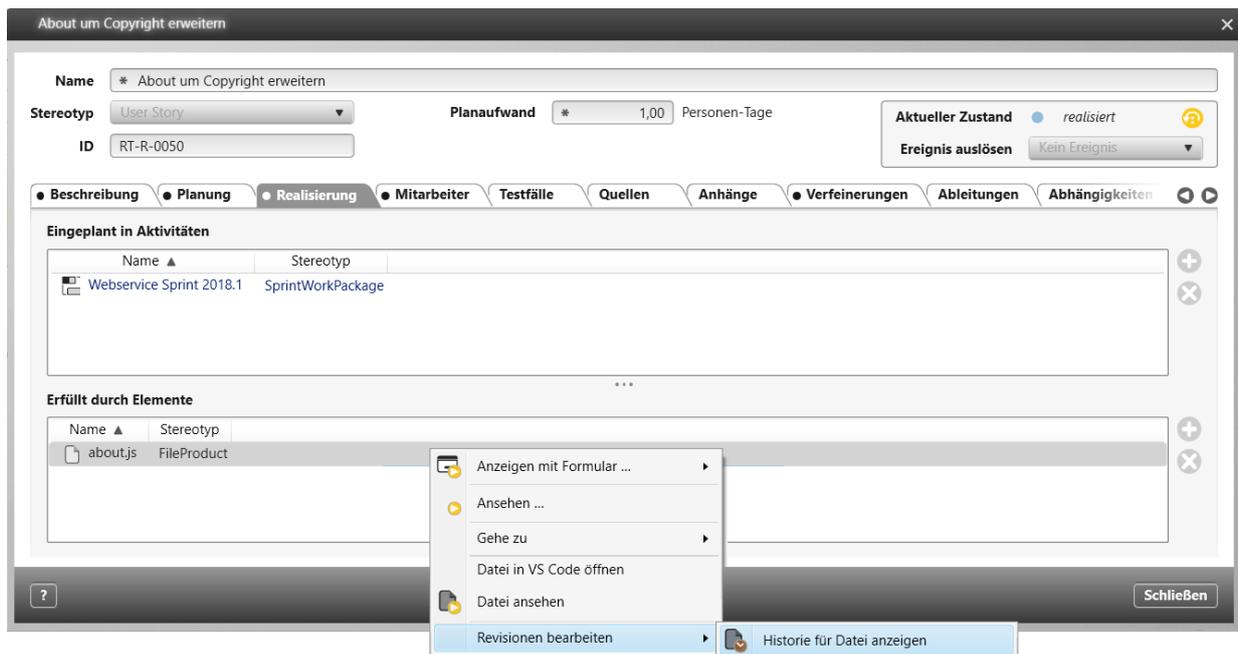
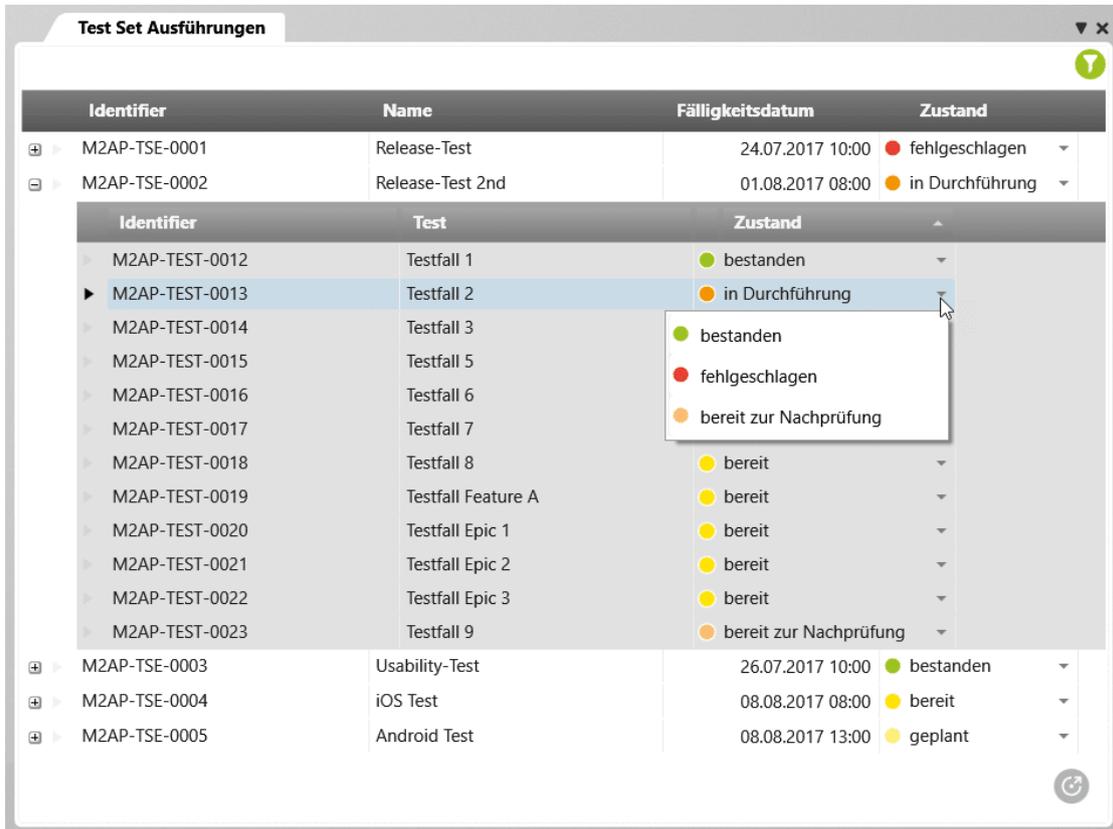


Abbildung 14: ... und welche Quellcode-Dateien damals dazu gehörten und welchen Versionsstand sie hatten. Der Quellcode-Vergleich zeigt Ihnen dann, was sich wann geändert hat.

Testmanagement

Die Realisierung der Anforderungen muss im Verlauf eines Projekts geprüft werden, z. B. durch Systemtests. Deshalb sollte bereits beim Requirements Engineering damit begonnen werden, Testfälle zu erstellen und mit Anforderungen zu verbinden.

Darauf basierend können Sie später Testläufe planen und Tests durchführen. *objectiF RPM* hilft Ihnen dabei mit einem vordefinierten, konfigurierbaren Testmanagement-Workflow.



Identifier	Name	Fälligkeitsdatum	Zustand
M2AP-TSE-0001	Release-Test	24.07.2017 10:00	fehlgeschlagen
M2AP-TSE-0002	Release-Test 2nd	01.08.2017 08:00	in Durchführung
M2AP-TEST-0012	Testfall 1		bestanden
M2AP-TEST-0013	Testfall 2		in Durchführung
M2AP-TEST-0014	Testfall 3		
M2AP-TEST-0015	Testfall 5		
M2AP-TEST-0016	Testfall 6		
M2AP-TEST-0017	Testfall 7		
M2AP-TEST-0018	Testfall 8		bereit
M2AP-TEST-0019	Testfall Feature A		bereit
M2AP-TEST-0020	Testfall Epic 1		bereit
M2AP-TEST-0021	Testfall Epic 2		bereit
M2AP-TEST-0022	Testfall Epic 3		bereit
M2AP-TEST-0023	Testfall 9		bereit zur Nachprüfung
M2AP-TSE-0003	Usability-Test	26.07.2017 10:00	bestanden
M2AP-TSE-0004	iOS Test	08.08.2017 08:00	bereit
M2AP-TSE-0005	Android Test	08.08.2017 13:00	geplant

Abbildung 15: Beispiel für die Übersicht von aktuellen Tests mit ihren Zuständen

Zu den Funktionen, die Ihnen *objectiF RPM* für das Testmanagement anbietet, gehören außerdem:

- das Ableiten von Bugs,
- das Erstellen von Reports über den aktuellen Testfortschritt,
- das Wiederholen von Testdurchläufen für Regressionstests.

Video-Links zu Design & Implementierung

- ▶ [Requirements Traceability](#)
- ▶ [Model-Based Development von Services](#)
- ▶ [Deployment von Microservices](#)

Projektmanagement: Herausforderungen und Lösungen

Aus Managementsicht geht es in allen Projekten um **Kosten**, **Termine** und **Ressourcen**. Deshalb finden Sie in **objectiF RPM** zahlreiche Funktionen für die Kosten-, Termin- und Ressourcenplanung. Aber Projekt ist nicht gleich Projekt. In vielen Branchen findet man neben agilen Projekten nach wie vor klassische Vorhaben, die mit Hilfe von Aktivitäten, Arbeitspaketen und Meilensteinen geplant werden. Manche Unternehmen kombinieren agiles und klassisches zu einem hybriden Vorgehen. In regulierten Branchen spielt dabei die Orientierung an Prozessstandards eine zentrale Rolle.

In Organisationen, die sich der digitalen Transformation und damit der Entwicklung neuer digitaler Geschäftsmodelle verschrieben haben, leitet sich das hybride Vorgehen oft aus den Erkenntnissen einer vorangegangenen strategischen Analyse ab. Für diese Organisationen steht die Nachvollziehbarkeit der Wertschöpfungskette **Bedarf – Lösung – Nutzen** im Vordergrund.

Ganz andere Herausforderungen stellen sich in agilen Projekten zur Entwicklung service-orientierter Anwendungen. Hier geht es um die geeignete Strukturierung der Domäne und die Orientierung innerhalb einer großen Zahl an Services.

objectiF RPM bietet deshalb zum Anlegen und Strukturieren von Projekten mehrere **Projektvorlagen** an, die speziell auf diese Anwendungsgebiete zugeschnitten sind. Sie erzeugen alles, was Sie brauchen, um ein Projekt sofort zu starten. Darüber hinaus können Sie für spezifische Projekttypen eigene Projektvorlagen erstellen. Projektmanagement mit **objectiF RPM** bedeutet also:

- Projektstart mit minimalem Vorbereitungsaufwand
- Einhalten von Prozessstandards
- Vergleichbarkeit Ihrer Projekte.

Projekte anlegen, strukturieren, planen und skalieren

Das leistet eine Projektvorlage: Sie erzeugt die Ablagestruktur für die Projektergebnisse, Backlogs, das User Story Board, einen initialen Projektplan samt Gantt-Darstellung, Dashboards, zahlreiche vorkonfigurierte Auswertungen, Dokumentvorlagen, initiale Dokumente und vieles mehr.

Das Einrichten der Projektumgebung kostet Sie somit nur wenige Klicks. Entsprechendes gilt für das Fortschreiben der Projektplanung: Hier sind es **Planungsmuster**, mit denen Sie den Projektplan um Aktivitäten und Work-Packages erweitern, Releases und Sprints anlegen, die zugehörigen Backlogs, Burn-Up Charts, **Cumulative Flow Diagrams**, Übersichten der **Mitarbeiterauslastung**, Instrumente der **Earned Value Analyse**, und die benötigten Ordnerstrukturen einrichten. Das alles mit wenigen Klicks.

Nach Bedarf können Sie Ihre Projekte auch mit eigenen Mustern strukturieren und planen, die Sie genau auf Ihre Prozesse anpassen können.

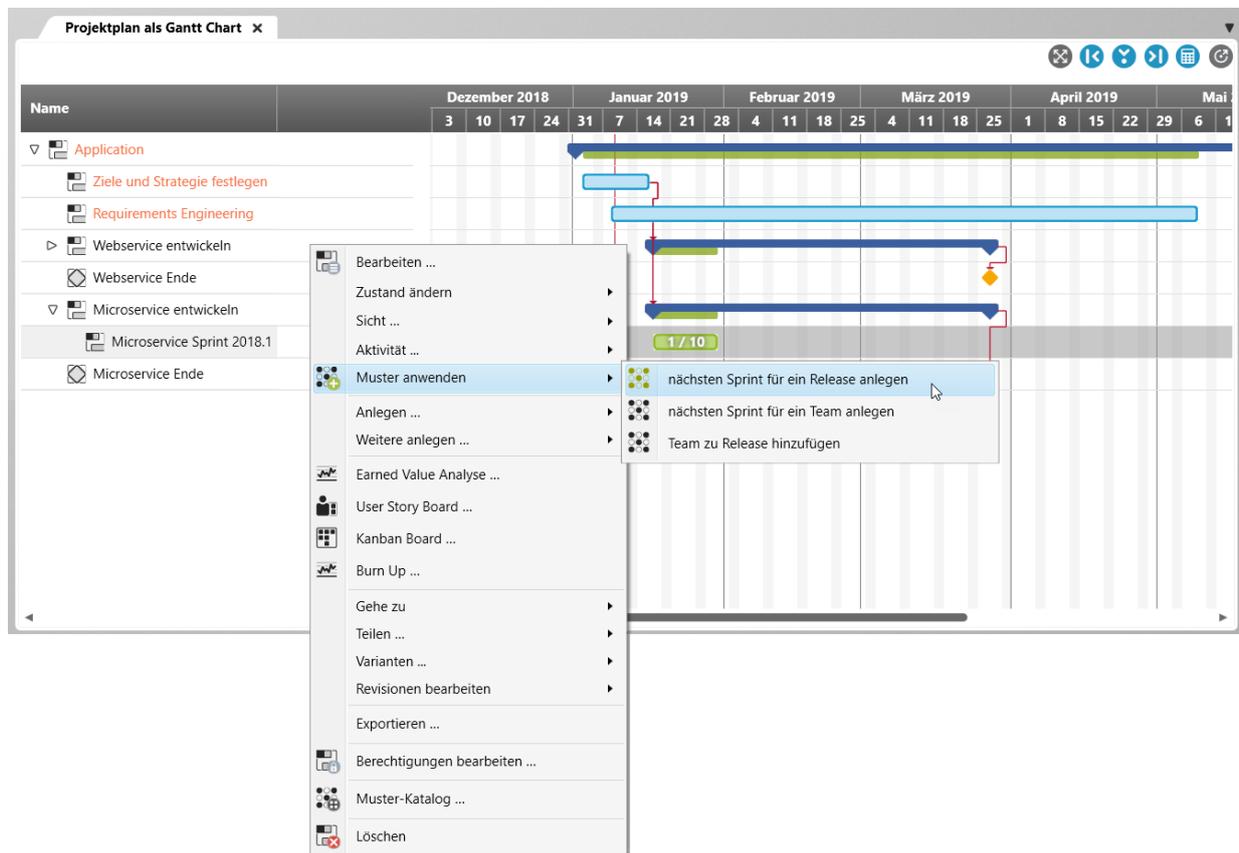


Abbildung 16: Das Anlegen eines neuen Sprints per Muster mit allem, was dazugehört, erfordert nur einen Klick

Und wie sieht es mit großen Projekten aus? Manche Unternehmen schicken zunächst nur ein Team an den Start und skalieren das Projekt dann schrittweise, indem sie nach und nach weitere Teams hinzunehmen. Die Herausforderung dabei: Jedes neue Team braucht eine eigene Arbeitsumgebung, in der es Anforderungen, User Stories, Testfälle etc. ablegen und versionieren kann. Es benötigt eigene Backlogs und User Story Boards bzw. Kanban Boards. Außerdem soll der Erfolg jedes Teams unabhängig von den anderen zum Beispiel in Burn-Up Charts gemessen werden.

objectiF RPM stellt ein spezielles **Muster** bereit, mit dem Sie ein Projekt durch wenige Klicks um neue Teams bzw. Ihre Projektplanung um neue Teamaktivitäten erweitern. Gleichzeitig richtet das Muster die Arbeitsumgebung für die Teams ein. Einfacher kann man Projekte nicht skalieren.

Aktivitätsorientiert oder agil planen

Sie haben die Wahl, ob Sie mit **objectiF RPM** ein ganzes Projekt oder Teile davon durch die Definition der Work-Breakdown-Structure aus Aktivitäten, Arbeitspaketen und Meilensteinen planen oder agil in Sprints vorgehen wollen.

- Für agile Release-, Team- und Sprintplanung bietet **objectiF RPM** einen vorbereiteten Workflow. Die Planungsschritte vom Product-Backlog über Release-, Team- und Sprint-Backlogs bis zum User Story Board sind jederzeit nachvollziehbar.
- Bei klassischer Projektplanung ordnen Sie einer Aktivität bzw. einem Arbeitspaket Anforderungen zu. Den Nachweis, welche Anforderung wann von wem im Kontext welcher Aktivität bearbeitet wird bzw. wurde, können Sie dann immer führen.

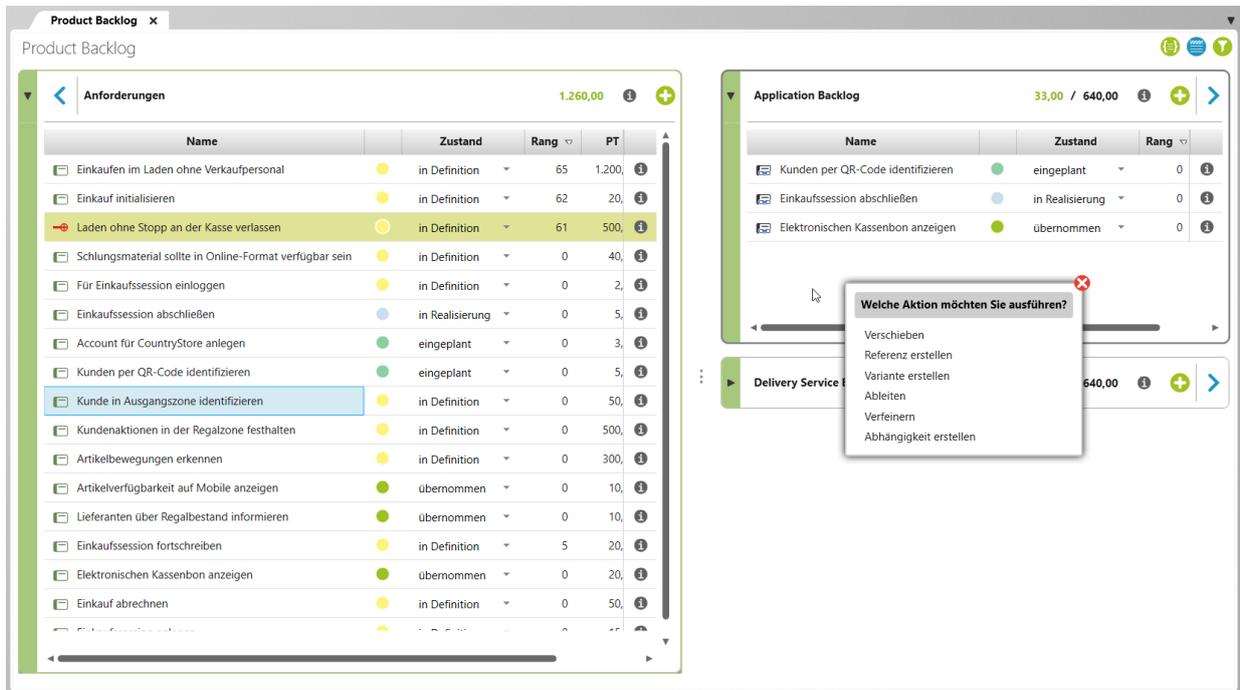


Abbildung 17: Anforderungen aus dem Product-Backlog in ein Release einplanen

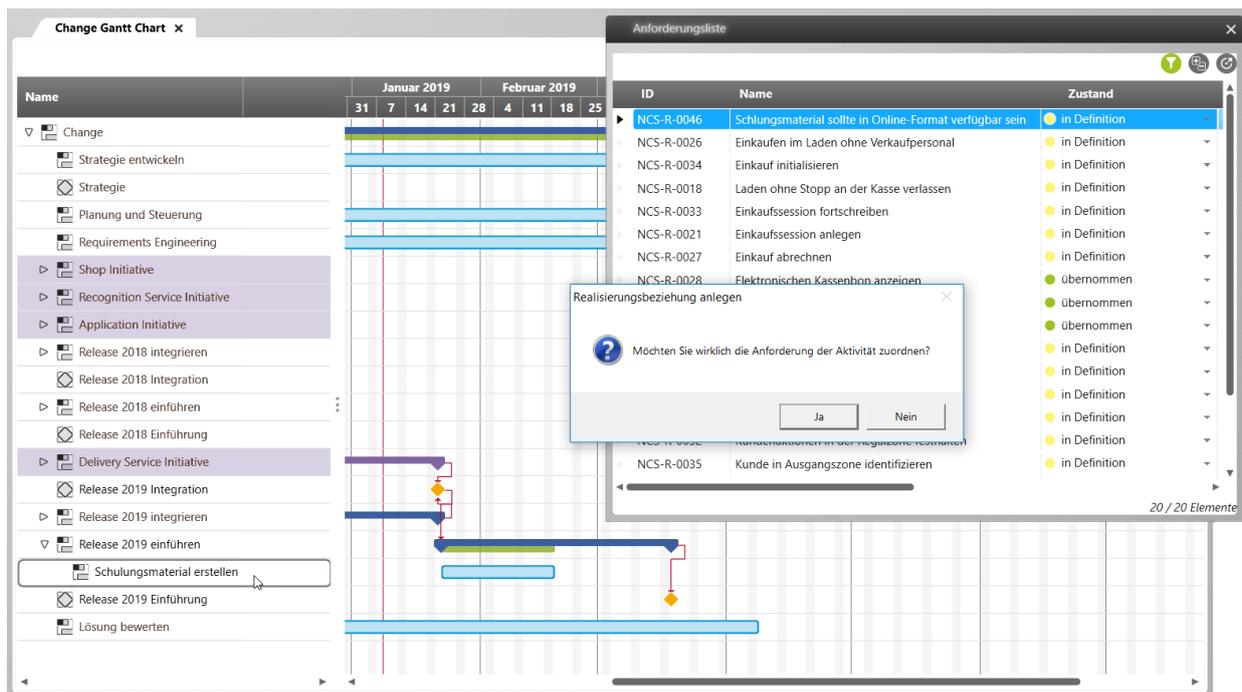


Abbildung 18: Anforderungen einer Projektaktivität zuordnen

Änderungen der Projektplanung nachvollziehen

Für die **Terminplanung** empfiehlt sich der Einsatz von Gantt-Diagrammen. Sie bilden die Projektstruktur mit Aktivitäten, Kontrollflüssen und Meilensteinen ab. Gantt-Diagramme sind nicht nur bei klassischer, sondern auch bei agiler Projektplanung äußerst hilfreich. Deshalb bringen die Projektvorlagen und die Planungsmuster für agile und hybride Projekte Gantt-Diagramme mit und verknüpfen sie mit den Backlogs.

Der Vorteil für Sie: Speziell in großen Projekten mit mehreren Teams sehen Sie auf einen Blick, wie weit die Entwicklung von Releases und Sprints vorangekommen ist und ob es in einem Release oder Sprint Kapazität für weitere Stories gibt.

Jedes Projekt erlebt immer wieder Veränderungen: Termine werden verschoben, Meilensteine werden gerissen, Mitarbeiter werden abgezogen usw. Um die Auswirkungen von Veränderungen zu überwachen, finden Sie passende Instrumente in **objectiF RPM**: Wenn sich z. B. ein Termin verschiebt und damit der Endtermin des Projekts gefährdet ist, sehen Sie dies im Projektplan am kritischen Pfad. Außerdem können Sie verschiedene Planungsstände – genauer: Versionen des Gantt-Diagramms – vergleichen. So erkennen Sie schnell, welche Aktivitäten und Meilensteine sich verschoben haben oder welche Arbeitspakete neu hinzugekommen sind.

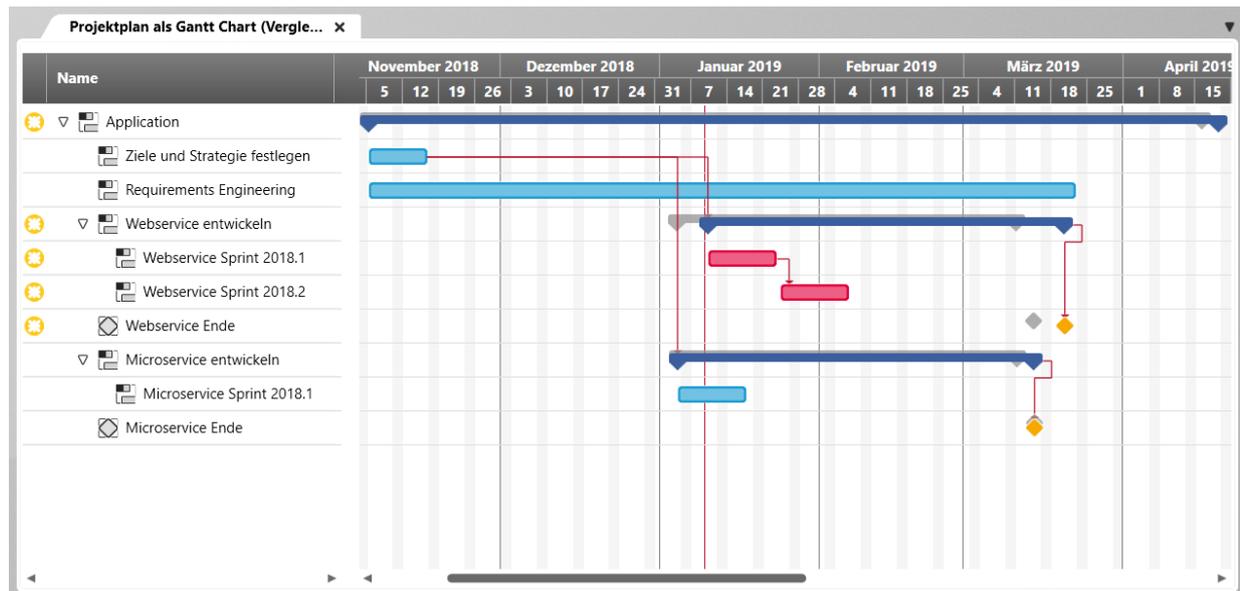


Abbildung 19: Vergleich zweier Versionen eines Gantt-Diagramms: Die rot dargestellten Sprints und die zugehörige Sammelaktivität haben sich verschoben. Grau hinterlegt erkennen Sie die ursprüngliche Planung der Sammelaktivität „Webservice entwickeln“

Mitarbeitereinsatz und Kosten planen

Für die Planung des **Mitarbeitereinsatzes** können Sie die Organisationseinheiten des Unternehmens mit den zugehörigen Mitarbeitern und Projekten in **objectiF RPM** abbilden. Pro Projekt werden u.a. Projektmitarbeiter und Projektgruppen mit ihren Kalendern und Aktivitätszuordnungen verwaltet. Verschiedene Sichten und Auswertungen zeigen Ihnen die **Mitarbeiterauslastung** in einem Projekt bzw. über sämtliche Projekte der Organisation.

Für die **Kostenplanung** können Sie pro Zeitraum interne und externe **Stundensätze** für Projekt, Mitarbeiter, Projektmitarbeiter und Projektgruppe definieren sowie **Zusatzkostenarten** pro Projekt festlegen.

Pro Aktivität liefert **objectiF RPM** Ihnen dann:

- die internen und externen Personalkosten Plan & Ist,
- zeitraumbezogene Budgets,
- Zusatzkosten Plan & Ist mit Fälligkeit und Buchungsdatum.

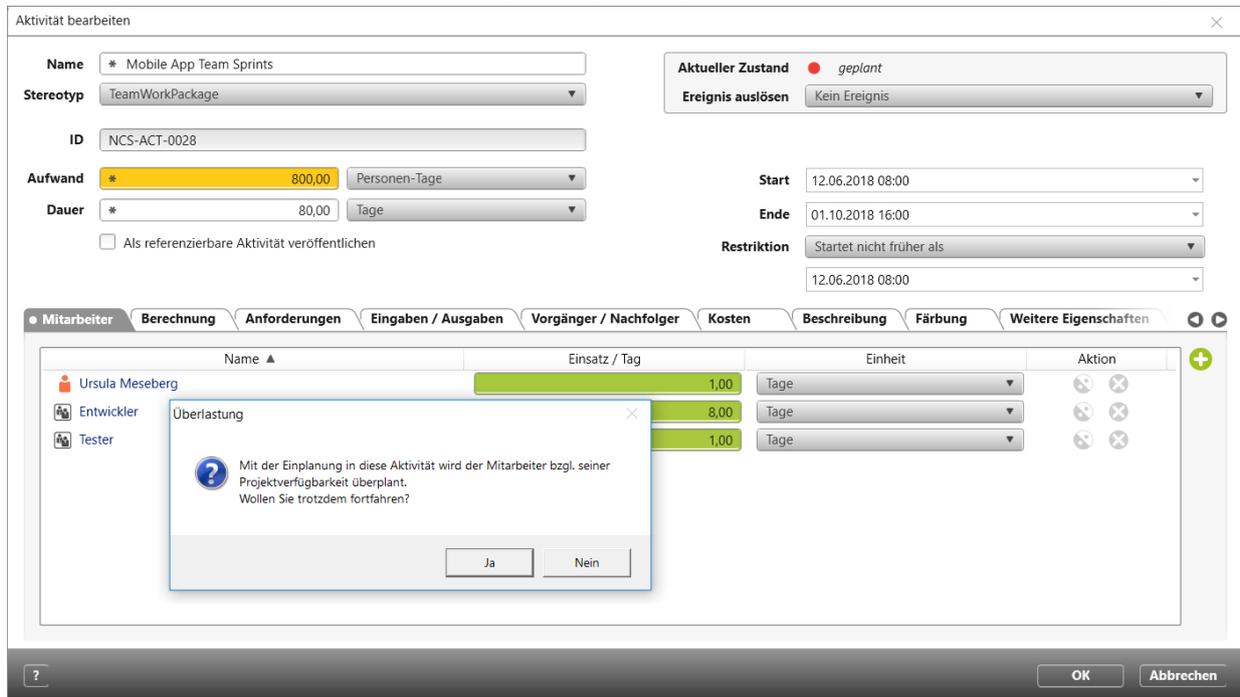


Abbildung 19: Schon bei der Planung einer Aktivität erhalten Sie Hinweise auf die Überplanung von Mitarbeitern

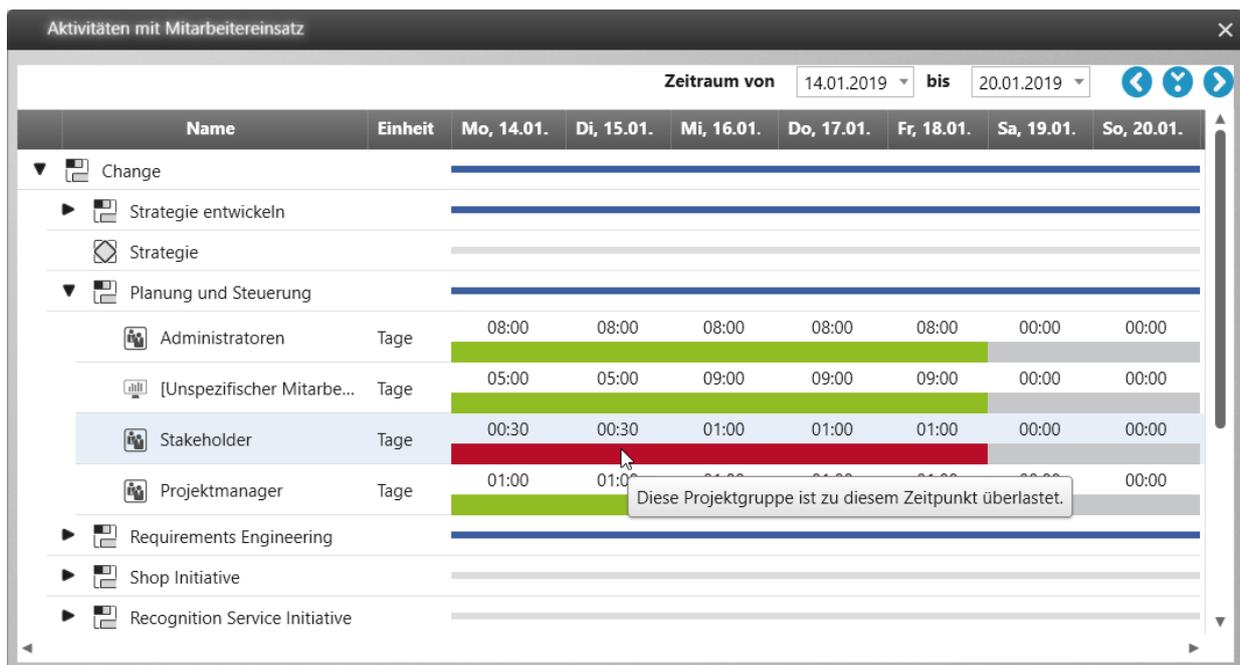


Abbildung 20: Eine von mehreren Sichten auf den Mitarbeiterinsatz

Projekt-Controlling

„Wie geht es meinen Projekten?“ – Nicht nur der Vergleich von Plan- und Ist-Kosten liefert darauf eine Antwort. Zum vollständigen Bild gehören u.a. auch, wie viele Anforderungen sich in welchem Zustand befinden, wie hoch die Zahl der realisierten und abgenommenen User Stories in einem Sprint ist, wie viele Bugs bekannt sind, etc. Mit **objectiF RPM** können Sie die gewünschten Informationen jederzeit in Echtzeit abrufen.

Als Manager sind Sie an Auswertungen und Analysen in komprimierter Form interessiert, um den Fortschritt Ihrer Projekte zu kontrollieren? **objectiF RPM** bietet dafür Dashboards in visuell ansprechender und leicht verständlicher Form.

Sie zeigen wichtige Kennzahlen der **Earned Value Analyse** wie z. B. die Kosten- und Zeiteffizienz. Außerdem können Sie z.B. **Burn Up-** und **Burn Down-Charts** sowie **Cumulative Flow Diagrams** in ein Dashboard übernehmen.

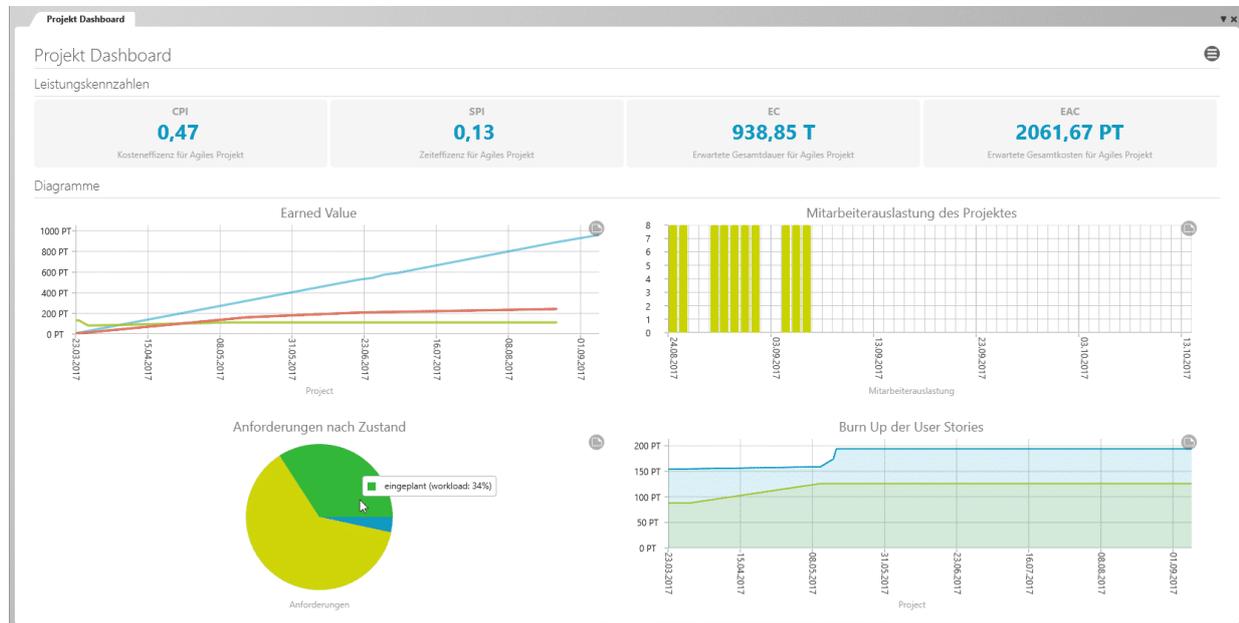


Abbildung 21: Beispiel eines Dashboards für das gesamte Projekt.

Sie bestimmen den Kontext eines Dashboards. So können Sie zum Beispiel individuelle Aktivitäten, einzelne Projekte oder ein Portfolio an Projekten in einem Dashboard überwachen.

Weitere Funktionen für das Projektmanagement:

- ✓ Road Map der Organisation
- ✓ Integrierte Zeiterfassung
- ✓ Projektübergreifender Mitarbeiterpool mit Projektanfragen
- ✓ Unternehmensweite Dashboards
- ✓ Generierung von Netzplänen
- ✓ Benutzerverwaltung mit Schnittstelle zu Active Directory
- ✓ Rechtemanagement
- ✓ Task-Manager für die zeitgesteuerte automatisierte Ausführung von Aufgaben
- ✓ Datenaustausch zwischen (verteilten) Projekten per Export/Import
- ✓ Projektsynchronisation per REST API Konfiguration mit JIRA

objectiF RPM auf Ihre Organisation zuschneiden

Müssen Sie spezielle Standards einhalten? Folgen Sie einem besonderen Prozess? Der Application Lifecycle sieht in jeder Organisation anders aus. Deshalb können Sie **objectiF RPM** Ihrem Bedarf entsprechend anpassen und sogar erweitern.

Benötigen Sie Ergebnistypen, die **objectiF RPM** nicht kennt? Sind Ihnen die angebotenen Formulare – z.B. zur Bearbeitung von Anforderungen – zu detailliert oder im Gegenteil nicht umfangreich genug? Gibt es wiederkehrende Workflows, die Sie maschinell unterstützen wollen? Fehlen Ihnen vielleicht sogar Tool-Funktionen? Das alles können Sie leicht ändern.

objectiF RPM anpassen

Sie wollen **objectiF RPM** optimal an Ihre Organisation anpassen? Dann reicht es oft schon aus, die vorhandenen Mittel geeignet zu konfigurieren:

- Ergänzen Sie die „mitgebrachten“ Ergebnistypen – z.B. Anforderungen oder Testfälle – um Eigenschaften, die für Ihre fachliche Domäne spezifisch sind.
- Legen Sie Muster an, mit denen Sie wiederkehrende Aufgaben z.B. bei der Projektplanung automatisieren.
- Definieren Sie eigene Echtzeitauswertungen.
- Erstellen Sie eigene Vorlagen für die Dokumentengenerierung.
- Richten Sie Dashboards, Backlogs, Sichten und Charts bedarfsgerecht ein.
- Passen Sie vorgeschlagene Workflows z.B. für Reviews an Ihre Abläufe an.

objectiF RPM erweitern

Diese Anpassungsmöglichkeiten allein reichen nicht aus, um Ihr Vorgehen zu unterstützen? Dann nutzen Sie die Erweiterungsmöglichkeiten von **objectiF RPM**. Und davon gibt es gleich mehrere:

- Sie können per Skript eigene Funktionalität für **objectiF RPM** entwickeln. Ihre neuen Funktionen binden Sie in die Kontextmenüs des Tools ein oder verwenden sie, um zeitgesteuert Aufgaben abzuarbeiten.
- Sie können eigene Ergebnistypen oder Spezialisierungen der vorhandenen anlegen, mit denen Sie dann genauso arbeiten wie mit den „mitgebrachten“. Beim Versionieren, Auswerten und Dokumentieren werden Sie keinen Unterschied bemerken.
- Sie können mit dem integrierten Formular-Designer eigene Formulare für die Bearbeitung von Ergebnissen generieren, die sich nahtlos in die Benutzeroberfläche von **objectiF RPM** einfügen.
- Sie können Beziehungen zwischen eigenen und vorhandenen Ergebnistypen definieren und damit für Traceability sorgen.

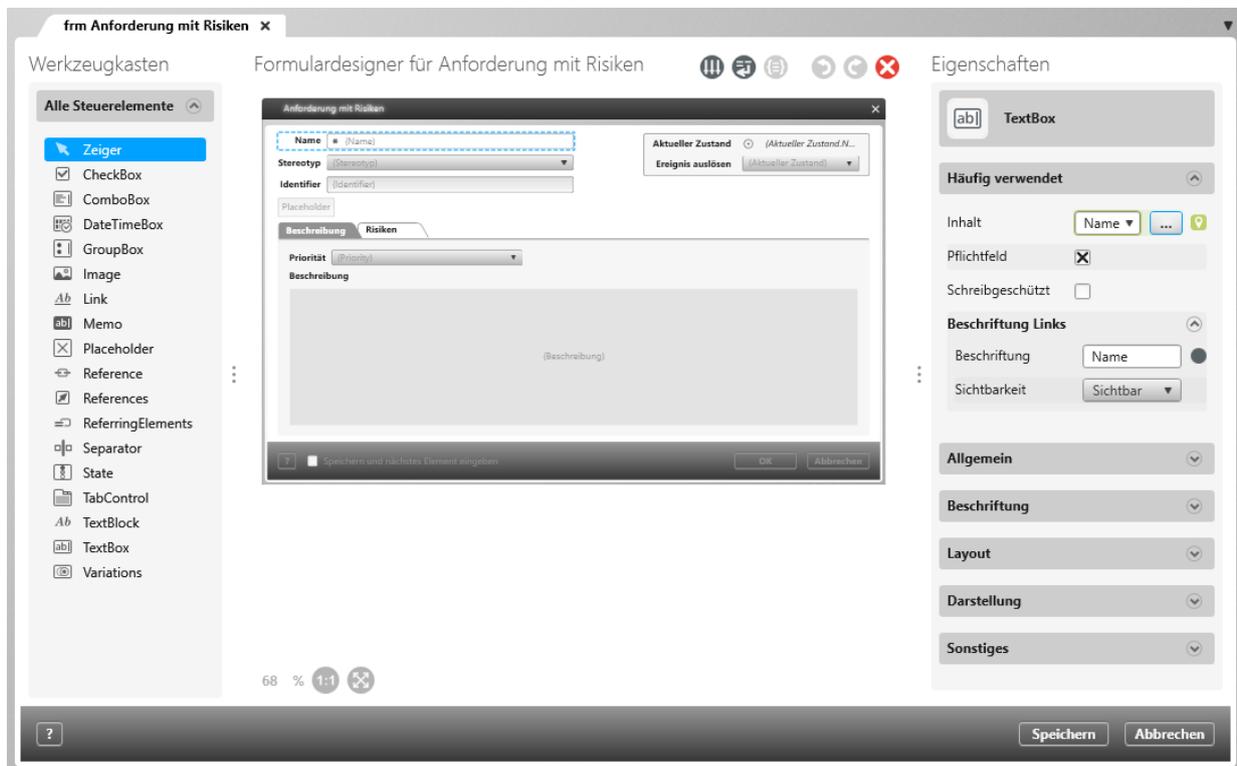


Abbildung 22: Der integrierte Formular-Designer von *objectiF RPM*:
Formular für einen vorhandenen und benutzereigenen Ergebnistyp entwickeln

Zustandsbasierte Workflows definieren

Und schließlich gibt es eine dritte Möglichkeit, *objectiF RPM* zur Unterstützung Ihrer Prozesse zu erweitern: die Definition zustandsbasierter **Workflows** für die Zusammenarbeit im Team. Damit können Sie zum Beispiel dafür sorgen, dass nur Anforderungen für einen Sprint eingeplant werden können, die ein Review durchlaufen haben.

Das „Geheimnis“ hinter den benutzerdefinierten Workflows: Jeder Ergebnistyp besitzt einen **Zustandsautomaten**, dargestellt als Zustandsdiagramm. Das zustandsabhängige Verhalten der Ergebnistypen passen Sie an Ihren Bedarf an, indem Sie die möglichen Zustände, bedingungsabhängige Zustandsübergänge und Aktionen festlegen – ganz einfach durch grafisches Editieren eines Zustandsdiagramms. Das heißt, mit einfachen visuellen Mitteln definieren Sie, welche Zustände z. B. eine Anforderung, ein Testfall, ein Dokument oder ein von Ihnen definiertes Ergebnis annehmen kann, wann ein Zustandswechsel eintreten soll und was passieren soll, wenn der Zustandswechsel ausgelöst wird.

Video-Links zum Erweitern von *objectiF RPM*

- [!\[\]\(674ed1a0c68b0bd1c10c40ff2fb5c580_img.jpg\) *objectiF RPM* um eigene Funktionen erweitern](#)
- [!\[\]\(be0406e9afddd5ed929b69591fd470f2_img.jpg\) *objectiF RPM* um eigene Artefakttypen erweitern](#)

Technische Details

Infrastruktur

Wenn Ihre Projekte wachsen, brauchen Sie ALM-Software, die mitwachsen kann. Wenn Sie mit vielen verschiedenen Szenarien verteilter Projektarbeit umgehen müssen, sollte Ihre ALM-Software außerdem eine flexibel konfigurierbare Architektur besitzen. **objectiF RPM** erfüllt beide Kriterien.

Sie können zwischen zwei Varianten von **objectiF RPM** wählen:

- **On-Premise:** für den Betrieb der Software auf Ihren Rechnern vor Ort (Lizenzkauf)
- **Software as a Service (SaaS):** für den Einsatz der Software in der Cloud von Amazon Web-Services (monatliche Miete).

Beide Varianten können auch in einem gemeinsamen virtuellen privaten Netzwerk (VPN) verknüpft werden. Verteilte Projektarbeit ist mit dem Zugriff per Web-Browser in beiden Varianten möglich. Welche Variante sollten Sie also wählen?

objectiF RPM als SaaS ist für Sie die richtige Wahl, wenn es morgen losgehen soll, wenn Sie ein Projekt ohne Aufwand und mit **geringen Investitionen in die Infrastruktur** sofort starten wollen.

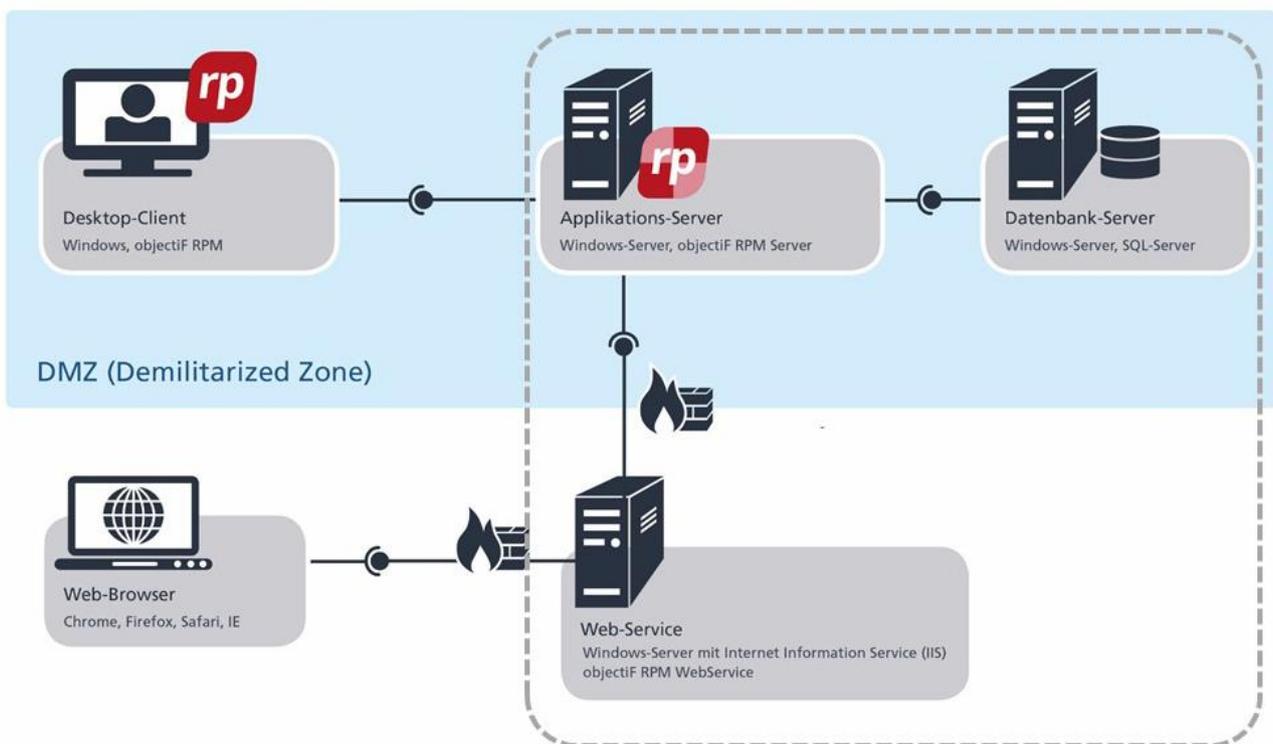


Abbildung 23: objectiF RPM On-Premise: Installation auf unternehmenseigenen Rechnern.

Applikations- und Datenbank-Server sowie Web-Service können auf gemeinsamer Hardware oder mehreren Rechnern installiert werden. Es geht auch ohne Installation von Client-Software: Einfach per Browser auf die Projekte zugreifen.

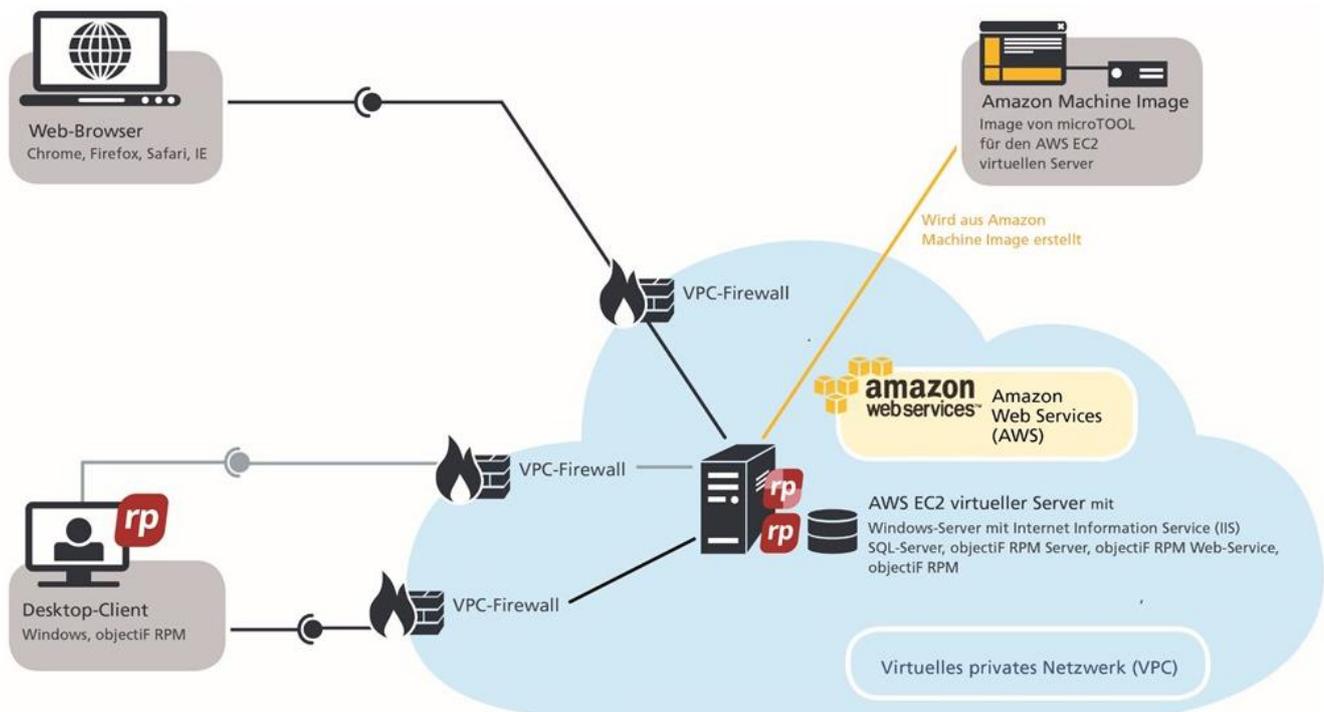


Abbildung 24: objectiF RPM als Software as a Service (SaaS) in der Cloud. Auch die SaaS-Lösung bietet Alternativen: Sie können mit lokalen Desktop-Clients arbeiten, auf denen die objectiF RPM Client Software installiert ist, oder in einem Amazon Workspace mit Windows, MS Office und objectiF RPM. Auf den lokalen Desktop-Rechnern wird keine objectiF RPM Software installiert. Es spielt es keine Rolle, ob Sie einen Windows-, Linux- oder OS X Desktop einsetzen.

Einsatzszenarios

objectiF RPM ist für die Arbeit im Team gemacht. Hier sind zwei typische Einsatzszenarios:

So arbeiten Sie in verteilten Teams zusammen: Die Architektur von *objectiF RPM* ermöglicht die unterschiedlichsten Formen der Zusammenarbeit. Grundlage der Architektur ist ein zentrales Realzeit-Repository. Seine Wirkung: Als Anwender sehen Sie bei der Arbeit immer den aktuellen Entwicklungsstand. Dabei spielt es keine Rolle, ob Sie im LAN mit einem Desktop Client arbeiten, auf dem die *objectiF RPM* Software installiert ist, oder ob Sie einfach im Web-Browser über das Internet von einem beliebigen Standort auf Ihr Projekt zugreifen.

So arbeiten Sie im Verbund mit externen Projekten: Um Daten – einzelne Artefakte, ganze Ordnerhierarchien und Work-Packages und ihre Beziehungen – mit anderen Projekten auszutauschen, die auch mit *objectiF RPM* durchgeführt werden, steht Ihnen eine integrierte **Export-/Import-Funktion** zur Verfügung. Mit externen Partnern, die *objectiF RPM* nicht einsetzen, können Sie Anforderungen per MS Excel austauschen oder das ReqIF-Format nach der Definition der OMG verwenden.

Schnittstellen

Schnittstelle	Anwendungszweck
Active Directory	Import von Mitarbeitern
JIRA	Import und Export von Issues
MS Excel	Import von Anforderungen, Klassen, Zielen, Aktivitäten und Testfällen Export von Listen
MS Word	Generierung von Dokumenten Import von Anforderungen
MS Visio	Export von Netzplänen
ReqIF	Import und Export von Anforderungen
REST API	Datenaustausch zwischen anderen Anwendungen
Visual Studio Code	Bearbeitung von Code-Dateien

objectiF RPM kennenlernen

Auf den vorangegangenen Seiten haben wir Ihnen die wichtigsten Schritte im Application Lifecycle mit *objectiF RPM* vorgestellt. Sicher sind noch Fragen offen: Was gibt es sonst noch an interessanten Features? Wie bedient sich *objectiF RPM*? Wie sieht die Arbeit konkret an einem Beispiel aus? Das entdecken Sie am besten selbst. Wir laden Sie jetzt ein:

Testen Sie *objectiF RPM* kostenlos mit einer funktional vollständigen Trial Edition:

[Hier geht's zum Download](#)