

THE BUSINESS AND IT ARCHITECTS

WHITEPAPER | TELEKOMMUNIKATION

RELEASEMANAGEMENT

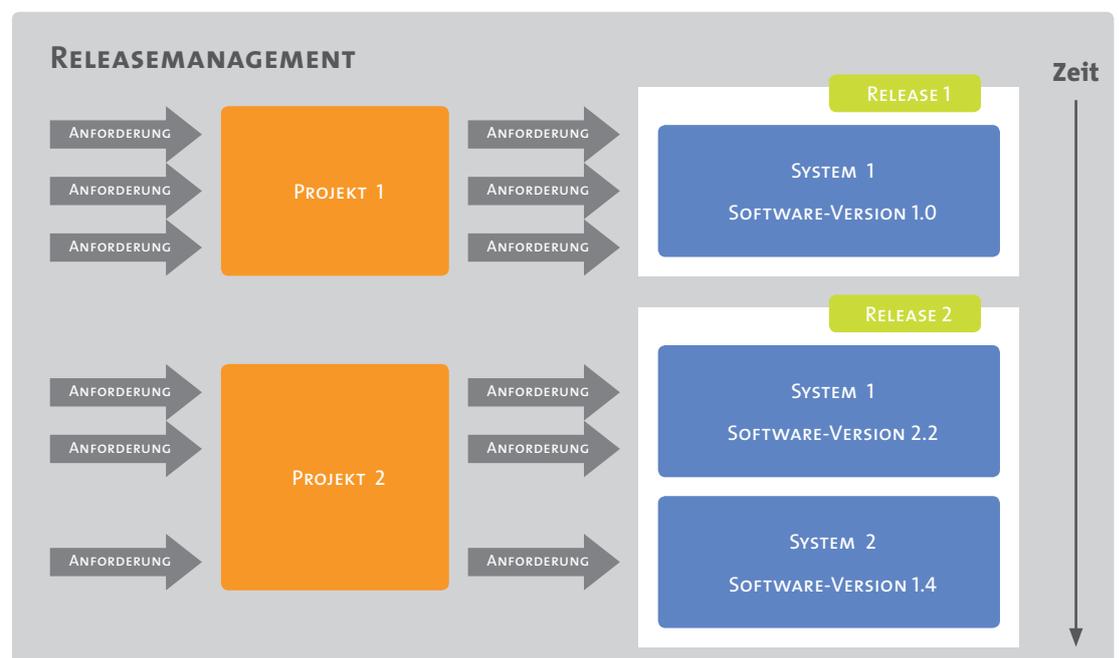
Klassisch oder agil?



EINLEITUNG Das Ziel eines Softwarebetreibers (Telekommunikationsunternehmen, E-Mail-Provider, Soziales Netz, Suchanbieter etc.) besteht darin, seinen Kunden mindestens die vertragsmäßige Verfüg- und Nutzbarkeit der Software zu gewährleisten. Das Releasemanagement sorgt dafür, dass der Stand der Software, mit der der Kunde in Kontakt kommt, von gleichbleibender bis steigender Qualität ist, aber ebenso, dass Neuerungen ohne bzw. mit möglichst geringer Beeinträchtigung den Wirkbetrieb aufnehmen. Diese Aufgabe kann verschiedenste Komplexität aufweisen. Sobald das angebotene Softwareprodukt aus mehreren Systemen (Hard- und Software) besteht, muss das Zusammenspiel unterschiedlicher Softwarestände sichergestellt werden.

BEGRIFFSDEFINITION Im Folgenden wird der Begriff „System“ für die Kombination aus Soft- und Hardware verwendet. Die Bezeichnung „Version“ beschreibt einen Softwarestand, wie er zu einem bestimmten Zeitpunkt vom Lieferanten (dem Softwarehersteller) an den Kunden (den Softwarebetreiber) geliefert wird. Eine Version umfasst mindestens eine Anforderung – sei es die Reparatur eines Fehlers oder eine als fehlend erkannte Funktionalität. Aus einer Version wird ein „Release“, wenn die Soft- und Hardware den Wirkbetrieb aufnimmt. Abbildung 1 stellt diese Zusammenhänge grafisch dar. Übergreifende Releases umfassen Änderungen an mehreren Systemen, teilweise auch über mehrere Abteilungen eines Unternehmens hinweg, um deren Zusammenspiel, z. B. bei einer Schnittstellenänderung, zu garantieren. Ein Release stellt zusammenfassend die Veröffentlichung eines Soft- und Hardwarestandes zu einem besonderen Zeitpunkt dar. Auch der Wechsel von Betriebssystemen oder die Reparatur von Hardware etc. erfolgen in Absprache mit dem Releasemanagement. Die Aufgabe des Releasemanagers besteht darin, den Wandel von Systemen über die Zeit hinweg zu verwalten.

Abbildung 1:
Releasemanagement.
Anforderungen an Systeme
werden in Releases gruppiert.
Eine Version wird zum Release,
wenn sie in der Wirkumgebung
ausgebracht wird.



**ABGRENZUNG
GEGEN CHANGE-,
VERSIONS-,
PATCH- UND
DEPLOYMENT-
MANAGEMENT**

Das Changemanagement koordiniert im Gegensatz zum Releasemanagement konfigurative Änderungen, Wartungsarbeiten, Fehlerkorrekturen und ähnliche Arbeiten. Es beschreibt die alltäglichen Änderungen an bestehende Releases und weist auf Betreiberseite viele Ähnlichkeiten zum Versionsmanagement des Softwarelieferanten auf.

Das Versionsmanagement ist nicht beim Softwarebetreiber, sondern beim Softwarelieferanten angesiedelt. Hier werden Anforderungen des Kunden, gefundene Fehler und Verbesserungen aus eigener Sicht an einem bestehenden oder neu zu entwickelndem Softwareprodukt gesteuert. Zu bedenken ist, dass nicht jede Version einer Software auch ausgeliefert wird.

Das Versionsmanagement hat aber auch Berührungspunkte mit dem Patchmanagement. Letzteres befasst sich mit der Behebung von Sicherheitslücken und der Auslieferung der korrigierten Software. Je nach Umfang oder Art der zu behebbenden Schwachstellen kann aus mehreren Patches durchaus eine neue Softwareversion entstehen.

Die Aufgabe des Deploymentmanagers ist die Koordination der Arbeiten am Wirtssystem und mögliche Interferenzen durch parallele Arbeiten in der Wirkumgebung auszuschließen.

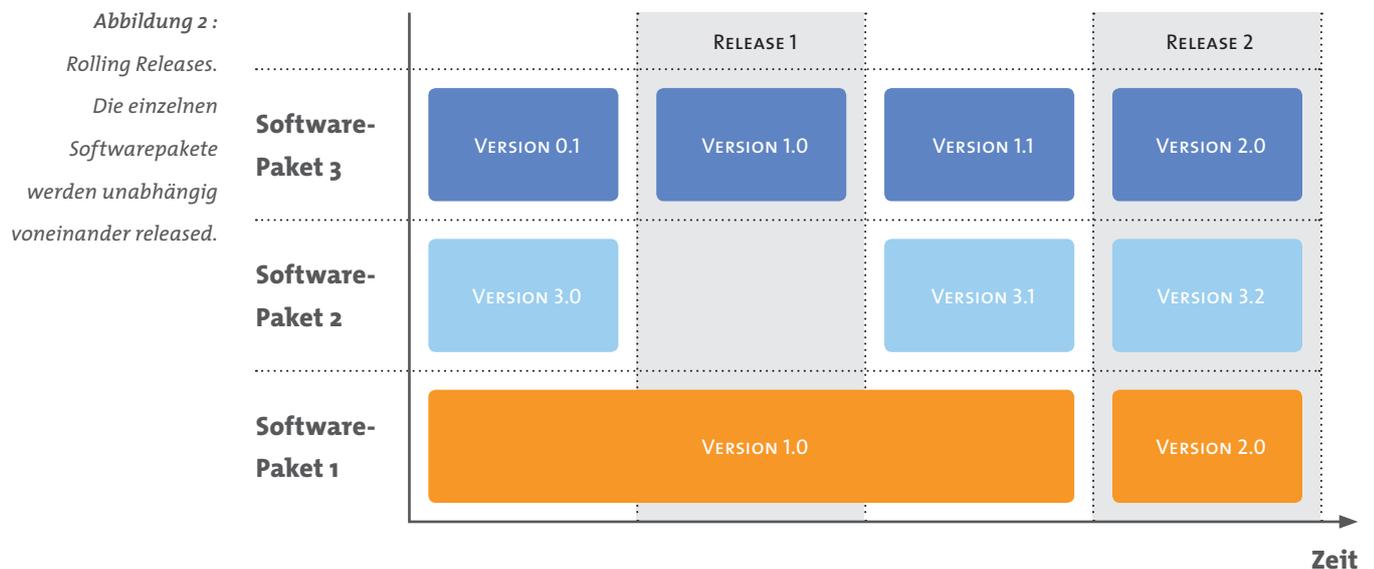
Unterstützt werden fast alle dieser Managementrollen durch ein einheitliches workflow-gestütztes Ticketsystem, das den aktuellen Soft- und Hardwarestand der Systeme, deren Änderungshistorie und einen kurzen Ausblick auf kommende Veränderung protokolliert, wie beispielsweise „JIRA“ der australischen Firma Atlassian.

**DIE VERSCHIEDENEN
RELEASEFORMEN**

Der Umfang eines Releases hängt sowohl von den fachlichen Anforderungen an das System als auch von der zugrunde liegenden Produktphilosophie ab. Am häufigsten findet man einen gleichmäßigen Releasezyklus – d.h. es wird beispielsweise jedes Quartal ein neues Release ausgebracht.

Im Gegensatz dazu stehen Projekte und Firmen, die in unregelmäßigen Intervallen releasen. Im Folgenden wird die Debian Linux Distribution als Beispiel genommen. Hier ist die Übereinkunft, dass der aktuelle Softwarestand eines Releases würdig ist, das Kriterium, um die Distribution zu veröffentlichen. Diese erhält dann einen Namen wie „Sid“, „Squeeze“, „Potato“ etc. Ebenso findet man in verschiedenen Linux-Distributionen den Ansatz von „Rolling-Release“. Darunter ist zu verstehen, dass die einzelnen Softwarepakete einer Distribution soweit wie möglich unabhängig vonei-

nander released werden. Dieses Konzept ist in Abbildung 2 dargestellt. Somit gibt es kein strenges Release für die Distribution, sondern die Pakete rotieren durch sie hindurch. Als bekanntestes Beispiel ist die „Testing“-Distribution von Debian zu nennen. Verwirrenderweise führt Debian aber auch „benahmste“ Releases für den Testingzweig. Somit benennt die Debian-Kommune eine zeitlich begrenzte Kombination an Softwarepaketen (Rolling-Releases) mit einem Namen und stellt dies parallel als „Testing“-Release zur Verfügung.

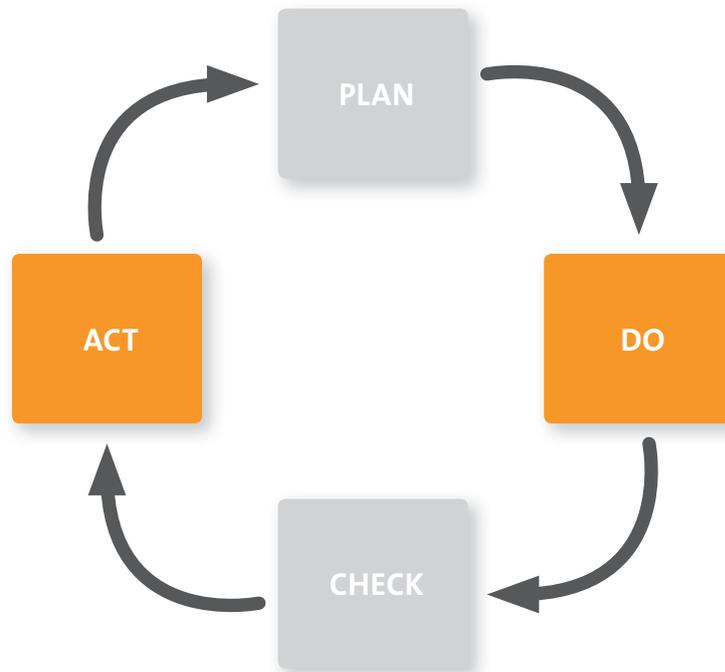


Man kann sich das wie eine übergreifende Markierung bzw. ein „tag“ in der Versionsverwaltung vorstellen – nur dass diesmal nicht ein Softwarepaket, sondern eine ganze Distribution markierend zusammengefasst wird.

RELEASEMANAGEMENT NACH ITIL

Nach diesem kurzen Abriss über die Kerntätigkeiten im Releasemanagement und verschiedener Releaseformen soll dieser Bereich im Größeren betrachtet werden (ITILv3). Im Plan-Do-Act-Check (PDCA, vgl. Abbildung 3) Modell der ITIL-Foundation ist das Releasemanagement in allen Bereichen von Bedeutung.

Abbildung 3:
 PDCA (Plan-Do-Check-Act) Model.
 Die vier Phasen des
 PDAC-Modells nach ITILv3.



Plan

In der Planungsphase werden Anforderungen an das Softwareprodukt gesammelt und gebündelt. Hier stellt das Releasemanagement die logische Klammer über ein Set an Anforderungen und koordiniert, dass entsprechende Änderungen, soweit benötigt, an allen involvierten Systemen veranlasst und durchgeführt werden. Damit sind zum Zeitpunkt der Betriebsaufnahme alle Systeme auf die auszurollenden Änderungen vorbereitet und das Zusammenspiel wurde vorab gründlich getestet.

Eine starke Überlappung des Releasemanagements mit dem Testmanagement und Anforderungsmanagement ist in der Planungsphase erforderlich. Nur dadurch lässt sich sicherstellen, welche Anforderungen an welche Systeme als Release zu einem bestimmten Zeitpunkt den Wirkbetrieb aufnehmen.

Während der Planungsphase müssen nicht nur die Abhängigkeiten zwischen den einzelnen Systemen berücksichtigt werden, sondern – soweit betroffen – auch die Integration und mögliche Interaktion mit Deployments aus anderen Fachbereichen/Abteilungen. Als Beispiel seien Wartungsarbeiten an Netzsegmenten, die den administrativen Zugriff auf die eigenen Systeme unterbinden könnten, sowie Wartungsarbeiten an der Stromversorgung der Rechenzentren etc. genannt.

Do

Des Weiteren ist es Aufgabe des Releasemanagers, eine Übersicht über den aktuellen Stand der Wirkumgebung mit den durchgeführten sowie geplanten Releases zu pflegen. Diese Sichten dienen den Team- und Projektleitern wie auch anderen Abteilungen als Planungsgrundlage. Es empfiehlt sich, die Übersichten pro System kalendarisch zu pflegen.

Die Rolle des Releasemanagers beinhaltet auch die rechtzeitige Kommunikation von Maßnahmen an weitere Release- und Changemanager, um die Vorbereitungen für Postdeployment-Tests (PDTs) vor dem Ausbringen eines Releases umsetzen zu lassen.

Check

Die Check-Phase des ITIL-Zyklus kann aus zwei verschiedenen Gesichtspunkten betrachtet werden. Im Folgenden wird zuerst von einem Release ausgegangen, das schon mindestens einen Zyklus durchlaufen hat. Es beinhaltet also Verbesserungen gegenüber dem vorausgegangenen Releases. Hierbei kommt dem Nachweis von System-, Verbund-, Performance- und Sicherheitstests eine große Wichtigkeit zu. Sie dienen als Verifikation der Funktionalität und Stabilität und liefern die Entscheidungsgrundlage für die Veröffentlichung in der Produktionsumgebung. Je nach Kritikalität der Softwarekomponente müssen die einzelnen Testergebnisse unterschiedlich gewichtet werden. Die Bewertung der Testnachweise und der Interessen der „Stakeholder“ erfolgt in einem eigens dafür einberufenen Meeting. Andere Formen der Kommunikation, z. B. telefonisch oder per E-Mail, sind ebenso gängig. Hier bestimmen der Umfang eines Releases und die Wichtigkeit der zu verändernden Systeme das Maß.

Nach einem erfolgreichen Rollout eines Releases sind die Ergebnisse der Postdeployment-Tests eine der wichtigsten Informationsquellen für das Releasemanagement. Diese Tests sollten nicht nur technischen, sondern auch fachlichen Charakter haben. Vor diesem Hintergrund ist schon in der frühen Phase der Releaseplanung und des Projekts auf den Aufbau einer umfassenden Menge an Regressions- und Postdeployment-Tests zu achten. Für die Durchführung der Postdeployment-Tests müssen ggfs. noch entsprechende Änderungen und Konfigurationen an weiteren Wirksystemen gemacht werden, welche durch den Releasemanager veranlasst werden.

Act

Die aktiven Tätigkeiten des Releasemanagers beruhen nicht nur auf der Koordination der einzelnen Deployments und deren Abhängigkeiten, sondern auch in der Verifizierung der Release-Tauglichkeit eines Softwareprodukts. Erkannte und ggf. sogar wiederholt auftretende Probleme und Engpässe müssen berücksichtigt und nach Möglichkeit auf ein Minimum reduziert werden.

DER AGILE WEG

„Agiles Releasemanagement“ mag auf den ersten Blick befremdlich klingen. Jedoch ist in agilen Projekten diese Aufgabe sehr wichtig, denn sie haben häufig einen hochfrequenten Releasezyklus. Je nach Organisationsform und gewähltem Vorgehensmodell überlappt sich die Rolle des Releasemanagers mit anderen agilen Rollen. Dies ermöglicht es, nicht nur fachliche und technische, sondern auch wirtschaftliche und marketing-spezifische Aspekte mit in die Releaseplanung zu integrieren und so einen ganzheitlicheren Ansatz zu wählen.

Deutlich wird dies am Beispiel des Product Owners in **SCRUM**. Diese Projekttrolle umfasst die klassischen Rollen des Projektleiters als auch des Produktmanagers. Neben der Release-/Sprintplanung und Backlogpflege beschäftigt sich der Product Owner mit Marktanalysen sowie fachlicher Projektsteuerung durch Priorisierung und versucht, eine einheitliche Produktvision in den Köpfen des SCRUM-Teams, der Kunden und Stakeholder zu festigen.

Neben SCRUM wird im agilen Umfeld auch **KANBAN** immer häufiger eingesetzt. Im Gegensatz zu SCRUM ist diese Vorgehensweise besser für die Pflege und Erweiterung bereits bestehender Produkte geeignet und eher evolutionär. KANBAN sieht einen Releasemanager in dieser Form nicht explizit vor. Dennoch gibt es Aufgaben, die einem klassischen Releasemanager/-verantwortlichen zugeordnet werden würden. Die eingehenden Anforderungen/Tickets und deren Weg durch den zugrunde liegenden Workflow müssen koordiniert werden. Der Releasemanager stellt sicher, dass eine in sich konsistente Kombination an Anforderungen/Tickets zu einem festen Zeitpunkt den letzten Schritt des Workflows (Deployment/Auslieferung) erreicht. Kanban ist ein Pull-basierendes System, d. h. die Anforderungen werden zusammengetragen und das Projektteam bearbeitet diese nach eigener Wahl. Für das Releasemanagement bedeutet dies aber, ein Release zu einem Zeitpunkt zu planen, an dem noch nicht alle Anforderungen an die neue Softwareversion bekannt sind.

Bisher noch nicht diskutiert ist der Punkt, dass ein agiles Release immer ein vom Kunden nutzbares Softwareinkrement darstellen muss – egal welches konkrete Vorgehensmodell gewählt wurde. Das ist im klassischen Releasemanagement in dieser Strenge nicht gefordert. Hier gibt es die Möglichkeit, Funktionalitäten über mehrere kurz aufeinander folgende Releases zu strecken oder gar BugFix-Releases zu veröffentlichen.

**ANFORDERUNGEN
AN EINEN
RELEASEMANAGER**

Das Releasemanagement stellt besondere Herausforderungen an den Verantwortlichen. Er muss nicht nur die Interessen der Kunden mit den Möglichkeiten des Projektteams (erfolgsrelevanter Faktor in agilem Umfeld!) und dem aktuellen Stand der Soft- und Hardware vereinen, sondern auch dafür Sorge tragen, dass eine stabile, ausgewogene Kombination an Änderungen in wohl getestetem Zustand ihren Weg in den Wirkbetrieb findet. Gerade Letzteres ist im häufigen Projektdruck nicht immer in vollem Umfang umsetzbar. Wirtschaftliche Aspekte und unverhältnismäßig hohe Kosten für umfassende Tests können durchaus an Gewicht gewinnen. Das Augenmerk des Release-Managers muss jedoch in erster Linie auf der Stabilität der eigenen Systeme liegen. Gerade bei Querschnitt-Systemen wie der Benutzerverwaltung (Identity Management Systems) ist dies von äußerster Wichtigkeit. Deren Ausfall würde zwangsweise die Nutzbarkeit einer größeren Menge an Produkten unterbinden.

FAZIT Im Vergleich zum klassischen Releasemanagement hat ein agiles Vorgehen durchaus Vorteile:

Es kann flexibler auf Änderungen reagiert werden. Sei es, dass sich die Vorstellungen der Kunden während der Entwicklung vom ursprünglichen Anforderungsspektrum entfernen, als auch, dass durch zusätzliche Stakeholder oder sich ändernde wirtschaftliche Situationen die Produktvision wächst oder schrumpft. Dies spiegelt sich neben einer dynamischeren Verfahrensweise mit Anforderungen auch in kürzeren zeitlichen Abständen zwischen den einzelnen Releases wider.

Im agilen Umfeld verschmelzen Release- und Versionsmanagement miteinander, da am Ende eines jeden Entwicklungszyklus ein Release veröffentlicht wird. Ähnlich ergeht es auch der Rolle „Releasemanager“. Häufig wird diese nicht mehr explizit definiert, sondern in größere Aufgabengebiete/Projektrollen integriert. Dadurch finden Aspekte wie Produktvision, Optimierung des Arbeitsprozesses und Analyse des auftretenden Anforderungsspektrums verstärkt Eingang in die Releaseplanung.

Diese Verbreiterung der Anforderungen an die unter anderem für das Releasemanagement zuständige agile Projektrolle birgt aber auch Gefahren in sich: Zum einen steigt die Belastung für die Betroffenen. Zum anderen werden bei der Besetzung der agilen Rolle die geforderten Qualifikationen gerne unterschätzt – z. B. die Besetzung der Product Owners in **SCRUM** durch den Fachbereich (eheml. Produktmanager). Dies kann in einem auf der technischen Ebene nicht ausreichend qualifizierten Kandidaten resultieren.

Die höhere Flexibilität in der Releasegestaltung und der ganzheitlichere Ansatz eines agilen Releasemanagers bieten eine sehr gute Basis für ein erfolgreiches Projekt oder Produkt. Agiles Releasemanagement fördert die Kundenzufriedenheit und sichert nachhaltig den Erfolg und die eigene Marktposition.

Die SYRACOM Consulting AG hat mit diesem Ansatz sehr gute, langjährige Erfahrungen in komplexen Geschäftsumfeldern gemacht. Dabei hat sich der Einsatz von Atlassian „JIRA“ als technisches Basis-System für das Releasemanagement bewährt: durch Erweiterung der Standardfunktionalität können alle benötigten Aspekte zügig und „in time & budget“ ergänzt werden. Unter dem Ansatz eines übergreifenden Anwendungslebenszykluses (ALM) lassen sich die oben angesprochenen Überlappungen zu den anderen Managementdisziplinen effektiv, effizient und ohne Medienbrüche implementieren. So entsteht eine praktikable Transparenz über die eigenen IT-Systeme.



SYRACOM

THE BUSINESS AND IT ARCHITECTS

Ansprechpartner:

Dr. Markus Maria Miedaner

Otto-von-Guericke-Ring 15

65205 Wiesbaden

Tel: +49 6122 9176-0

www.SYRACOM.de

Markus.Miedaner@SYRACOM.de