

LISTICLE

Sieben Bausteine für ein erfolgreiches Change Management in Software-Integrationsprojekten

München, 18. Februar 2021 – Komplexe Integrationsvorhaben in der Softwareentwicklung sind oft zeitaufwendig. IT-Dienstleister Consol nennt sieben Best Practices, mit denen Unternehmen Change-Management-Prozesse in Software-Integrationsprojekten beschleunigen können.

Die Lead Time for Changes ist der wichtigste Messindikator für ein erfolgreiches Change Management in der Softwareentwicklung. Diese Lead Time beschreibt dabei die Zeitdauer, die erforderlich ist, um eine Source-Code-Änderung ab einem neuen Commit erfolgreich im Produktivbetrieb nutzen zu können.

Basierend auf den Erfahrungswerten aus zahlreichen Software-Integrationsprojekten hat Consol sieben Best Practices aufgestellt, die ein erfolgreiches Change Management maßgeblich fördern.

1. Uneingeschränkte Zugriffsmöglichkeit auf die Infrastruktur

Verzögerungen bei der Lead Time for Changes sind oft auf infrastrukturelle Gründe zurückzuführen: So ist zum Beispiel ein Team für den Applikationsbetrieb in der Produktiv-Umgebung verantwortlich, während ein anderes Team für das Betriebssystem und Netzwerkthemen wie Firewall-Einstellungen zuständig ist. Geteilte Verantwortlichkeiten finden sich zudem häufig bei Testumgebungen oder der CI- und CD-Plattform, sodass bei Problemen oder geplanten Änderungen immer auf Beistellungen anderer Teams zurückgegriffen – und gewartet – werden muss.

Best Practice: Die Testumgebungen müssen unkompliziert für alle Teammitglieder nutzbar sein. Dabei sollten Mitarbeiter mit Betriebsverantwortung einen möglichst vollen Root-Zugriff auf die Test- und Produktivumgebung haben: von der Hardware über das Betriebssystem und Drittanbieter-Software bis zur Applikation selbst.

2. Cloud-native Architektur

Große Server-Applikationen benötigen viel Zeit für die Kompilierung, Paketierung und Installation. Automatisierte Regressionstests dauern oftmals stundenlang und werden in vielen Projekten deshalb nur nachts ausgeführt.

Will ein Unternehmen aber Software-Updates häufig – mehrmals am Tag und schnell, etwa innerhalb von 15 Minuten – in Produktion bringen, muss die Architektur dies auch unterstützen. Einerseits muss dabei das Risiko von Seiteneffekten so gering wie möglich sein, andererseits müssen auch der Build-, Test-, Deployment-, Startup- und gegebenenfalls der Fallback-Prozess schnell ablaufen. Und Nutzer der Applikation sollten nach Möglichkeit keine Unterbrechungseffekte wie eine Downtime bemerken.

Best Practice: Eine Microservice-Architektur mit kleinteiligen und voneinander unabhängigen Cloud-nativen Applikationen erlaubt eine fundamentale Beschleunigung. So kann der technische Anteil an der Lead Time for Changes von mehreren Stunden auf wenige Minuten verkürzt werden, und zwar durch einen Umstieg von Komplettinstallationen auf inkrementelle Mini-Updates.

3. Funktionierende DevOps-Organisation

Das Sprichwort „Viele Köche verderben den Brei“ trifft auch auf das Change Management zu. Hinsichtlich der Lead Time for Changes gilt vor allem auch: Sie verlangsamen die Prozesse.

Best Practice: DevOps-Teams, in denen alle Kompetenzen vom Software Engineering über die Qualitätssicherung und das Installations- und Konfigurationsmanagement bis hin zum Applikationsbetrieb gebündelt sind, können autark und ohne Warten auf Beistellungen alle Änderungen und Optimierungen selbst durchführen. Durch den permanenten Wissensaustausch sind Interessen- oder Kapazitätskonflikte vermeidbar.

4. Agiles Mindset

Wenn nicht jedes Team-Mitglied nahezu jede Tätigkeit ausüben kann, sind auch Verzögerungen der Lead Time for Changes nicht ausgeschlossen.

Best Practice: Mit agilen Konzepten werden die Zusammenarbeit innerhalb von Teams und die Kompetenzen der Mitarbeiter gestärkt. Zu den wichtigen Methoden aus der agilen Welt

zählen etwa ein Taskboard, Refactorings, Heartbeat-Retrospektiven, Pair Programming oder Peer Reviews.

5. Null-Fehler-Toleranz

Vorhandene Fehler oder auch schon der Verdacht, dass potenzielle Fehler vorhanden sind, bremsen die Bereitschaft, Software häufig und schnell zu aktualisieren – gemäß dem Motto „Never change a running system“. Umgekehrt fördert aber eine dauerhaft nachgewiesene Fehlerfreiheit und Zuverlässigkeit das Vertrauen bei Software-Aktualisierungen, und zwar das Vertrauen aller Beteiligten vom Anwender über den Softwareentwickler bis hin zum Projekt-Sponsor.

Best Practice: Jedes Unternehmen sollte der Null-Fehler-Toleranz eine hohe Priorität einräumen. Alle vorhandenen Fehler müssen beseitigt werden und die korrekte Funktionsweise ist durch automatisierte Tests mit maximaler Abdeckung abzusichern. Durch Code-Generierung, -Reviews und -Refactorings kann redundanter, fehlerhafter oder überflüssiger Code vermieden werden. Die fehlerfreie Umsetzung von Änderungen wird so deutlich beschleunigt.

6. Automatisierung von Interaktionen innerhalb des Lead-Time-Intervalls

In der gesamten Prozesskette der Lead Time for Changes sind vor allem die menschlichen Interaktionen die größten Zeitfresser: Es beginnt vielfach schon beim Zusammenführen von Source-Code-Änderungen. Zudem sind manuelle Schritte häufig bei der Installation und Testausführung oder bei Konfigurations-Änderungen anzutreffen. Manuelle Tätigkeiten verursachen nicht nur zeitliche Verzögerungen, sondern beinhalten auch die Gefahr menschlicher Fehler, die unnötige Reparaturaufwände nach sich ziehen können.

Best Practice: Durch Nutzung einer Trunk-basierten Entwicklungsmethode liegen zum Zeitpunkt der Release-Entscheidung alle Änderungen schon fix und fertig vor. Sie müssen damit nicht noch manuell zum Beispiel in einen Release-Branch übertragen und geprüft werden. Eine CI-/CD-Pipeline kann zudem alle Schritte wie die Release-Paketierung, Testinstallationen oder die Produktiv-Inbetriebnahme vollautomatisiert ausführen. Jede menschliche Interaktion, die automatisiert wird, vermeidet unnötige Wartezeiten innerhalb der Lead Time for Changes und entlastet zudem das gesamte Team.

7. Schlanke Change-Management-Prozesse

In manchen Fällen sind menschliche Interaktionen unverzichtbar und auch nicht automatisierbar, etwa bei der fachlichen Überprüfung und Qualitätssicherung einer umgesetzten Änderung. Allerdings gibt es auch hier Möglichkeiten, manuelle Schritte durch Automatisierung zu unterstützen und zu beschleunigen.

Best Practice: Alle Change-Management-Prozesse werden dahingehend verschlankt, dass manuelle Schritte nach Möglichkeit nur noch für kontrollierende oder explorative Tätigkeiten erforderlich sind. Testergebnisse können beispielsweise automatisiert so aufbereitet werden, dass alle Änderungen seit dem letzten Testlauf hervorgehoben und schnell überprüfbar sind.

Automatisierung reduziert Lead Time for Changes enorm

„Unter den richtigen Voraussetzungen können in Software-Integrationsprojekten individuelle Kundenanforderungen innerhalb einer Stunde umgesetzt und Änderungen produktiv in Betrieb genommen werden“, erklärt Christian Wied, Teamleiter Software Engineering bei Consol. „Durch automatisierte Test- und Installationsroutinen etwa kann die Lead Time for Changes vielfach auf 15 Minuten reduziert werden. Unter zusätzlicher Berücksichtigung von menschlichen Interaktionen wie Abstimmungs-, Review- oder Quality-Assurance-Prozessen ist damit eine Lead Time for Changes von unter 60 Minuten realisierbar.“

**Dieses Listicle und Bildmaterial sind im
Internet abrufbar unter www.pr-com.de/companies/consol**

Über Consol

Die Consol Consulting & Solutions Software GmbH betreut seit mehr als 30 Jahren Kunden aller Branchen bei nationalen und internationalen IT-Projekten. „Wir unternehmen IT“ ist dabei das Credo, auf dessen Basis die Spezialisten, Umsetzer und Innovationstreiber bei Consol passgenaue IT-Lösungen für den gesamten Software-Lifecycle erarbeiten: High-End IT-Beratung, Software Engineering, IT Operations und DevOps sind die Kernkompetenzen des 1984 gegründeten Unternehmens mit Hauptsitz in München.

Die technologischen Schwerpunkte liegen unter anderem auf Software-Architektur, Cloud-native, CI/CD, Testautomatisierung und Monitoring. Consol verfolgt hierbei einen agilen Arbeitsansatz und nutzt unter anderem Open-Source-Lösungen. Darüber hinaus entwickelt und vertreibt das Unternehmen die Software Consol CM, eine Plattform zur Digitalisierung von Geschäftsprozessen.

Consol ist Red Hat Premier Partner und unterhält strategische Partnerschaften zu AWS oder Microsoft Azure. Zu den Kunden zählen Großunternehmen wie Haribo, Daimler oder Telefónica.

Die Faszination der Consol-Mitarbeiter für technologische Herausforderungen bildet die Basis des Unternehmenserfolgs. Aktuell beschäftigt Consol rund 260 Mitarbeiter an seinen Standorten München, Düsseldorf, Wien, Krakau, Dubai sowie San Francisco.

Weitere Informationen unter <https://www.consol.de>, <https://cm.consol.de> und <https://labs.consol.de> sowie auf Twitter unter https://twitter.com/consol_de.

Pressekontakt

ConSol Consulting & Solutions Software GmbH

Isabel Baum

St.-Cajetan-Straße 43

D-81669 München

Fon: +49-89-45841-101

Fax: +49-89 45841-111

E-Mail: Isabel.Baum@consol.de

Web: <https://www.consol.de>, <https://cm.consol.de> und <https://labs.consol.de>

PR-COM GmbH

Nicole Oehl

Sendlinger-Tor-Platz 6

D-80336 München

Fon: +49-89-59997-758

Fax: +49-89-59997-999

E-Mail: nicole.oehl@pr-com.de

Web: www.pr-com.de