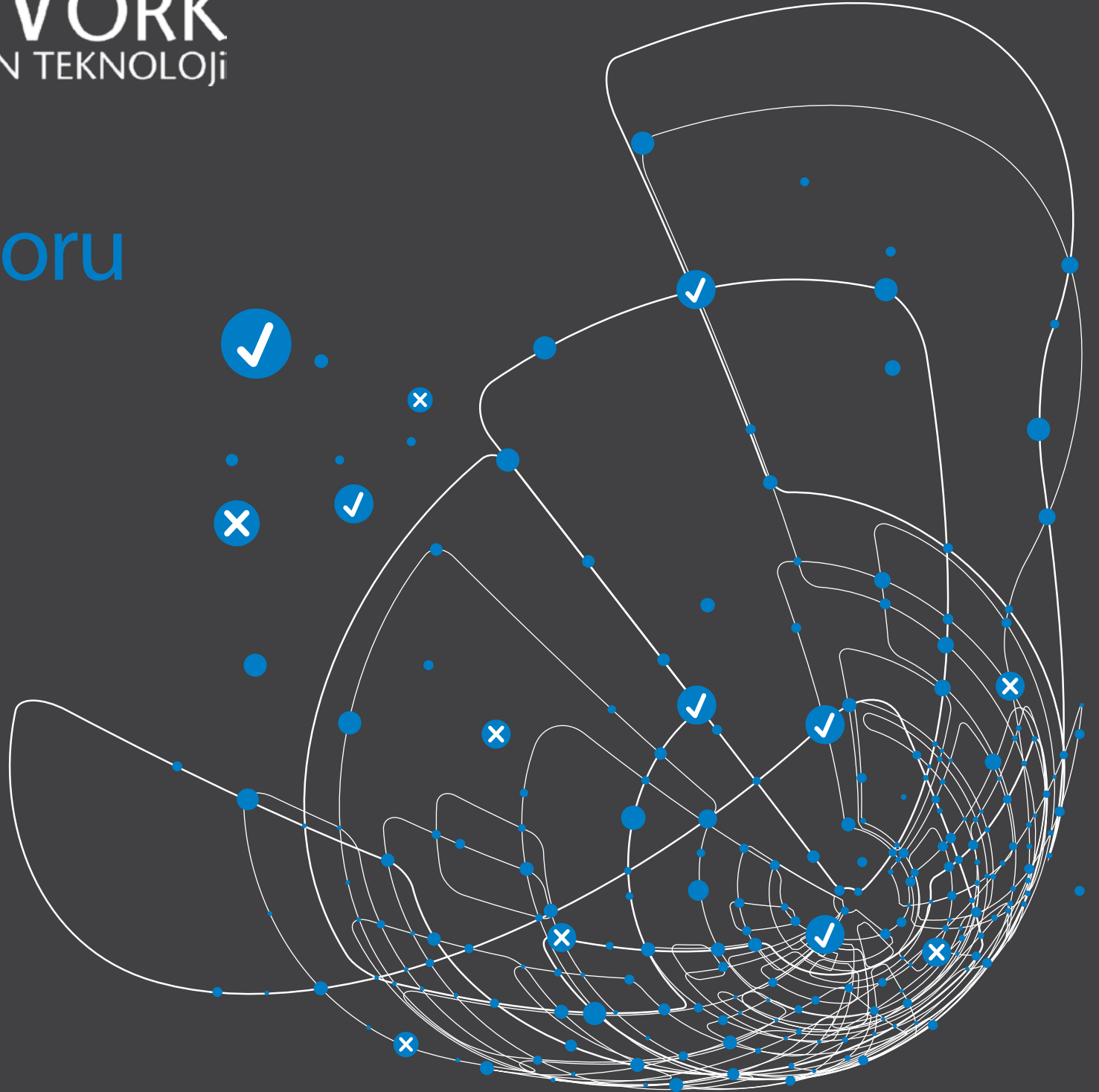


# Kural Motoru





# İş Kuralı Nedir?

Tüm işletmeler kural merkezli çalışırlar. Kurallar hangi fırsatların takip edildiğini veya geride bırakıldığını belirler. Kurallar fiyatlandırmanın nasıl yapılacağını, kimin onay veya ret yetkisi olduğunu, hattaha kimin işe alınabileceğini belirler. Bilgi, malzeme, para akışları, projelerin nasıl ilerleyeceği hep kurallar aracılığı ile belirlenir.

Kısacası kurallar işin nasıl yapılacağına dair yönergeler sağlar. Basit evet/hayır cevaplarından kompleks, çoklu kritere göre skor tabanlı cevaplar üretebilir.

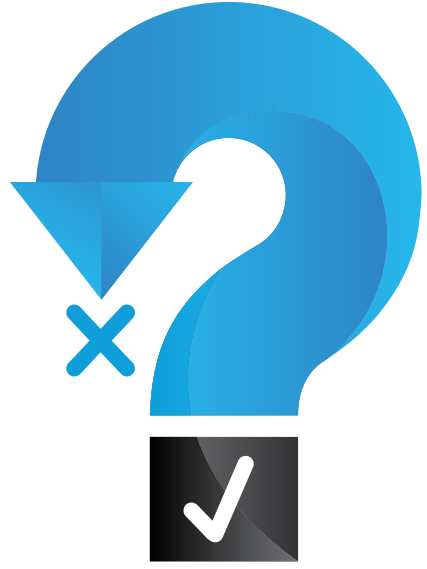
Her işimizi kurallar ile yaptığımız için soru “Benim kurumumun kurallara ihtiyacı var mı?” dan çok “Benim kurumum iş kurallarını otomatikleştirmeden ne faydalar sağlayabilir?” olmalıdır.

# İş Kuralı Örnekleri



Aşağıda iş kurallarına çeşitli örnekler verilmiştir;

- Sözleşmenin tipine, bedeline, tedarikçiye veya ürüne göre sözleşmeleri kim/kimler imzalayabilir?
- Müşteri alış/veriş esnasında arama tarihçesine göre, sipariş geçmişine göre hangi ürünler tavsiye edilmelidir? Veya hangi indirimlere hak kazanmıştır?
- Yapılan işlemin tutarına veya miktarına göre ne tür ek işlemler yapmak gerekir?
- Hangi birim zamanda, ne kadar işlem yapan müşteri “Özel Müşteri” olarak kabul edilir?
- Çözüm ortakları, bayiilerin işlem komisyonları nasıl hesaplanır?
- Hangi veri geçerli bir veridir?
- Hangi işlemler Genel Müdür seviyesinde onaya çıkmalıdır?
- Hangi durumlarda çift yönetici onayına ihtiyaç duyulur?

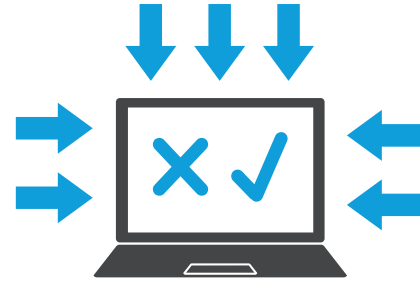


# Kural Motoruna neden ihtiyacım var?

İş akışlarında ve elektronik formlarda yapılan kontroller, bazı sabit değişkenler, iş mantığı merkezi bir yapıda toplanmalıdır. PaperWork Kural Motoru tüm bu bilgilerin merkezi olarak yönetilmesini sağlar. Daha sonra kuralda yapılacak değişiklik ile formlar ve iş akışlarında değişiklik yapmadan tüm sistemin yeni parametrelere göre çalışması sağlanır.

## Merkezi

Kurum içerisindeki tüm kurallar merkezi tek bir yapı üzerinde toplanabilir. Bu şekilde Kurumun tüm kurallarının daha şeffaf olması ve tüm kurallar bazında optimizasyon yapılabilmesi sağlanır.

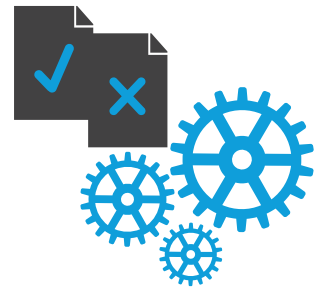


## Takip edilebilirlik

Merkezi bir hale getirilen kurallar daha kolay takip edilir. Kurallarda yapılacak değişikliklerin nereleri etkileyeceği önceden ön görülebilir.

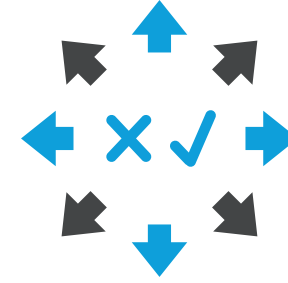
## Tekrar kullanılabilirlik

Kuralların merkezi hale gelmesi ile beraber kurumun değişik birimlerinde, süreçlerinde, elektronik formlarında kullanılan ihtiyaç durumunda tekrar kullanılabilir. Böylece kurum içerisindeki işleyiş standartlaştırılabilir.



## Entegre test ortamı

Entegre test ortamı sayesinde oluşturulan kurallar anında çalıştırılabilir. Daha üretim ortamına geçmeden olası ek ihtiyaçlar, değişiklikler, hatalar anında görülerek gerekli önlemler alınabilir.

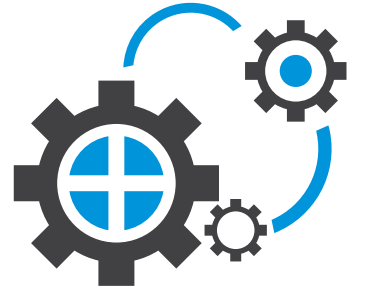


## Bakım Kolaylığı

Kuralların tek bir noktada toplanması ile beraber bakım ve destek kolaylığı sağlanır. Olası değişikliklerde tek bir kuralı değiştirerek bağlı olan tüm fonksiyonların da değişmesi sağlanır.

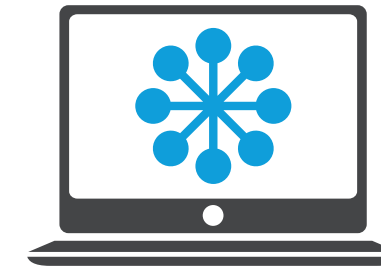
## İş Akışı ve Elektronik Form Desteği

Oluşturulan her kural birden fazla iş akışında ve elektronik formda kullanılabilir. PaperWork Kural motoru tüm PaperWork ürünleri ile tam entegre çalışır.



## SDK Desteği

Tüm PaperWork ürünlerinde 2 tip geliştirme katmanı bulunmaktadır. Bunlardan birincisi .NET kütüphaneleri ile sağlanır. PaperWork ara yüzleri de bu kütüphane ile geliştirilmiştir. Aynı fonksiyonlar SOAP WEB Servisleri ile de sağlanmaktadır. Kurum içerisindeki tüm uygulamalar bu katmanlardan birisi kullanılarak Kural Motoru fonksiyonlarına erişebilir. Böylece kurum içerisindeki tüm uygulamaların tek bir kural motoru ile yönetilebilmesi sağlanabilir.



# Kural Çeşitleri Nelerdir?

PaperWork Kural Motorunda mevcut kurallarınızın tanımlanabilmesi için 4 tip kural çeşidi bulunmaktadır.

## Global Değerler

Tüm akışlarda ve elektronik formlarda kullanılan, zaman içerisinde değişiklik gösterebilecek değerlerdir. Yazı, sayı, para, mantık ve tarih/saat veri tiplerinde olabilir. Örneğin hesaplamalarda kullanılan KDV oranı gibi.

```
if (!Current.Instance.RuleEngine.KDV((string)OTV_KDV.Value))
{
    MSG_VALIDATION="'KDV' oranı uygun değil!";
    return false;
}
```



## Listeler

Kural Motorunda tanımlanan listeler varlık kontrollerinde kullanılırlar. Örneğin 'İş Akışının belli bir adımında akıştaki değer liste değerlerinden birisine eşit ise şu adıma git' gibi tanımlamalar yapılabilir. Liste değerleri için de yazı, sayı, para, mantık ve tarih/saat değerleri kullanılabilir. Aşağıdaki örnekte süreç içerisindeki müşterinin VKN'si kontrol edilerek özel müşteri olup olmadığına karar verilir.

```
1 if (RuleEngine.inList(RuleEngine.Musteriler,FormData.getAttribute("VKN")))
2   MusteriList="OK";
3 else
4   MusteriList="NOTOK";
```

## Tablolar

PaperWork Kural Motorunda MS Excel'de olduğu gibi tablo oluşturmak mümkündür. Amaç verilen satır veya sütun değerlerine göre geçerli veriyi alarak kullanmaktır. Sınırsız sayıda kolon ve satırdan oluşabilir. Tablo içindeki hücre değerleri yazı, sayı, para, mantık ve tarih/saat alanlarından oluşabilir.

Örnek olarak faiz tablosu verilebilir. Şekilde görüldüğü gibi TL,USD,Euro bazında 3,6,9 ve 12 aylık faiz oranları girilmiştir.

| Rows | UcAylık | AltıAylık | DokuzAylık | OnikiAylık |
|------|---------|-----------|------------|------------|
| TL   | 15      | 13        | 9          | 7          |
| USD  | 9       | 8         | 6          | 5          |
| EURO | 7       | 6         | 5          | 4          |

## Fonksiyonlar

Karmaşık olan ve tanımlamalar ile uyarlanamayacak kurallar bütünleşik kod yazma ortamında C# kullanılarak yazılırlar. Bu fonksiyonlar .NET standart kütüphanelerini, PaperWork geliştirme katmanını da kullanabilir. Kısacası PaperWork ara yüzlerinde bulunan tüm fonksiyonlar da kullanılarak kurallar yazılabilir.

Örneğin aşağıda Faiz oranları tablosu kullanılmış, verilen gün değerine ve para formatına göre ilgili faiz oranı alınmıştır.

```
Parametre: currency   Tipi: string   Dönüş: [OK] [+] [F]

int vDay
string currency

1 string dayRange=getDayRange(vDay);
2 object v=inTable(FaizOranlari, dayRange, currency);
3 if(v==null)
4 return -1;
5 return (Double)v;
```

|   |   |
|---|---|
| ✓ | ✗ |
| ✗ | ✓ |
| ✗ | ✓ |



**GGSoft**

YOUR DOCUMENT PARTNER