

# Regelgestützte Validierung von Benutzereingaben

**In Softwaresystemen mit hoher Benutzerinteraktion** können die Anforderungen an die Validierung der Benutzereingaben sowie an die Navigation zwischen einzelnen Masken zu Beginn der Implementierung oftmals nicht klar spezifiziert werden. Erst im

Rahmen von Tests erhalten die Endanwender ein Gefühl für das spezifizierte Verhalten – nicht selten resultiert dies in Anforderungsänderungen, die eine aufwändige Anpassung des Quellcodes nach sich ziehen.



## Ausgangssituation und Problemstellung

### Ausgangssituation

Durch die Einführung von regelgestützten Maskenvalidierungen können Anwendungssysteme implementiert werden, bevor die konkrete Validierungslogik der einzelnen Bausteine bekannt ist. Die Definition und Anpassung von Validierungsregeln kann dann im laufenden Betrieb durchgeführt werden.

Der nachfolgende Quelltext-Ausschnitt zeigt eine klassische Programmlogik zur Prüfung von Nutzereingaben:

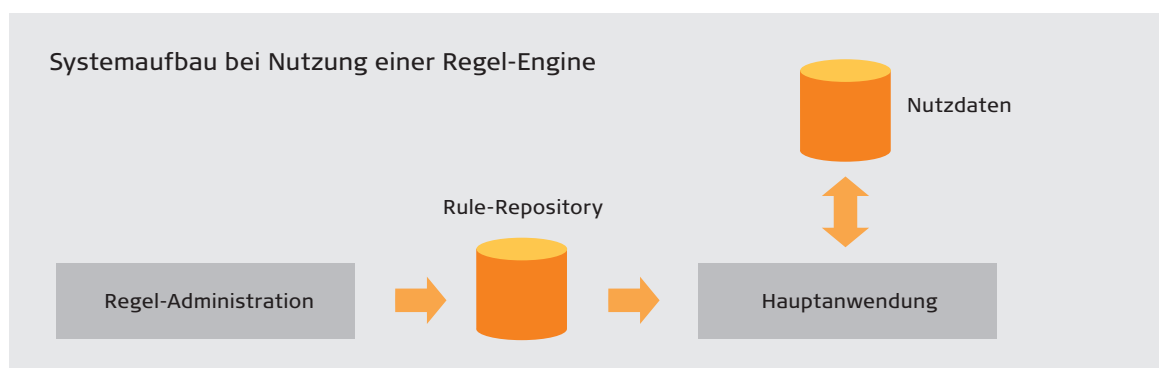
```
if (kreditbetrag < minKredit){  
    validationMessages.add("Fehler aufgetreten");  
}
```

Abgesehen von der Parametrisierung der Bedingungen muss selbst bei kleinen Änderungen im Anwendungs-Code der gesamte Release-Prozess durchlaufen werden, bevor die Anwendung produktiv genutzt werden kann. Dies wirkt sich negativ auf den Aufwand aus. Zudem wird viel Zeit benötigt, den Release-Prozess zu durchlaufen. Die Time-to-Market-Zeitspanne erhöht sich und reduziert gerade in Systemen mit hohem Anpassungsbedarf (Marketing-Kampagnen etc.) die Wettbewerbsfähigkeit des Unternehmens.

## Lösungen und Kompetenzen

### Lösungen

Die nachfolgende Abbildung zeigt einen Vorschlag für eine Anwendungsarchitektur auf Basis eines Regelsystems. Innerhalb des Rule-Repositorys werden alle Regeln der Anwendung gehalten und administriert. Die Hauptanwendung greift über definierte Schnittstellen (z. B. Webservices) auf das Rule-Repository zu, um die für die Validierung notwendigen Regeln zu laden und auf Basis der genutzten Daten auszuführen. Zur Pflege der Regeln existiert eine Administrationsanwendung, auf die Benutzer mit entsprechenden Rechten zugreifen können.



Durch eine klassische Implementierung können zwar Eingabegrenzen und Einstellungen konfiguriert werden, komplett neue Validierungsregeln sind an dieser Stelle allerdings nur durch Anpassung des Quellcodes möglich, was zu einem Durchlauf des Release-Prozesses führt (siehe oben). Diese Problemstellung ist prädestiniert für die Anwendung von Regelsystemen: Die Syntax von Geschäftsregeln passt zum Bedingungs- und Aktionsteil beim Validieren von Eingaben. Darüber hinaus gehören regelbasierte Systeme wie JBoss Drools oder Visual Rules zur Gattung der dynamisch-gebundenen Sprachen. Diese müssen nicht wie klassischer Quell-Code zum Ausführen kompiliert werden, sondern werden zur Laufzeit interpretiert. Das ermöglicht Änderungen an diesen dezidierten Stellen durchzuführen, ohne die gesamte Anwendung neu auszurollen.

Durch den kombinierten Einsatz von DSL(Domain Specific Languages)-Templating können technische Ausdrücke in fachliche transformiert werden, sodass das obige Beispiel wie folgt aussehen könnte:

```

Wenn Kreditbetrag < MindestKredit
Dann Fehlermeldung ausgeben: „Fehler aufgetreten“
  
```

Dadurch können auch ohne Kenntnis der konkreten Programmiersprache neue Regeln in das Rule-Repository geladen oder bereits existierende Regeln geändert werden.

## Auf einen Blick unsere Kompetenzen

### **Fachliche Kompetenzen:**

- Auf ihre Anwendungsfälle zugeschnittene Definitionen der Domain Specific Languages
- Hohe fachliche Kompetenz in unterschiedlichen Fachgebieten wie Kredit und Wertpapier

### **Technische Kompetenzen:**

- Architektur-Konzeption
- Implementierung und Einführung von Anwendungssystemen auf Basis von Rule-Engines
- Integration von Rule-Engines in die bestehenden IT-Landschaften

### **Methodische Kompetenzen:**

- Agiles Vorgehensmodell beim Einführen neuer Architektur-Paradigmen
- Aufbereitung von Entscheidungsvorlagen und Business-Case-Berechnungen

## Nutzen, Vorteile, Besonderheiten

Cofinpro hat in verschiedenen Referenzprojekten bereits erfolgreich anhand von fachlichen Anforderungen Anwendungsarchitekturen auf Basis von Rule-Engines konzipiert, umgesetzt und eingeführt. Dabei lag der Fokus neben der Realisierung auch auf der Anpassung der beteiligten Prozesse der Systeme. Cofinpro bietet die Kombination aus fachlicher und technologischer Expertise, gepaart mit der Transformation fachlicher Anforderungen in technische Lösungen als Kernkompetenz. Neben der hohen fachlichen Kompetenz in verschiedenen Domänen (z.B. Kredit und Wertpapier) besitzt Cofinpro auch ein besonderes Know-how bei der Konzeption, Implementierung und Einführung von Systemen in bereits bestehende IT-Landschaften.

Fragen zu diesen oder anderen Themen beantworten wir Ihnen jederzeit gerne.

**Cofinpro AG**

Untermainkai 27–28, 60329 Frankfurt am Main

Tel: +49 (0) 69 - 2 99 20 87 60

Mail: [welcome@cofinpro.de](mailto:welcome@cofinpro.de)

[www.cofinpro.de](http://www.cofinpro.de)



**Cofinpro unterstützt Deutschlands führende Finanzdienstleister** bei der Verbesserung von Geschäftsprozessen. Wir transformieren Strategien in Prozesse und implementieren diese Prozesse in der IT. Durch Business Transformation schaffen wir für unsere Kunden entscheidende Wettbewerbsvorteile in einem Finanzmarkt, der sich immer noch sehr schnell verändert und vor spannenden Herausforderungen steht. Unser Erfolgskonzept ist dabei ein ganz besonderes: Wir kombinieren hohe Fach- und Technologiekompetenz mit den Methoden unseres Business Engineering Frameworks – durch diese einzigartige Kompetenzbündelung erreichen wir für unsere Kunden exzellente Ergebnisse: „Finest Processes in Finance“.