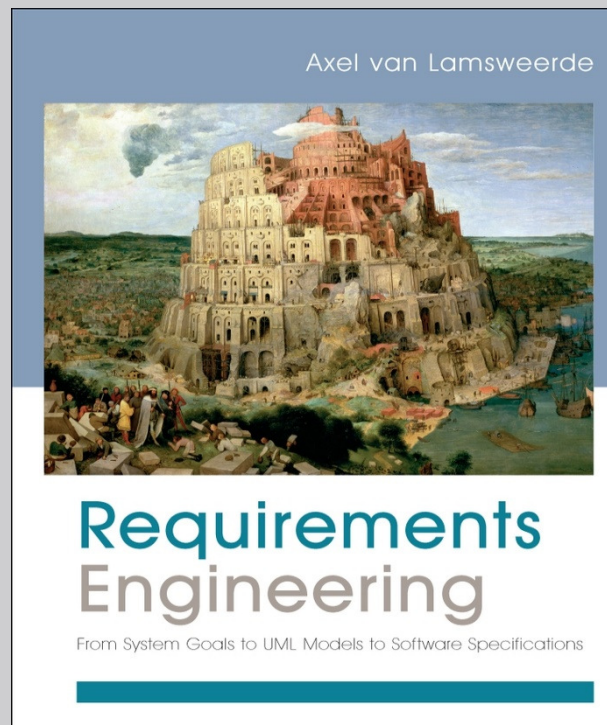


# Requirements Engineering

From System Goals  
to UML Models  
to Software Specifications



Axel Van Lamsweerde

# Hazırlık Aşaması (Devam)



- ◆ Gereksinimler Mühendisliği Nedir? (Devam)
  - RE' de kullanılan bildirim (statements ) tipleri :
    - tanımlayıcı (descriptive)
    - öngörmeli - kurallı (prescriptive)
  - Gereksinimlerin sınıflandırması
    - functional ve. fonksiyonel olmayan gereksinimler

## Bildirimler (Statements)

- ◆ Tanımlayıcı -niteleyici (Descriptive ) bildirimler sistemin ne tür bir davranış gösterdiğine bağlı olmadan geliştirilecek sistemin özelliklerini içerir (bildirme kipi - indicative mood).
  - Doğal kanunlar (natural law) , fiziksel sınırlayıcılar (physical constraint) gibi tanımlanabilir.
  - “Trenin kapıları kapalı ise , açık değildir
  - “Trenin ivmesi pozitif ise, bir hıza sahiptir»”
- ◆ Kurallı -emredici (Prescriptive ) bildirimler gerçekleşmesi istenen özellikleri içerir. Bu bildirimler de sistemin gösterdiği davranışa bağlı değildir (istek kipi- optative mood).
  - “Tren hareket halinde iken kapılar daima kapalı kalacaktır”

# Bildirimler (Statements)

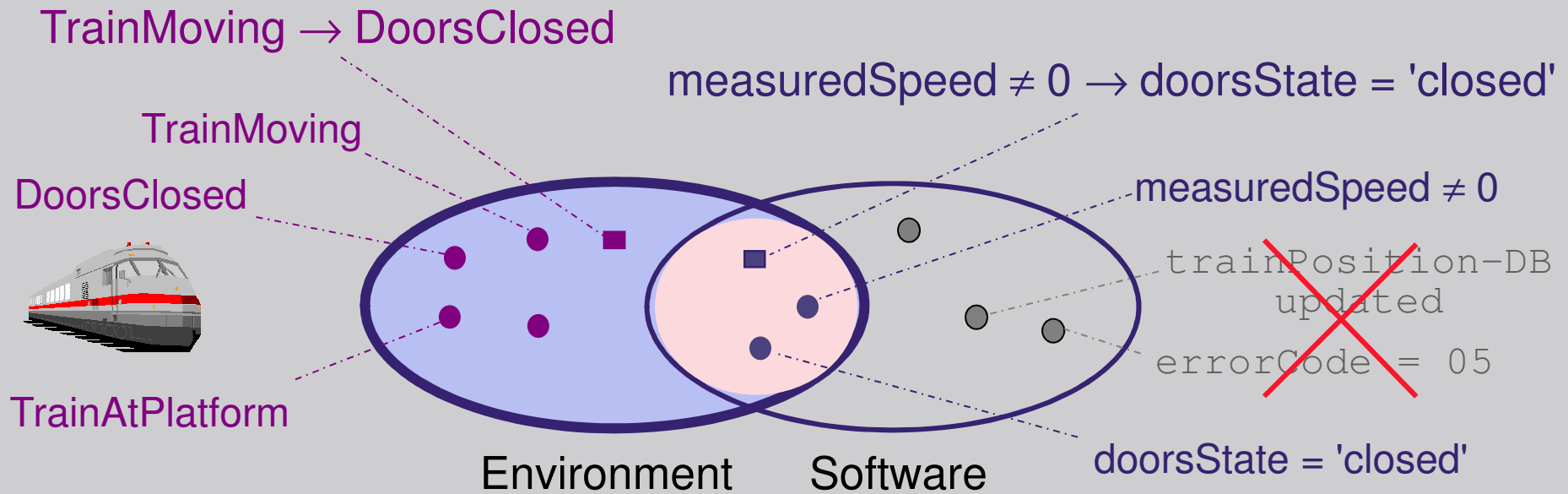
RE 'de tanımlamalı ve kurallı bildirimler arasındaki önemli fark

- ◆ **Kurallı bildirimler (prescriptive statements)**  
ihmal edilebilir (negotiated),  
daha zayıf cümlelere dönüştürülebilir (weakened),  
başka cümlelerle değiştirilebilir (replaced by alternatives)
- ◆ **Tanımlayıcı - niteleyici bildirimler (descriptive statements)**  
yukarıda yapılabilenlerin hiç birini gerçekleştiremez.

# Bildirimler kapsamlarına göre deęişebilir (Statements may differ in scope)

- ◆ Bir RE bildirimini bir olaya referans vermelidir. Bu:
  - Sadece çevresel olabilir (owned by the environment) ya da
  - Çevre ve software-to-be arasında olabilir :

Çevre kontrol edilirken , dięer taraftan ekranda da görüntülenmesi örnektir



# Bildirim tipleri

## sistem gereksinimleri, yazılım gereksinimleri

- ◆ Sistem gereksinimi: : kurallı (*prescriptive*) bir bildirimdir .  
Paylaşılması gerek yoktur ve çevre olayına işaret eder.
  - software-to-be yapısı ile birlikte diğer sistem bileşenlerini kullanır yürütür.
  - Sözcüksel olarak ifade edildiğinde tüm ayrıntılar verilmiş olur.

TrainMoving → DoorsClosed

- ◆ Yazılım gereksinimi : kurallı (*prescriptive*) bildirimdir.  
Paylaşılan olaylara işaret eder.
  - Sadece software-to-be tarafından yürütülür
  - Yazılı geliştiricilerin terimleri cinsinden belirlenir.

measuredSpeed  $\neq$  0 → doorsState = 'closed'

## Deyim tipleri sistem gereksinimleri, yazılım gereksinimleri

- ◆ Bir yazılım gereksinimi aynı zamanda bir sistem gereksinimidir.
- ◆ Bir sistem gereksinimi aynı zamanda bir yazılım gereksinimi değildir.

# Deyim Tipleri

## domain properties, assumptions, definitions

- ◆ Alan Özelliği (Domain property): dünya fenomeni ile ilgili tanımlayıcı (*descriptive*) bildirimlerdir ve bu bildirimler herhangi bir software-to-be 'ye bağlı  $\text{trainAcceleration} > 0$   
 $\rightarrow \text{trainSpeed} \neq 0$

- ◆ Varsayımlar(Assumption): software-to-be 'nin çevresi tarafından sağlanmış olan deyimdir
  - Çevre fenomenleri cinsinden ifade edilir
  - Genellikle kurallı deyimlerdir (sensorler ya da alıcılar (actuators) ile ilgilidir)

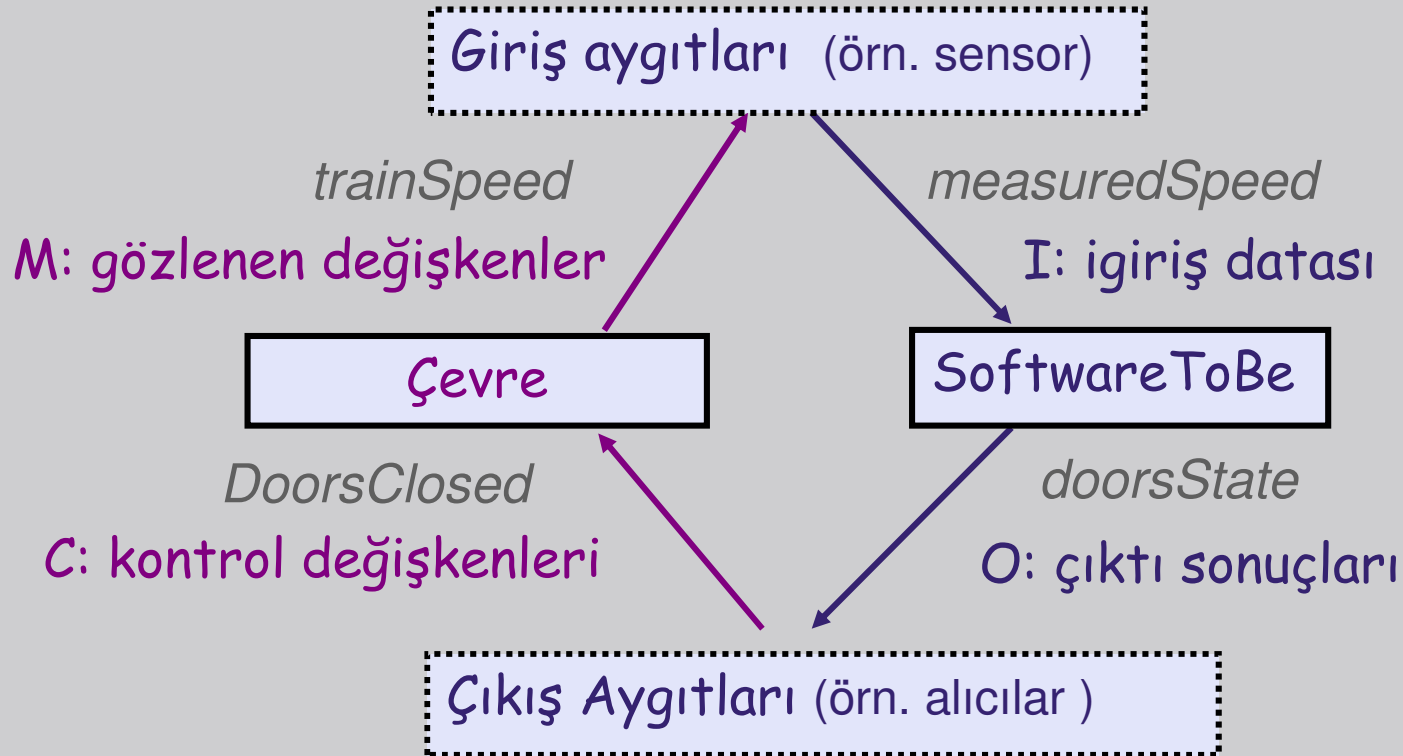
$\text{measuredSpeed} \neq 0$  iff  $\text{trainSpeed} \neq 0$

- ◆ Tanım(Definition): sistem kavramlarına ya da yardımcı terimlere kesin bir anlam sağlayan deyimdir
  - Doğruluk( *truth*) değeri yoktur

“measuredSpeed» trenin hız ölçeri tarafından tahmin edilen hızdır.



# yazılım gerek. nin sistem gerek. ile ilişkisi : 4-değişken modeli [Parnas95]



$SysReq \subseteq M \times C$  gözlenmiş çevre/kontrol edilmiş değişkenlerin ilişkisi

$SofReq \subseteq I \times O$  yazılım giriş/çıkış değişkenlerinin ilişkisi

$SofReq = Map(SysReq, Dom, Asm)$  bağıntısında alan özellikleri ve varsayımlar kullanılarak  $SysReq$ 'in dönüşümü gerçekleşir

# sistem gerek.'inin yazılım gerek.'lerine dönüşümünde argümanlar doğru eşleşmelidir

SOFREQ, ASM, DOM  $\models$  *SysReq*

"SOFREQ 'nin içerdiği yazılım gereksinimleri içerisinde ASM varsayımları ve DOM alan özellikleri bulunuyorsa, istenilenlerin tümü uygun (doğru) olarak gerçekleştirilmiştir. Böylece sistem gereksinimleri , *SysReq* belirlenmiştir " .

( Yukarıdaki biçimsel betimleme bir **if-then** cümlesidir)

SofReq:  $\text{measuredSpeed} \neq 0 \rightarrow \text{doorsState} = \text{'closed'}$

ASM:  $\text{measuredSpeed} \neq 0$  iff  $\text{trainSpeed} \neq 0$   
 $\text{doorsState} = \text{'closed'}$  iff  $\text{DoorsClosed}$

Dom:  $\text{TrainMoving}$  iff  $\text{trainSpeed} \neq 0$

---

*SysReq*:  $\text{TrainMoving} \rightarrow \text{DoorsClosed}$

# Gereksinimlerin Genel Sınıflandırması

## Fonksiyonel ve Fonksiyonel olmayan Gereksinimler

- ◆ Fonksiyonel gereksinimler (Functional requirements) : software-to-be 'nin hangi hizmetleri (servisleri) sağlayabileceğini (should cümleleri) belirtir.
  - Planlanan yazılımın çevre üzerindeki etkilerinin çıkartılması, uygulanabilirlik koşulları
  - Kodlama (yazılımsal operasyonlar olarak adlandırılabilir) ile gerçekleştirilen küçük parçaların (birimlerin-units) fonksiyonelliği

“Yazılım tüm trenlerin hızlarını kontrol edecektir.”

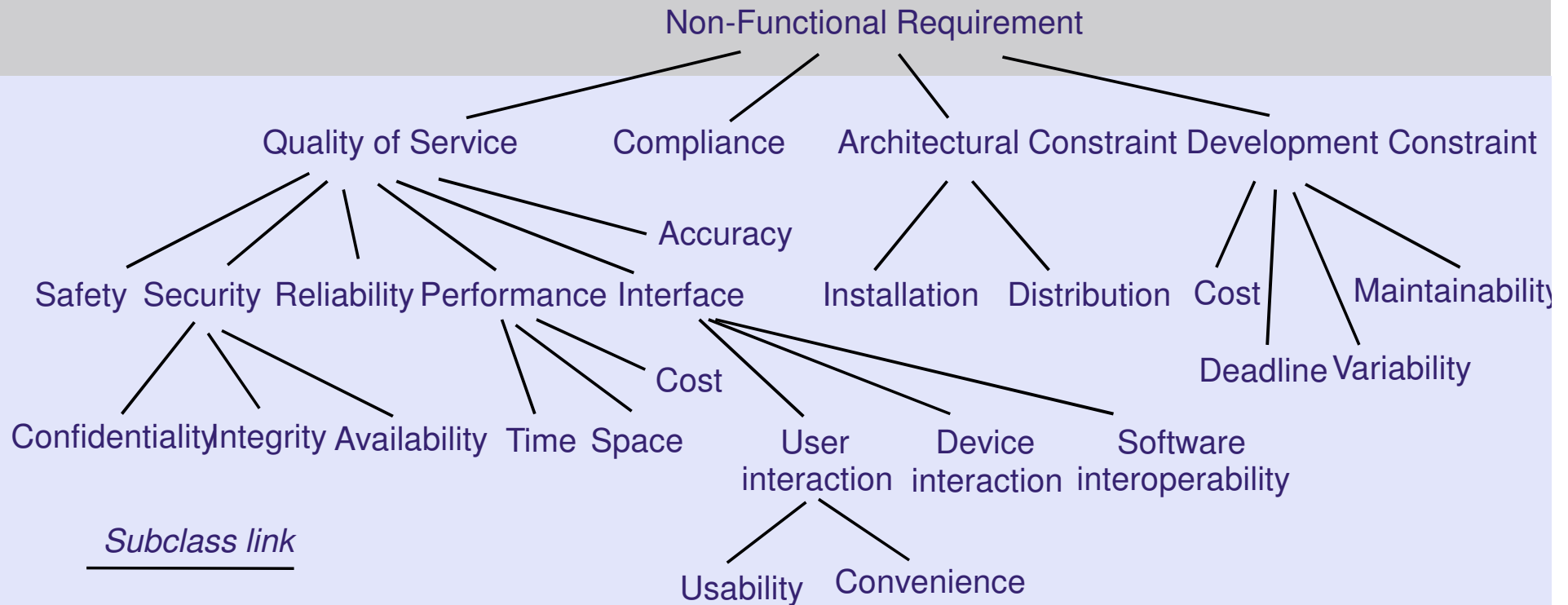
# Gereksinimlerin Genel Sınıflandırması

## Fonksiyonel ve Fonksiyonel olmayan Gereksiniler

- ◆ **Fonksiyonel olmayan gereksinimler (Non-functional requirements):** Hizmetlerin (servislerin) nasıl sağlanabileceğine (should be provided) ilişkin kısıtlardır (constraints)
- ◆ **Nitelik gereksinimleri:** güvenlik (safety), güvenlik(security), doğruluk(accuracy), zaman/uzay performansı (time/space performance), kullanılabilirlik (usability), ...

"Hız komutları trenlerin herbirine her 3 saniyede gönderilecektir."

# fonksiyonel olmayan gereksinimlerin taxonomisi



# Fonksiyonel Olmayan Gereksinimler ile ilgili genel özellikler

- ◆ Fonksiyonel olmayan gereksinimlere keskin sınırlar konulamaz, çakışmaları mümkündür.
- ◆ **Fonksiyonel /fonksiyonel olmayan gereksinim çakışması:**  
Güvenlik duvarı yönetimi (firewall management) için belirlenen fonksiyonel gereksinimler, aynı zamanda güvenlik (security) ile de ilgilidir.
- ◆ **Fonksiyonel olmayan gereksinimlerin de çakışması:**  
“tren bildirimlerinin yüksek frekansı (high frequency of train commands)” fonksiyonel olmayan gereksinimler olarak  
hem performans  
hem *güvenlik (safety)*  
kriterleri olarak betimlenebilir.

## «Security and Safety» Fonksiyonel Olmayan Gereksinimleri

- Güvenliğin sağlanması durumu (**security assurance case**) belirlenmiş bir dizi kriterin gerçekleşmesini gerektirir. Böylece sistem kendi güvenlik özelliklerine göre önceden belirlenmiş spesifik beyanları sağlayacaktır (ISO 15026)
- Kanıtların, argümanların ve varsayımların biçimsel olarak betimlenmesidir. Böylece bir sistem ya da bir ürün güvenlik gereksinimlerini (**safety requirements**) sağlayacaktır (Yellow Book)

# «Security and Safety» Fonksiyonel Olmayan Gereksinimleri

- ◆ **Safety requirements:** Nitelik (Quality) gereksinimleridir.

Olumsuz durumlarla sonuçlanabilecek (might) (kaza, bir parçanın aşınarak bozulması , çevreye zarar gibi) faktörlerin yazılıma etkisine ait bildirimlerdir.

Trenlerin kontrol edilebilen hızları en kötü koşulda durma mesafesini (worst-case stopping distance) ard arda giden trenler arasında her zaman koruyacaktır (shall cümlesi)

- ◆ **Security requirement** : Nitelik (Quality ) gereksinimleridir. Sisteme ait nesnelerin istenmeyen çevre koşullarına göre korunması ile ilgili kuralları belirler (prescribe)

Örnekleri farklı kategorilerde verilir.



# Güvenlik (Security) Gereksinimlerinin ayrıntılı örnekleri

- ◆ **Gizlilik Gereksinimleri (Confidentially req.)** Yetkisi olmayanların hiçbir şekilde erişemeyeceği bazı kritik bilgilerin olması durumudur.

Hangi kitabın kimin tarafından alındığını personel dışında sistemi kullanan hiç kimse bilemez .

- ◆ **Gizlilik Gereksinimleri (Privacy Req. )** Bazı özel bilgiler o bilginin sahibinin onayı olmadan açılmaz.

Herhangi bir katılımcının günlük kısıtları kendisinin onayı olmadan diğer katılımcılar tarafından görüntülenemez.

# Güvenlik (Security) Gereksinimlerinin ayrıntılı örnekleri

## ◆ Bütünlük ( tümlük, tamamiyet) Gereksinimleri (Integrity Req.)

Bazı bilgiler ancak doğru yapıldığında ve izin ile (authorization) değiştirilebilir.

Kitap kopyalarının iadesi doğru olarak ve yalnızca kütüphane personeli tarafından kodlanacaktır.

## ◆ Geçerlik gereksinimleri (Availability Req.) Bazı bilgiler ve kaynaklar gerekli olduğu zamanda ve durumda ve kullanımına izin verildiğinde (authorized) kullanılabilir

Trenin bulunduğu pozisyonlar ile ilgili bilgiye acil durum istasyon bilgisayarında her zaman ulaşılacaktır .

# Güvenilirlik Gereksinimleri (Reliability Requirements) fonksiyonel olmayan gereksinim

- ◆ Yazılımın işlevliğinin (güncel olarak kullanılabilmesinin) uzun süre devam etmesi beklenir. Yazılımın gerçekleştirecekleri de, bazı istisnaların olması olasılığına rağmen, doğru (correct) ve güçlü (robust) olarak sağlanmalıdır.

Tren sevkiyat kontrol yazılımının *güncel olarak (işlevsel olarak) kullanılabileceği ortalama bir süre* arızalar da göz önüne alındığında  $10^9$  saattir.

## Doğruluk Gereksinimleri (Accuracy req.)

- ◆ Nitelik gereksinimi olarak yazılım tarafından işlenen bilginin (information) durumunu kısıtlar. Böylece ortamdaki fiziksel bilgiye ait durumun doğru olarak aktarımı gerçekleşir.

1.Tren kontrol birimi tarafından kullanılan tren pozisyonları hakkında bilgi, en çok X metreye kadar olan trenlerin gerçek konumunu doğru olarak yansıtacaktır.

2.Kitabın bir kopyasının alınması , yazılım sistemi tarafından sadece ve sadece kitap kütüphanede (raflarda) mevcutsa mümkün olacaktır.

3.Toplantı çizelgesinde kullanılan kısıtlamalar davet edilen katılımcıların gerçek kısıtlamalarını doğru bir şekilde yansıtmalıdır

## Performans Gereksinimleri

- ◆ Nitelik gereksinimleri olarak yazılımın operasyonel koşullarındaki kısıtlarını , zaman ya da mekan (time or space) bakış açısından belirler. Bunlar

aktivitelerin sıklığı (periyodu),

iş çıkarma yeteneği yani verim (throughput) ,

giriş / çıkışların büyüklüğü

olabilir.

1. Her bir trene her 3 saniyede ivme (acceleration) talimatı verilir.

2. Toplantı düzenleyen kişi paralel olarak 3 oturum gerçekleştirebilir.

3. Özgeçmiş sorgularının dönüşü 2 saniyeden az olacaktır.

## Arayüz Gereksinimleri (Interface Req.)

- ◆ software-to-be ve çevre (environment) tarafından paylaşılan fenomenin kısıtlandığı nitelik gereksinimidir.

yazılım ile çevrenin statik ve dinamik etkileşimlerini,  
giriş /çıkış formatlarını ,

çevrenin gerçekleştireceği bir dizi etkileşimlerin sağlamasını hedefler.

İnsan ile etkileşimde kullanılabilirlik (usability) gereksinimleridir.

1.Sorunsuz ve güvenli olarak trenlerin çalışmasını sağlamak için, trene gönderilen iki ivmelenme (accelaration) komutu arasındaki süre en fazla X olmalıdır.

2. Görüşmeye katılacakları yormamak için, davetli katılımcılar arasındaki görüşme sayısı olabildiğince az sayıda organize edilmelidir .