

## **PRESSEMITTEILUNG**

### **Consol stellt Techniken für die iterative Evolution von Softwarearchitekturen auf den Prüfstand**

**München, 18. Juni 2019 – Die Modernisierung von Altsoftware stellt viele Unternehmen vor große Probleme. Die Entwicklung und Implementierung einer komplett neuen Software ist oft nicht der beste Weg. Als Alternative stehen Techniken zur iterativen und inkrementellen Evolution vorhandener Softwarearchitekturen zur Verfügung. Der IT-Dienstleister Consol stellt gängige Verfahren mit ihren jeweiligen Vor- und Nachteilen vor.**

In den meisten Unternehmen findet sich in der Systemlandschaft eine Vielzahl von Altsoftware. Sie durch neue Software im Rahmen eines Big-Bang-Szenarios zu ersetzen, ist oft nicht einfach beziehungsweise immer mit erheblichen Risiken verbunden. Um diese Systeme dennoch zu modernisieren und für künftige Anforderungen fit zu machen, empfiehlt sich deshalb in den meisten Fällen ein schrittweises Vorgehen – eine iterative und inkrementelle Evolution der vorhandenen Softwarearchitektur. Dafür gibt es mehrere Verfahren, Consol stellt die fünf Techniken Modularisierung, Extraktion, Abstraktion, Abschnüren sowie Klonen und Löschen auf den Prüfstand.

#### **1. Evolution durch Modularisierung**

Bei der Modularisierung werden zunächst Code-Bestandteile der Altsoftware identifiziert, die zu einer gemeinsamen Domäne gehören. Dann werden diese Code-Bestandteile etwa durch Refactoring und Anwendung von Clean-Code-Prinzipien modularisiert. Und zum Schluss wird ein Interface für diese Domäne eingeführt.

Die Identifikation von Domänen und die Modularisierung sind Voraussetzungen, um die Altsoftware zerteilen und gezielt modernisieren zu können. Der Nachteil bei dieser Vorgehensweise ist, dass die einzelnen Domänen oft eng mit anderen Bestandteilen der Architektur verknüpft sind. Um erfolgreich zu modularisieren, ist deshalb ein gutes Verständnis der Domäne sowie des Codes der Altsoftware erforderlich.

#### **2. Evolution durch Extraktion**

Bei der Extraktion werden unabhängige Domänen innerhalb der Altsoftware identifiziert und in ein eigenes System extrahiert. Anschließend wird das System der nun eigenständigen Domäne verbessert, gegebenenfalls auch unter Nutzung der Technik „Abstraktion“, um das System oder Teile davon neu zu entwickeln. Letzter Schritt ist die Verbesserung der Altsoftware, die nun weniger Aufgaben hat.

Dieses Vorgehen ist nur möglich, wenn es innerhalb der Altsoftware unabhängige Domänen gibt. Der Nachteil dabei ist, dass ein bestehender Code wiederverwendet wird. Dadurch können „versteckte“ Abhängigkeiten zwischen dem neuen eigenständigen System und der Altsoftware entstehen.

### **3. Evolution durch Abstraktion**

Bei der Abstraktion wird in einem ersten Schritt ein externes, „ideales“ Interface für die Altsoftware definiert und die Kommunikation mit der Altsoftware über eine Fassade oder ein Fassadensystem gekapselt. Nach der Umstellung aller Clients auf das neue Interface erfolgt die iterative und inkrementelle Erstellung besserer neuer Software für das Interface. Anschließend kann die Altsoftware schrittweise durch die neue Software ersetzt werden.

Dieses Vorgehen ist nur möglich, wenn sich eine klare Schnittstelle für bestehende Services definieren lässt. Der Nachteil: Falls sich das Interface zu sehr an der Altsoftware orientiert, werden Fehler nicht korrigiert.

### **4. Evolution durch Abschnüren**

Zunächst wird ein „schlechter“ Code-Teil innerhalb der Altsoftware identifiziert, der ersetzt werden soll. Anschließend wird ein Switch (oder Dispatcher, Proxy, Load Balancer) eingefügt, der auf den schlechten Teil der Altsoftware weiterleitet. Nachdem eine neue Implementierung für den schlechten Teil geschrieben ist, wird die Last vom Switch übergangsweise sowohl auf den schlechten Teil der Altsoftware als auch auf die neue Implementierung weitergeleitet. Schließlich erfolgt eine zunehmende Reduzierung der Last auf dem schlechten Teil der Altsoftware und eine Erhöhung der Last auf der neuen Implementierung bis der schlechte Teil der Altsoftware nicht mehr verwendet wird.

Voraussetzung für dieses Vorgehen ist, dass sich der schlechte Teil der Altsoftware, der ersetzt werden soll, beispielsweise über ein Interface abgrenzen lässt. Der Nachteil ist, dass sich das neue System an der Interaktion der anderen Systeme mit der Altsoftware orientie-

ren muss. Und diese anderen Systeme nutzen eventuell nicht dokumentierte Eigenschaften der Altsoftware oder sind abhängig von nicht spezifiziertem Verhalten und etablierten Workarounds.

## 5. Evolution durch Klonen und Löschen

Beim Klonen und Löschen werden die Anwendergruppen identifiziert und die gesamte Altsoftware für jede Anwendergruppe geklont. Dann wird in jedem Klon alles gelöscht, was die jeweilige Anwendergruppe nicht benötigt. Gegebenenfalls werden Gemeinsamkeiten extrahiert, besser ist aber eine Code-Duplizierung, um zu starke Abhängigkeiten zu vermeiden. Schließlich erfolgt eine Softwareoptimierung für jede Anwendergruppe.

Voraussetzung für dieses Vorgehen ist ein eigenes (Produkt-)Team pro Klon. Der Nachteil dabei ist, dass die Klone via Datenbank für lange Zeit gekoppelt sind.

„Modularisierung, Extraktion, Abstraktion, Abschnüren sowie Klonen und Löschen sind vielfach bewährte Techniken, um vorhandene Altsysteme schrittweise in eine moderne, zukunftssichere Softwarearchitektur zu transformieren“, erklärt Christoph Ehlers, Technical Lead Software Engineering beim IT-Dienstleister Consol in München. „Die Nutzung einer bestimmten Technik hängt aber immer von den jeweiligen Gegebenheiten und Anwendungsfällen ab. Eine generelle Empfehlung kann man nicht aussprechen, es ist immer eine Einzelfallentscheidung zu treffen.“

**Diese Presseinformation und Bildmaterial sind im  
Internet abrufbar unter [www.pr-com.de/consol](http://www.pr-com.de/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> und <https://cm.consol.de> sowie auf Twitter unter [https://twitter.com/consol\\_de](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](mailto:Isabel.Baum@consol.de)

Web: <https://www.consol.de> und <https://cm.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](mailto:nicole.oehl@pr-com.de)

Web: [www.pr-com.de](http://www.pr-com.de)