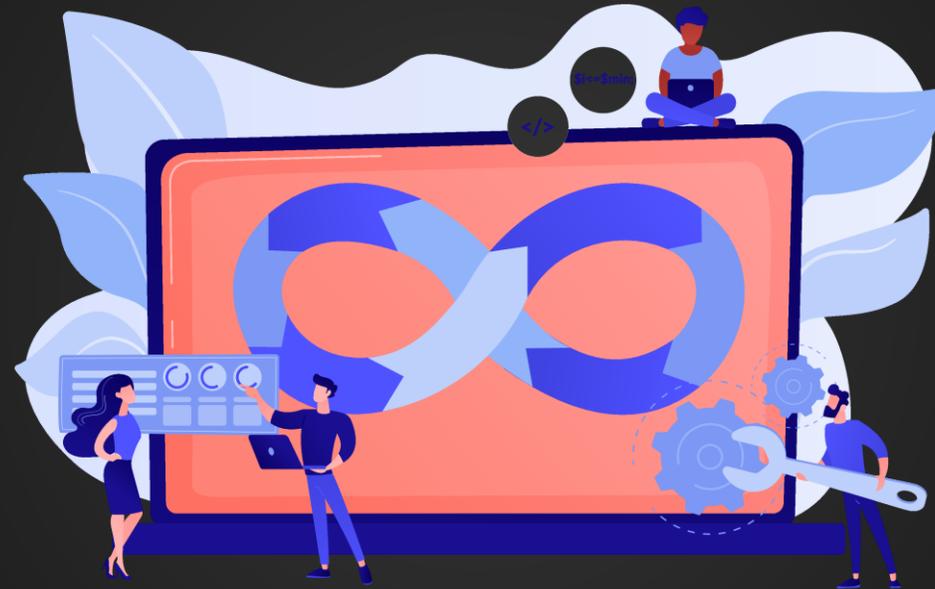


## Application Lifecycle Management (ALM)



Volatile Geschäftsumfelder bringen ständige Herausforderungen mit sich, auch im Hinblick auf die Softwareentwicklung und -bereitstellung. Um sich den ständig verändernden Anforderungen gut stellen zu können, werden flexible, leicht adaptierbare Softwarewelten geschaffen, um langfristige und träge Mammut - Softwareprojekte zu vermeiden. Das Application Lifecycle Management unterstützt, bei einem effizienten Projektmanagement einerseits und dem erfolgreichen Management des Software-Lifecycles andererseits.

### Ansprechpartner

Angelina Schock  
Managing Partner – GCS Consulting GmbH  
E-Mail: [schock@gcs-consulting.de](mailto:schock@gcs-consulting.de)  
Mobil: +49 89 8913650

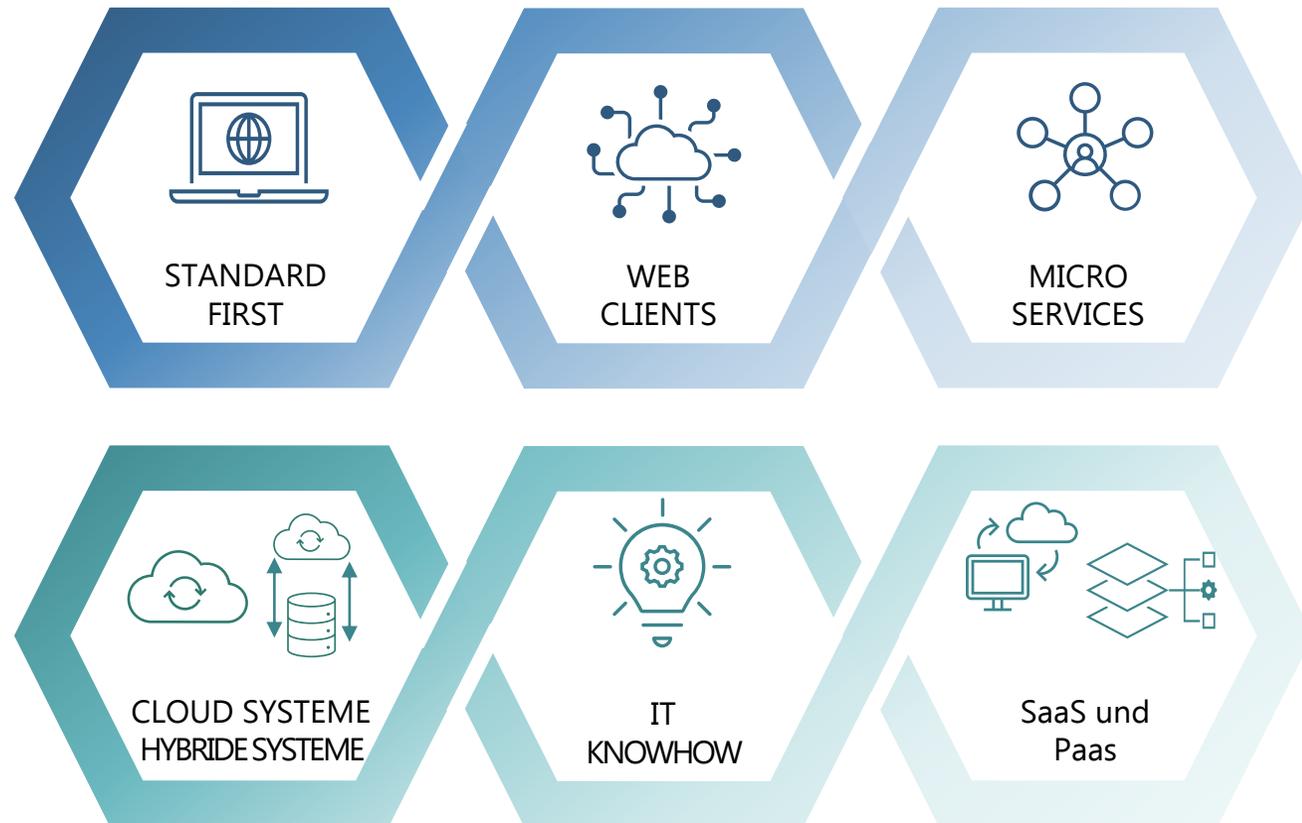
## Trends bei IT-Systemen

Die fortschreitende Digitalisierung hat weitreichende Auswirkungen auf verschiedene Aspekte der Geschäftswelt. Es verändert sich nicht nur die Natur der Geschäftsprozesse selbst, sondern auch die Dynamiken der internen und externen Zusammenarbeit. Im Kontext der Digitalisierung erfährt auch die Entwicklung der IT-Systeme einen Wandel. Projekte werden neu gedacht und Prioritäten neu gesetzt. Folgende Trends bei der Entwicklung von IT-Systemen lassen sich beobachten:

- + **Standardsysteme vor individualisierten Systemen**  
Standardsysteme sorgen für größere Flexibilität im Vergleich zu hochindividualisierten Systemen
- + **Moderne Architekturen**  
Client-Server-Systeme werden durch Web Clients abgelöst
- + **Micro-Services statt monolithischen Systemen**  
Die App im operativen Alltag erhöht die Usability
- + **Moderne Betreiber Modelle**  
Cloud basierte Systeme und hybride Systeme statt On-Premise-Lösungen
- + **IT- Systeme als Kernkompetenz**  
Knowhow Aufbau in der eigenen Organisation für mehr Innovation, Flexibilität und Schnelligkeit
- + **Bedarfsangepasste Lizenz-Modelle**  
SaaS (Software as a Service) und PaaS (Plattform as a Service) für optimale Skalierbarkeit

Moderne Systeme erfordern ein durchgängiges Application Lifecycle Management, um die Potentiale solcher Lösungen optimal zu nutzen und einen effizienten Betrieb zu gewährleisten.

## TRENDS BEI SYSTEM-EINFÜHRUNGEN



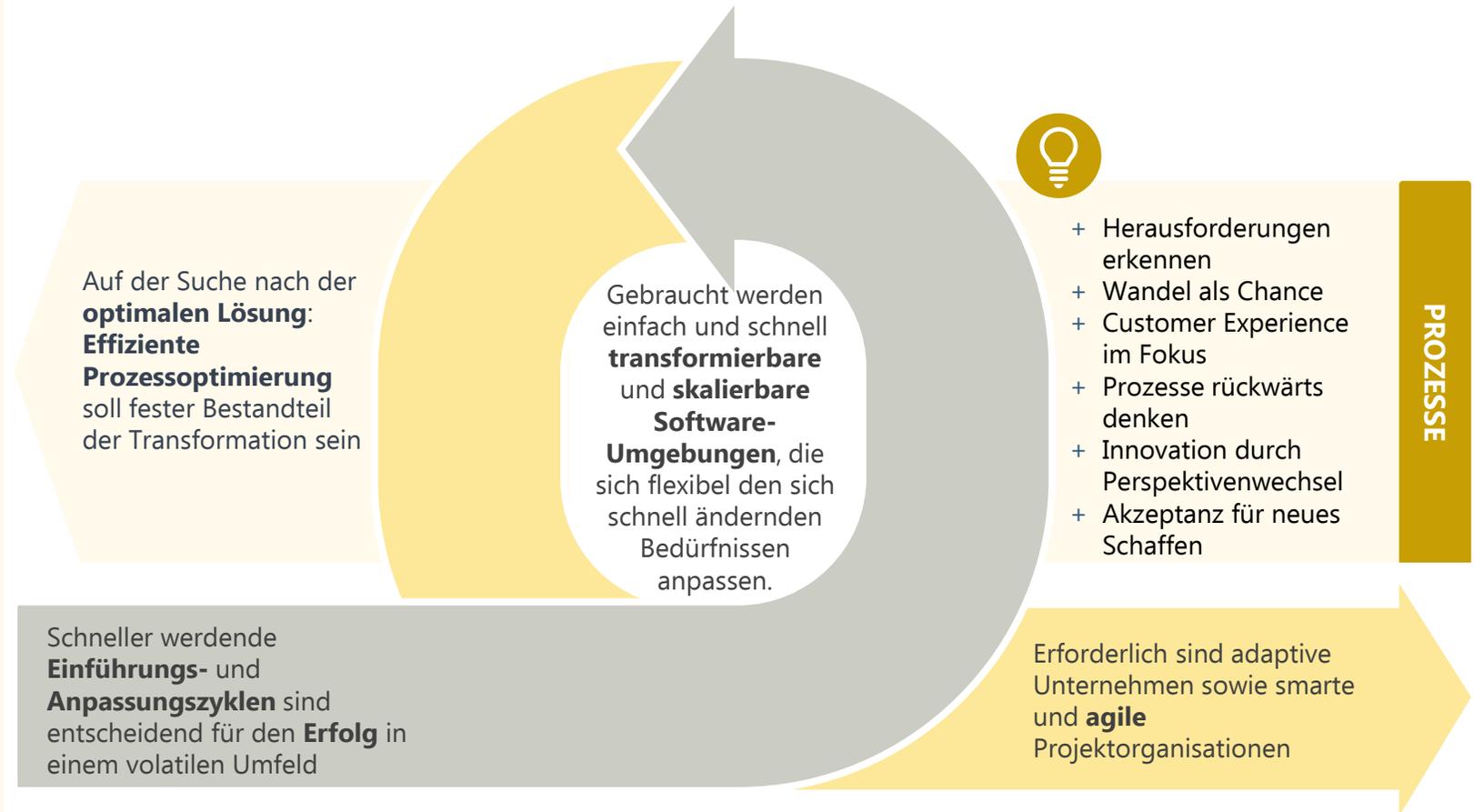
## Warum Application Lifecycle Management?

Auch wenn es sich bei Softwareprojekten in Unternehmen in der Regel um abteilungsübergreifende, komplexe Projekte, mit einem Impact auf viele Mitarbeiter und großer Sichtbarkeit innerhalb der Organisation handelt, möchte man mit der Nutzung von ALM eine andere Dynamik erzielen. So sollen rollierende Anforderungsaufnahmen und entsprechende Softwareanpassungen dafür sorgen, lange Projektlaufzeiten mit großem Scope zu vermeiden, sondern die Software ständig in kleineren Schritten anzupassen. Statt eines GoLive, der einem Event gleich kommt, nach einer langen Projektdauer, setzt man auf die kontinuierliche Erneuerung.

Moderne Cloud-basierte Systemlösungen machen ALM unumgänglich, da es auch nach der Einführung einer intensiven Betreuung bedarf, durch ein Team von Key-Usern, das über den IT-Bereich und die Entwicklung hinausgeht. Zum einen verlangen veränderte Rahmenbedingungen eine kontinuierliche Anpassung von Software-Systemen mit einhergehenden Prozessanpassungen. Zum anderen müssen auch Releasewechsel gemanagt werden. Setzt man auf Standardsysteme, stellen Softwareanbieter regelmäßig neue Updates zur Verfügung, für die das entsprechende Deployment bindend ist, will man nicht den Support des Softwareherstellers verlieren.

Application Lifecycle Management Systeme bieten den Rahmen dafür, die Software kontinuierlich zu managen, da die gesamte Lebensdauer der Systemlösung berücksichtigt wird und somit das Monitoring von Wartungsarbeiten und Anpassungen begleitet werden kann.

## TRANSFORMIERBARE SYSTEME FÜR EINE VOLATILE GESCHÄFTSWELT



## Was ist Application Lifecycle Management?

Unter Application Lifecycle Management (ALM) versteht man einen ganzheitlichen Ansatz für die Verwaltung und Steuerung einer Software-Applikation über den gesamten Lebenszyklus hinweg. Dabei geht es nicht nur um das Managen einer Software nach der Einführung, sondern vielmehr um die permanente Weiterentwicklung, um langfristige, große Softwareprojekte, die "from the scratch" angesetzt werden müssen, zu vermeiden. Mit ALM will man die Software permanent auf einem technisch- und inhaltlichen aktuellen Stand halten, angepasst an die aktuellen Geschäftserfordernisse. Damit bleibt nicht nur die Software aktuell, sondern auch das Unternehmen wettbewerbsfähig und Workarounds, um sich aktuellen Geschäftsgegebenheiten zu stellen, können vermieden werden.

In früheren Entwicklungsprozessen in der Wasserfallmethode waren die einzelnen Bereiche traditionell oft getrennt. Das Application Lifecycle Management unterstützt auch die integrierte Verwaltung von Aufgaben und Teams. Hierbei geht es im Sinn einer agilen Vorgehensweise um die Einbindung von Entwicklung, IT-Betriebsteam und nicht zuletzt den Anwendern, was zu einer größeren Effizienz im Unternehmen führt. Dieses Vorgehen beschleunigt Deployments, verbessert die Transparenz der Workflows und sorgt damit für optimierte Produkte, eine verbesserte Qualität und schafft Zufriedenheit bei allen Beteiligten. Durch diese umfassende Methode wird nicht nur ein reibungsloser Betrieb sichergestellt, sondern es kann auch flexibel auf veränderte Bedürfnisse reagiert werden, indem entsprechende Anpassungen geplant und vorgenommen werden.

Durch dieses Vorgehen sowohl bei der Aktualisierung der Software als auch bei dem Projektmanagement werden Ressourcen eingespart und das Unternehmen bleibt technisch auf dem aktuellsten Stand.

## APPLICATION LIFECYCLE MANAGEMENT



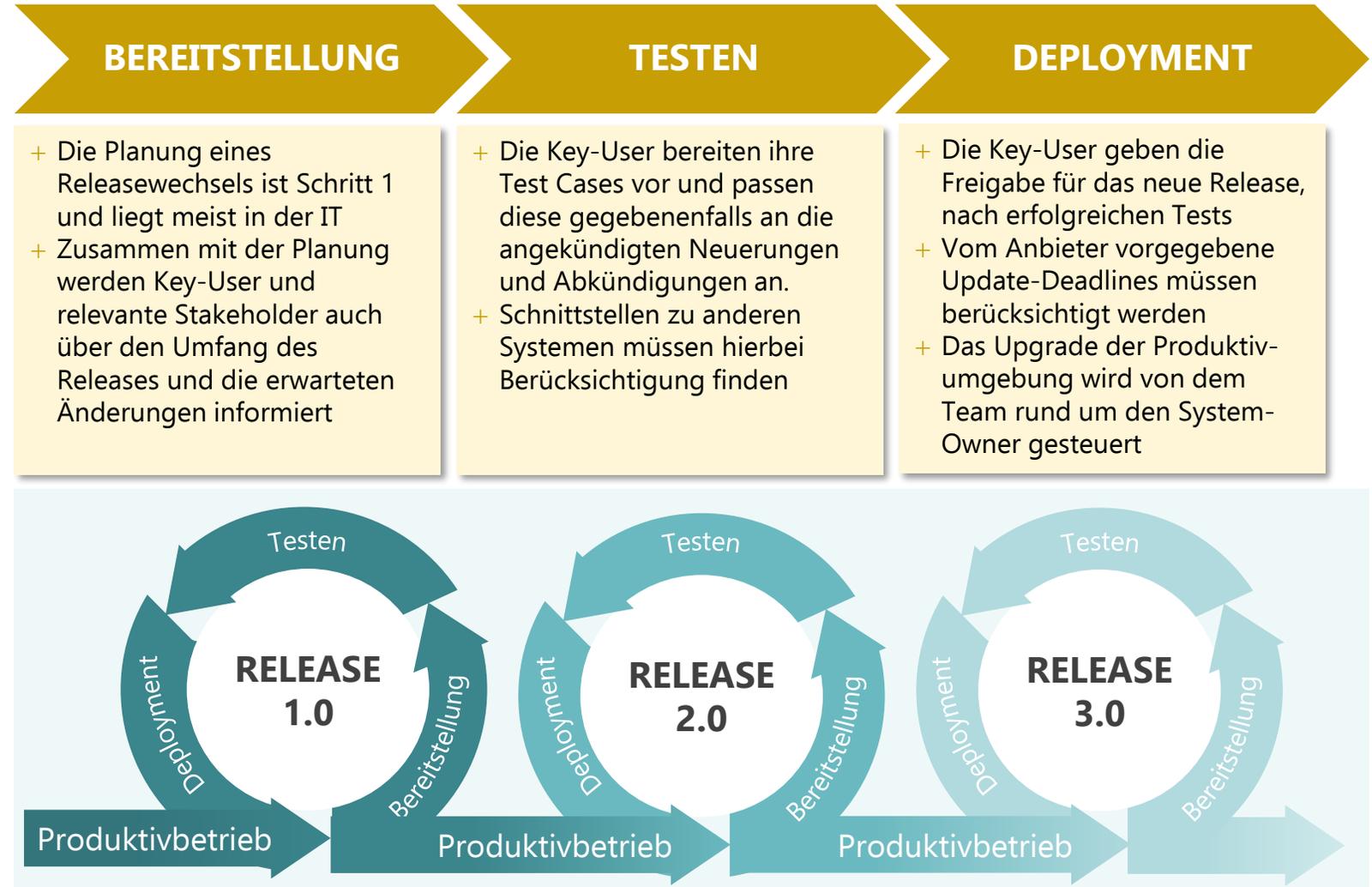
## Management von Releasewechselln

Softwaresysteme sind regelmäßigen Änderungen und Aktualisierungen unterworfen. Als Releases werden aktuelle Entwicklungsstände bezeichnet, die in der Regel mit Versionsnummern organisiert sind. Folgende Punkte sind ursächlich für neue Releases:

- + **Generelle Neuerungen und Veränderungen**  
Diese Veränderungen werden im Vorfeld von dem Softwareanbieter über Release-Notes bekanntgegeben. Es handelt sich dabei um Weiterentwicklungen und Verbesserungen. Etwaige Plug-Ins und Eigenentwicklungen müssen dementsprechend angepasst werden, damit die gewünschten Funktionen auch mit den Neuerungen noch funktionieren.
- + **Abkündigung von Funktionen**  
Auch für diesen Bereich gibt es eine Information über die Release-Notes. Es muss geprüft werden, ob ggf. Prozesse innerhalb der Organisation geändert werden müssen, wenn Funktionalitäten zukünftig nicht mehr verfügbar sein werden.
- + **Generelle Bugfixes**  
Auch allgemeine Fehlerbehebungen werden mit einem neuen Release ausgeliefert.

Die jeweils neuste Version wird im Rahmen eines Releasewechsels mit einem Deployment auf die Produktivumgebung und in den operativen Einsatz gebracht. Diese regelmäßigen Software-Updates sind Teil des Regelbetriebs und wollen gut geplant und vorbereitet sein, damit es zu keinen Einschränkungen im Produktivbetrieb kommt. Hier kommt es vor allem auf ein gutes Zusammenspiel zwischen Softwarehersteller, IT-Betriebsteam bzw. Solution Architekten und den Key-Usern an.

## ABLAUF EINES RELEASEWECHSELS



## Testmanagement

Auch wenn der Testumfang bei einem Releasewechsel nicht dem gleichen Umfang entspricht, wie dem bei einer Systemeinführung, führen diese regelmäßig wiederkehrenden Tests trotzdem zu einem hohen zeitlichen Aufwand für die Key-User. Da es jedoch entscheidend ist, sicherzustellen, dass die Aktualisierung reibungslos verläuft und keine neuen Probleme mit der neuesten Version auftreten, gerade auch im Hinblick auf angrenzende Systeme, die mittels direkten Interfaces oder einer Middleware angebunden sind, sind die Tests zu jedem Releasewechsel unumgänglich. Nur so lassen sich Ausfallzeiten aufgrund von Fehlern vermeiden.

Um dennoch den Testaufwand so gering wie möglich zu halten, gibt es verschiedene Ansätze, die in Betracht gezogen werden können:

- + **Aufsetzen von Testszenarien**  
Aufbau von guten und vollständigen Testszenarien, die für jeden Releasewechsel wieder verwendet werden können und die nur im Fall von grundsätzlichen Systemänderungen angepasst werden müssen.
- + **Priorisierung von Tests**  
Identifizierung von kritischen Geschäftsfunktionen. Diese müssen zuerst durchgeführt werden, da Ausfallzeiten durch Fehler besonders kritisch wären.
- + **Automatisierung von Tests**  
Wiederholbare Tests lassen sich, um Zeit zu sparen, auch automatisieren. Hierbei ist jedoch Aufwand / Nutzen gut abzuwägen, da die Tests sehr gut und detailliert formuliert sein müssen.

## TESTUMFANG FÜR RELEASEWECHSEL

### Funktionalitäten

- + Einzelne Funktionalitäten werden getestet
- + Key-User können unabhängig voneinander testen
- + Etwaige Bugs in einzelnen Funktionalitäten haben keinen Einfluss auf andere Tests

### Neuerungen und grundlegende Veränderungen

- + Es muss ausführlich und umfangreich getestet werden
- + Es ist besonders wichtig, nicht nur vorhandene Daten zu testen, sondern auch grundlegende Neuanlagen zu machen

### E2E Prozesstests

- + Es soll entlang der Prozesslandkarte getestet werden, um sicherzustellen, dass die Abläufe funktionieren
- + Besonderes Augenmerk gilt hier den Schnittstellen und den angebundenen Drittsystemen

Die **genaue Definition** der **Test-Cases** sowie der **Testumfang** liegt bei den **Key-Usern**. Es muss sichergestellt werden, dass alle relevanten **Benutzerszenarien erfolgreich** getestet werden.

- + Wie wird die Priorität der Funktionalitäten und Prozesse definiert?
- + Wie viele Test-Cases sind notwendig?
- + Welche Edge-Cases werden getestet?

## Fazit

ALM-Tools unterstützen das Application Lifecycle Management indem sie Funktionen für das Anforderungsmanagement, Konfigurationsmanagement, Testmanagement, die Fehlerverfolgung sowie für die Zusammenarbeit bieten. Durch die Integration dieser Tools können Teams effizienter arbeiten und die Gesamtqualität ihrer Softwareprodukte verbessern. ALM trägt dazu bei, die Transparenz im Entwicklungsprozess zu erhöhen und ermöglicht eine bessere Kontrolle über den gesamten Lebenszyklus der Anwendung.

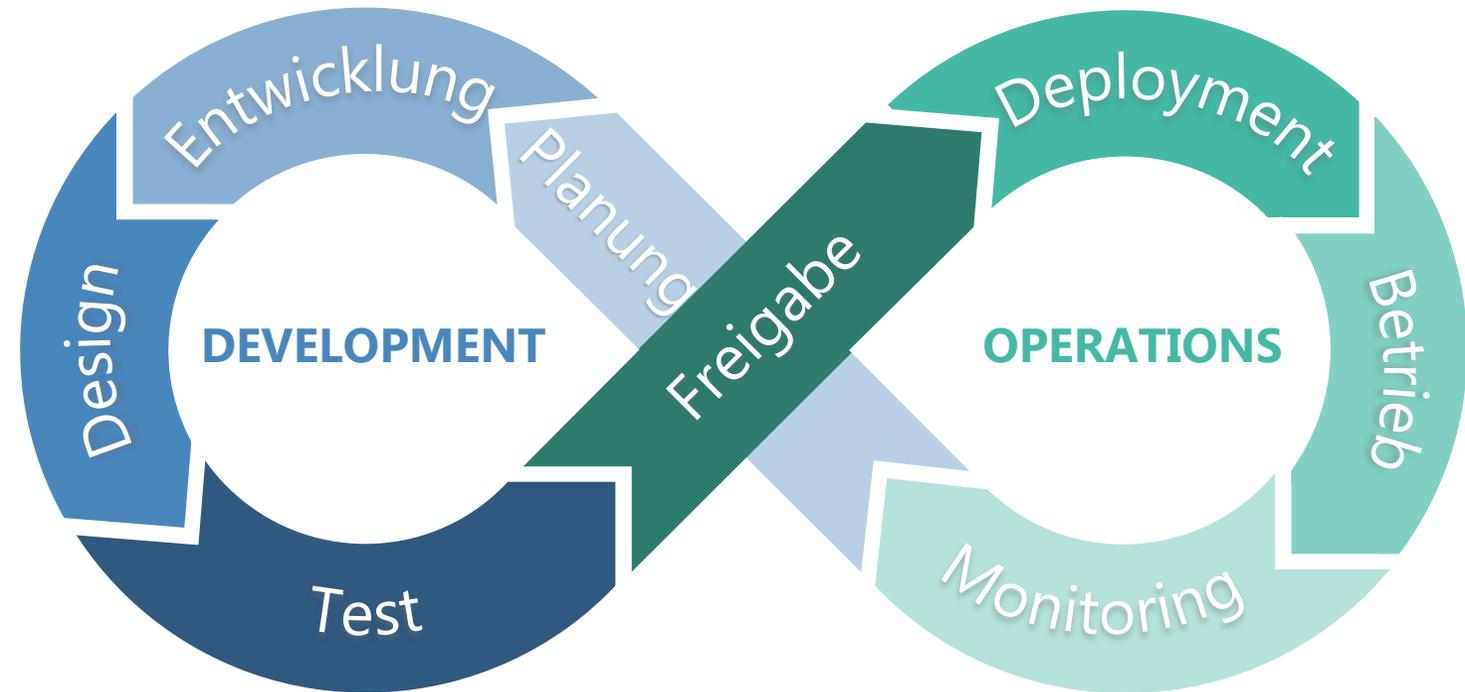
ALM-Tools fungieren dabei im Wesentlichen als Projektmanagement-Tools, mit denen Menschen und Prozesse zusammengebracht werden. Darüber hinaus sollten solche Tools auch den Entwicklungsprozess unterstützen, unabhängig davon, ob die Methodik agil, im Wasserfall oder semi-agil ist. Im besten Fall kann auch das Testmanagement über das gleiche Tool abgebildet werden.

Beispiele für ALM-Tools sind:

- + Atlassian Jira
- + Basecamp
- + CA Agile Central
- + IBM ALM-Lösungen
- + Microsoft Azure DevOps
- + Tuleap

ALM-Tools unterstützen verschiedene Aspekte des Application Lifecycle Managements und müssen nach den spezifischen Anforderungen der eigenen Organisation ausgewählt werden.

## ALM-TOOLS



Mit Application Lifecycle Management Tools, gelingt der Link zwischen Entwicklung über Releasemanagement bis hin zur Ablösung einer Software und sorgt für Transparenz während des gesamten Lebenszyklus eines Systems.

# Kontakt

... wir freuen uns auf Ihre Anfrage:

 **Angelina Schock** | Managing Partner  
schock@gcs-consulting.de

 **Elvira Traub** | Business Consultant  
traub@gcs-consulting.de

 **Anschrift**  
GCS Consulting GmbH  
Firkenweg 1  
85774 Unterföhring

 **Fon** +49 89 891365 -0

 **E-Mail** [info@gcs-consulting.de](mailto:info@gcs-consulting.de)

 **Website** [www.gcs-consulting.de](http://www.gcs-consulting.de)