



Cözüm Mimarı  
Software AG Türkiye

Onur FENAR  
onur.fenar@softwareag.com

## SOA CORNER

### İş Kuralları Yönetimi (BRM)



**İş kuralları seti; bir organizasyona uygulanan, organizasyonun amacına ulaşabilmesi için gerekli tanımlamaları, operasyonları, kısıtlamaları içerir. Günümüzde iş kuralları yönetimini sağlamak amacıyla bu alanda birçok uygulama geliştirilmiştir. Bu uygulamalar arasında ILOG ve Fair Isaac en yaygın kullanılan çözümlerdir.**

**K**urumsal uygulamaların çözümleme, tasarım, geliştirme ve test aşamalarında giderek yaygınlaşan bir yaklaşım iş kurallarının bir kural motoru ile işletilmesidir. Bu yaklaşım, çözümleme aşamasında sözlü olarak ifade edilen iş kurallarının bir kural veri tabanına, genelde bir kural dili ile yazılmasına dayanır. Yazılımın bakım aşamasında iş kuralları değiştiği zaman, kural veritabanının kolayca güncellenmesi ile bakım işlerinin büyük bir bölümünün kısa süre içinde tamamlanma avantajına rağmen, bu şekilde yazılım geliştirmenin tasarım ve test konularında soru işaretleri bulunmaktadır.

### Neden İş Kuralları Motoru?

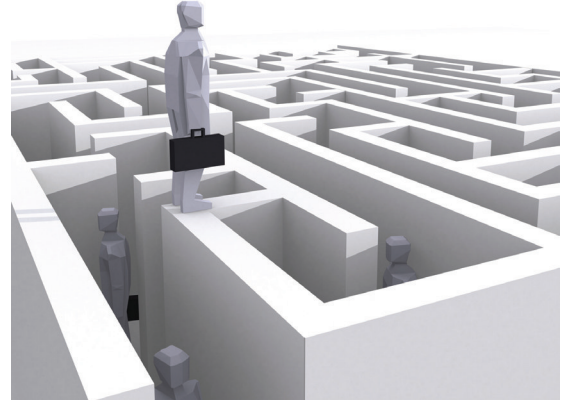
Sıradan bir yazılım sürecini düşündüğümüzde önce yazılımı gerçekleştirilecek olan sistemin analizi yapılır. Bu analiz sonucunda iş mantığının ve sistemin ihtiyaçları belirlenir ve bu ihtiyaçların karşılanabileceği tasarım fazına geçilir. Tasarım sonucunda belirlenen yapıda yazılım yapıldıktan sonra test edilir ve ortaya çalışan bir ürün çıkmış olur. Burada bir noktaya dikkatinizi çekmek istiyorum. Analizin yapıldığı süre ve ürünün teslim edildiği süre yazılımın boyutu büyüdükçe artmakta ve bu süre zarfında sürekli olarak ihtiyaçlar değişmektedir. Sizden istenen yazılım, siz o yazılımı teslim edene kadar pek çok güncellemeden / revizyondan geçer. Hatta ürün teslim edildikten sonra da ihtiyaçlar sürekli olarak değişmekte ve eski ihtiyaçlara göre yazılan uygulamalar bazı durumlarda kullanılamaz hale gelmektedir.

Bu önemli problemin çözümü olarak iş yapış şekli ve kodlamanın birbirinden ayrılması düşünülmüş ve böylece kural motorları ortaya çıkmıştır.

### Problem Tanımı:

- İş kuralları sürekli değişir
- Teslim edilmesi planlanan ürün, güncelliğini yitirir
- Teslimat süresi uzar
- Yeteri kadar test edilemez
- Ürün kalitesiz olur

**Çözüm Önerisi:** Kural motorları



### Ne zaman kural motoru uygulanmalıdır?

- İş akışları kod seviyesinde yönetilemeyecek kadar karışıkça:  
İş akışları karışık olan sistemlerde, bu akışların kodlama seviyesinde yönetilmesi durumunda sistemin geliştirilmesi ve test edilmesi iş akışının karmaşıklığı nedeniyle hem uzun zaman almakta hem de iş akışının her hangi bir aşamasında yapılan ufak bir değişikliğin, sistemin geri kalanında yaratacağı etkileri gözlemlemeyi olanaksız hale getirmektedir. Bu nedenle iş akışı karışık olan sistemlerde kural motoru kullanılması önerilmektedir.

### İş akışı karışık olan sistemlerde kural motoru kullanılması önerilmektedir.

- İş yapış şekli sürekli değişiyorsa: İş kuralları sürekli değişen sistemlerde, değişen her bir iş kuralı için yeniden kodlama yapılması gerekmekte ve sürekli olarak yeni sürümler edinme ihtiyacı ortaya çıkmaktadır. Sistemin kararlılığının (stability) sağlanması açısından bu tür sistemlerde kural motorları kullanılmalıdır.
- Değişikliklerin hemen uygulamaya alınması gerekiyorsa: İş kuralları değiştiğinde, bu değişikliğin sisteme hemen yansıtılması imkansızdır. Yeni kuralı gerçekleştirebilmek için yeni kodlar yazılmalı ve yeni sürüm test edilmelidir. Bu da kritik sistemlerde işi yavaşlatmaktadır. Bu tür sistemlerde kural motoru kullanmak zaman kayıplarını engeller.

### Kural motoru kullanılması sonucunda;

- İş yapış şekli tanımları kolayca anlaşılır
- İş yapış şekli programlama dillerinden bağımsız olur
- İş yapış şekli bilgisini kodlama mantığından ayırır
- Kaynak kod değiştirilmeden akış değiştirilebilir
- İş yapış şeklindeki değişiklikler hızlıca uygulamaya konur