

## Bölüm 3

### Gereksinimlerin Ortaya Çıkarılması

*Bu bölümde, gereksinimlerin bulunabileceği, keşfedilebileceği, ele geçirilebileceği veya zorlanabileceği birçok yolu keşfedeceğiz. Bu içerikteki bu terimlerin hepsi eş anlamlıdır , ancak toplama ile ortaya çıkarma eş anlamlı değildir.*

*Gereksinimler anî bir şekilde ortaya çıkmamışlardır.*

*Gereksinimler genellikle kolay değildir.*

*Daha detaylı ve karmaşık olanların çoğu süreklilikle ve titizlikle gözden geçirilmelidir.*

*Farklı kullanıcı ve paydaş sınıfları için gereksinim belirlemeyi yürütmek için seçebileceğiniz birçok teknik vardır. Muhtemelen sizde herkes gibi bu tekniklerden yararlanacaksınız.*

*Bu teknikler aşağıdakiler gibidir:*

- Beyin Fırtınası
- Kart sıralama
- Tasarımcı olarak çırak geliştirme(JAD)
- Alan analizi
- Etnografik gözlem
- Hedefe dayalı yaklaşımlar
- Grup çalışması
- Röportajlar
- İçe bakış
- Ortak uygulama
- Merdiven
- Protokol analizi
- Prototipleme
- Kalite işlevi dağıtımı

### GEREKSİNİMLERİN ORTAYA ÇIKARILMASI İÇİN HAZIRLIK

Tüm müşterileri ve paydaşları belirlemek, gereksinimlerin ortaya çıkarma hazırlığının ilk adımıdır. Ancak paydaş, gruplar ve müşteriler homojen olmayabilir, bu nedenle her alt gruba farklı davranmanız gerekir. Örneğin, evcil hayvan mağazası (POS) sistemi için kullanıcının farklı alt sınıfları şunları içerir:

- Cashiers(kasiyerler)
- Managers(yöneticiler)
- System maintenance personel(sistem bakım personeli)
- Store customers(mağaza müşterileri)
- Inventory/warehouse personel(stok/depo personeli)
- Accountants (to enter tax information)(muhasebeciler(vergi bilgilerini girmek için))
- Sales department (to enter pricing and discounting information)(satış departmanı(fiyatlandırma ve indirim bilgilerini girmek için))

Bu kullanıcı alt gruplarının her birinin farklı eksiklikleri vardır ve bunların belirlenmesi gerekir.

Bu nedenle, seçime hazırlanmak için süreç şu şekildedir:

- Tüm müşterileri ve paydaşları belirleyin.
- Müşterileri ve diğer paydaş gruplarını ilgi alanlarına, kapsamlarına, yetkilendirmelerine veya diğer ayırt edici faktörlere göre sınıflara ayırın (bazı sınıfların birden çok bölümlere düzeyine ihtiyacı

olabilir).

-Her kullanıcı sınıfı ve paydaş için bir şampiyon veya temsili grup seçin

-Her sınıftan veya paydaş grubundan ilk girdileri almak için uygun teknikleri seçin.

Kullanıcı sınıfı bölümlenimin başka bir örneği: Bagaj taşıma sistemi için aşağıdakiler de dahil olmak üzere birçok farklı paydaş bulunmaktadır.

-Travelers(gezinler)

-System maintenance personnel(sistem bakım personeli)

-Baggage handlers(bagaj işleyicileri)

-Airline schedulers/dispatchers(havayolu zamanlayıcıları/göndericileri)

-Airport personnel(havaalanı personeli)

-Airport managers and policymakers(havaalanı yöneticileri ve politika yapıcıları)

Ama farklı ihtiyaçları olan çeşitli gezginler de var:

Örneğin, aşağıdaki alt sınıfları göz önünde bulundurun:

-Children(çocuklar)

-Senior citizens(yaşlılar)

-Business people(iş adamları)

-Casual travelers(sıradan gezginler)

-Military personnel(askeri personel)

-Civilians(siviller)

-Frequent flyers(sık uçanlar)

Bu alt sınıfların her birine farklı ortaya çıkarma teknikleri ile yaklaşılması gerekebilir.

Örneğin, anketler çocuklar için uygun olmayabilirken, odak gruplar askeri personel için daha az yararlı olabilir.

Bu alt sınıfların çoğu örtüşür.

örneğin, bir kişi hem iş seyahatinde hem de sıradan bir gezgin olabilir.Bu veriler analiz edilirken ve çıkarım yapılırken örtüşmeler dikkate alınmalıdır.

## **Ortaya Çıkarma Teknikleri Anketi**

Şimdi ortaya çıkarma tekniklerini incelemeye başlama zamanı. Bu teknikleri alfabetik sırayla sunuyoruz; herhangi bir tercih ima edilmemektedir. Bölümün sonunda biz Bu tekniklerin farklı durumlarda yaygınlığını ve uygunluğunu tartışacağız.

### **Beyin Fırtınası:**

Beyin fırtınası, müşteriler ve diğer paydaşlarla yapılan resmi olmayan oturumlardan oluşur.

Beyin fırtınası resmileştirilebilir belirlenmiş bir gündemi, tutanak tutmayı ve resmi yapıların kullanımını içerir (örn.Sipariş Kuralları).

Ancak bir beyin fırtınası toplantısının formalitesi muhtemelen bunun tersidir. Bu tür toplantılar muhtemelen gayri resmi, hatta kendiliğinden olmalıdır.

Beyin fırtınası oturumları sırasında bazı ön gereksinimler oluşturulabilir, ancak bu husus ikincildir. JAD tekniği beyin fırtınasını (ve çok daha fazlasını) içerir. Grup odaklı uyarma teknikleri, dolaylı olarak bir tür beyin fırtınası içerir. Beyin fırtınası, misyon veya görev gibi genel amaç belirleme için de yararlıdır.

#### **Kart Sıralama:**

Kullanıcıların nesnelere nasıl gruplamaya eğilimli olduklarını ortaya koyma amaçlı bir araştırma metodudur.

Kullanıcılara belirli öğelerin isimlerinin yazılı olduğu kartlar verilir; bunları ilgili biçimde gruplamaları ve oluşturdukları grupları adlandırmaları istenir. Kullanıcıların bir internet sitesi ya da uygulamanın bilgi mimarisi hakkındaki zihinsel yönelimlerini tanımlamak için kullanılır.

Müşterilerin ve paydaşların kartları tamamlamasına izin veren süre önemli karardır.

Bu süre çok kısa tutulursa paydaşlar işi çok hızlı yapacağından muhtemelen önemli eksik işlemlere yol açacaktır.

Ancak paydaşlara çok fazla zaman vermek süreci gereksiz yere uzatacaktır.

Minimum 1 hafta en fazla 2 haftalık süre dilimi tavsiye edilir. Kartların tamamlanmasına izin verilir.

Diğer bir alternatif ise, müşterilerin kartları 2 saatlik bir oturumda tamamlaması ve ardından 1 hafta sonra başka bir oturum için geri gelmelerini sağlamaktır.

Her durumda, her kart oluşturma oturumundan sonra, gereksinim mühendisi bu kartları bir şekilde düzenler, genellikle işlevleri mantıksal olarak kümeler.

Bu kümeler, belirlenen gereksinimlerin temelini oluşturur. Sıralanan kartlar ayrıca CRC geliştirme sürecine girdi olarak kullanılabilir (yetenek, sorumluluk, nihai kodda program sınıflarını belirlemek için işbirliği) kartları

Bir diğeri

kısaca tartışılacak olan teknik, QFD, bir kart sıralama etkinliğini içerir.

Her kart, işlevselliğin yalnızca kısa bir açıklamasını ve bir öncelik derecesini içerir. Bir önceki slayttaki gibi gereksinim mühendisi bu kart destesini analiz eder ve iki karta karar verir.

Kartlar "müşteri yönetimi" işlevlerine, iki kart "vergi işlevlerine" ve bir kart "envanter özellikleri" veya işlevlerine aittir ve kartları sırayla düzenler.

Müşteriye, düzeltme için bu sıralanmış işlevler listesi veya eksik özellikler gösterilebilir. Ardından, gerekirse yeni bir kart turu oluşturulabilir.

Süreç, gereksinim mühendisi ve müşteri memnun kalana kadar devam eder.

#### **Çırak Olarak Tasarımcı:**

Çırak olarak tasarımcı, gereksinim mühendisinin yeterince öğrenmek için müşterinin "omzunun üzerinden baktığı" bir gereksinim keşif tekniğidir.

Müşterinin ihtiyaçlarını anlamak için yaptığı çalışmalar hakkında. Aralarındaki ilişki müşteri ve tasarımcı usta ile çırak arasındaki gibidir.

Yani, tıpkı gereksinim mühendisinin (tasarımcının) müşterinin işi hakkında müşteriden öğrenmesini istediğimiz gibi, çırak da ustadan bir beceri öğrenir.

Müşterinin bir tür öğretme becerisine sahip olması gerekiyormuş gibi görünebilir.

Bu tekniğin çalışması için, ama bu doğru değil. Bazı müşteriler hakkında konuşamaz etkili bir şekilde çalışırlar, ancak ortaya çıktıkça bunun hakkında konuşabilirler. Ayrıca, müşterilerin onu sunmanın en iyi yolunu ya da güdülerini bulmak zorunda değiller; sadece ne yaptıklarını açıklamalıdır.

Bir çırak bir görevin birden çok örneğini gözlemleyerek çalışma stratejileri ve teknikleri ve varyasyonları dahil ederek, kendi başlarına nasıl yapacaklarına dair bir anlayış oluşturmak. Bu tekniğin çalışması için gereksinim mühendisinin ne iş yaptığını anlaması gerekir. Aşağıdakiler dahil, çalışmanın yapısı ve anlamı:

- İş halletmek için strateji
- Yola çıkan kısıtlamalar
- Çalışmayı desteklediği için fiziksel ortamın yapısı
- İşin bölünmüş şekli
- Tekrarlayan aktivite kalıpları
- Bunların herhangi bir potansiyel sistem üzerindeki etkileri

Tasarımcı işi anladığını ve herhangi bir yanlış anlamının düzeltilebileceğini müşteriye göstermelidir. Son olarak, tasarımcıyı çırak yaklaşımı olarak kullanmak, gereksinimleri keşfetmenin ötesinde başka proje faydaları da sağlar. Örneğin, bu tekniği kullanmak modellenen süreci iyileştirmek için yardımcı olabilir.

### **Etki Alanı Analizi:**

Alan bilgisine sahip olmanın önemini daha önce vurgulamıştık (ister gereksinim mühendisi ve/veya müşteri tarafından) gereksinim mühendisliğinde bulunur.

Etki alanı analizi, tasarlanan sistemle ilgili ve rekabet eden uygulamaların "manzarasını" değerlendirmeye yönelik herhangi bir genel yaklaşımı içerir.

Çok bir yaklaşım, temel işlevlerin belirlenmesinde faydalı olabilir ve daha sonra, eksik işlevsellik. Etki alanı analizi, daha sonra yeniden kullanılabilirliği belirlemek için de kullanılabilir.

### **Etnografik Gözlem:**

Etnografik gözlem, dolaylı gözlemin yapıldığı herhangi bir tekniği ifade eder. ve doğrudan faktörler, gereksinim mühendisinin çalışmalarını bilgilendirir. Etnografik

gözlem, sosyal bilimlerden ödünç alınan bir tekniktir.

insan faaliyeti ve işin gerçekleştiği çevre bilgi vermek için kullanılır.

bilim adamı bazı fenomenlerin çalışmasında. En dar anlamıyla, etnografik

gözlem, bir araç olarak kullanılması için uzun gözlem periyotlarını içerir.

Etnografik gözlemi örneklemek için, bir antropolog farklı kültürleri incelerken aralarında

yaşar,avlanır,uyur,yemek yer toplumun nasıl işlediğine ve inanç sistemlerine ilişkin bilgileri doğrudan kanıtlı bir şekilde alır.

Gereksinimlerin ortaya çıkarılmasına etnografik gözlem uygularken, gereksinim mühendisi müşterinin işyeri kültürüne kendini kaptırır.

Gereksinim mühendisi,otomatikleştirilecek işi veya faaliyeti gözlemlemeye ek olarak,aynı

zamanda,doğrudan iletişim kurulamayan ortamlarda bir sonraki sayfadaki teknikler kullanılır.

- Bir müşteri için akıllı bir ev için gereksinimleri topluyorsunuz.
- Müşteriyle ne yaptıkları hakkında uzun süre röportaj yapmak istiyorsunuz.
- Müşteriler günlerini geçirirken onlarla etkileşim kurmak için zaman harcarsınız ve müşterilere sorular sorun ("bulaşık makinesini neden gece çalıştırıyorsunuz, neden sabah çalıştırmıyorsunuz?)
- Evin kendisinden başka bilgiler edirsiniz - kitaplıktaki kitaplar, duvardaki resimler, mobilya stilleri, hobilerin kanıtı, çeşitli cihazlarda aşınma ve yıpranma vb.

Etnografik gözlem çok zaman alıcı olabilir ve gözlemcinin önemli ölçüde eğitim almasını gerektirir.

Fizikte iyi bilinen bir prensip vardır. Heisenberg belirsizlik ilkesi olarak, ki bu, sıradan kişilerin

terimleriyle şu anlama gelir : Olduğun şeyi etkilemeden bir şeyi tam olarak ölçemezsin

ölçmek. Bu nedenle, örneğin, bir çalışma ortamını gözlemlerken

müşteri, süreçler ve davranışlar değişir çünkü herkes etkilemek için dışarı çıkar - yani bir durumun yanlış resmi oluşur ve hatalı kararlara yol açar.

### **Hedefe Dayalı Yaklaşımlar:**

Hedefe dayalı yaklaşımlar, gereksinimlerin belirlendiği herhangi bir ortaya çıkarma tekniğini içerir.

Hedefe yönlendiren bir dizi yoluyla misyon beyanından kaynaklandığı kabul edilmiştir.

Bu hedefler bir veya daha fazla kez alt bölümlere ayrılabilir.

Alt düzey hedefler belirli üst düzey hedeflere ayrılır

Üst düzey gereksinimler daha düşük düzeyli gereksinimler oluşturmak için kullanılır.

Örneğin,bagaj taşıma sistemi misyon beyanını göz önünde bulundurun:

Hedef yolcunun çıkışa kadar bagaj taşımasının tüm yönlerini otomatikleştirmek

- Hedef 1: Check-in'den check-in'e kadar bagaj takibini tamamen otomatikleştirmek
- Hedef 2: Bagajın check-in kontuarından(check-in için ayrılan bölge) uçağa yönlendirilmesini tamamen otomatikleştirmek.
- Hedef 3: Kayıp bagaj miktarını %1'e düşürmek.

Bu hedefler daha sonra yapılandırılmış bir sistem kullanılarak gereksinimlere ayrıştırılabilir.

hedef-soru-metrik yaklaşımı olan GQM önemli bir tekniktir.

Gereksinim mühendisliği gibi sistem mühendisliğinin bir çok alanında kullanılır.

GQM şunları içerir:

1.adım;sistemin amaçlarını veya hedeflerini belirtin; her hedeften hedefe ulaşıp ulaşılmadığını belirlemek için cevaplanması gereken soruları türetin; ne gerektiğine karar verin.

2.adım;Hedefe ulaşıp ulaşılmadığını belirlemek için cevaplanması gereken soruları çoğalt.

3.adım;Sorulara ve ne gerektiğine karar ver.

Örneğin, bagaj taşıma sistemi söz konusu olduğunda 3.adımı göz önünde bulundurun. Sorular sorun (havaalanı/havayolu/uçuş/zaman aralığı/vb.?)Bu soru formun bir gereksinimini ortaya koymaktır.

### **Grup Çalışması:**

Grup çalışması, çalışma sırasında kullanılan her türlü grup toplantısı için genel bir terimdir.

gereksinimlerin keşfi, analizi ve takip süreçleri. en kutlanan

gereksinimlerin ortaya çıkarılması için grup odaklı çalışmanın ortak uygulama tasarımıdır.

Grup etkinlikleri birçok kişiyi bir araya getirmek açısından çok verimli olabilir.

ancak çatışma ve bölünme potansiyelini riske atar. Başarının anahtarı

her türlü grup çalışmasında grup toplantılarının planlanması ve yürütülmesidir.

İşte grup toplantıları hakkında hatırlanması gereken en önemli şeyler:

- Ödevini yap, kuruluşun tüm yönlerini, sorunları, siyaseti, çevreyi vb. araştırın
- Toplantıdan birkaç gün önce bir gündem yayınlayın (her madde için ayrılan süre ile)
- Elinizde özel bir not defteri bulundurun
- Kişisel sorunların araya girmesine izin vermeyin
- Herkesin sesini duyurmasına izin verin
- İlk fırsatta fikir birliği arayın.
- Gündemdeki tüm maddeler yeterince tartışılana kadar ayrılmayın.

Toplantının kapanışını birkaç gün içinde toplantı tutanaklarını yayınlayın

ve katılımcıların değişiklik önermesine izin verin.

JAD:Bilgi teknolojileri projelerinde tasarım sürecini hızlandırmak için kullanılır. Geliştiriler ve

kullanıcılar (müşteri) bir araya gelerek ortak bir çözüm oluşturmaya çalışırlar

Bu ilkeler,gereksinimlere JAD yaklaşımı ile devreye girecektir.

Her türden grup çalışmasının bir çok dezavantajı vardır.İlk olarak,grup toplantıları organize edilmesi ve birçok paydaşın konulara odaklanmasını sağlamak zordur.

Açıklık ve samimiyet sorunları da ortaya çıkabilir çünkü insanlar her zaman halka açık bir forumda gerçek duygularını ifade etmeye istekli değildir.Herkesin farklı bir kişiliği olduğundan belirli kişiler toplantıya hükmedebilir.

Birkaç kişinin toplantıya sahip olmasına izin vermek diğer çoğu katılımcı için "dışlanmış" olma duygusunu ön plana çıkarır.

Etkili toplantılar yürütmek ve dolayısıyla grup çalışmasını kullanmak son derece gelişmiş liderlik organizasyon ve kişilerarası beceriler gerektirir. Bu nedenler gereksinim mühendisi mümkün olduğunda bu becerileri geliştirmeye çalışmalıdır.

## İNTERVIEWS(Ropörtajlar)

Mülakatlar yoluyla elde etme, iki bireysel paydaş veya küçük bir paydaş grubu (bazen odak grubu olarak da adlandırılır) arasında yüz yüze iletişimi içerir. Mülakatlar, özellikle kullanılabilirlik gereksinimleri olmak üzere paydaşlardan sistem düzeyi gereksinimleri çıkarmak için kullanımı kolay bir tekniktir.

Teşvik faaliyetlerinde kullanılacak üç tür görüşme vardır ve bunlar bireylere veya odak gruplara uygulanabilir:

- Yapılandırılmamış
- Yapılandırılmış
- Yarı Yapılandırılmış

Muhtemelen en yaygın tür olan yapılandırılmamış görüşmeler, doğası gereği konuşma niteliğindedir ve katılımcıları rahatlatmaya hizmet eder. Kendiliğinden bir "itiraf" gibi, bunlar, gereksinim mühendisi ve paydaş bir arada olduğunda herhangi bir zamanda ve herhangi bir yerde ortaya çıkabilir ve bu şekilde bilgi yakalama fırsatı asla kaybolmamalıdır. Ancak görüşmecinin becerisine bağlı olarak, yapılandırılmamış görüşmeler isabetli olabilir veya iskanabilir. Bu nedenle, yapılandırılmış veya yarı yapılandırılmış mülakat tercih edilir.

Yapılandırılmış görüşmeler, doğası gereği çok daha resmidir ve titizlikle planlanmış önceden tanımlanmış soruları kullanırlar. Şablonlar, yapılandırılmış stili kullanarak ara görüntülerken çok faydalıdır. Yapılandırılmış görüşmelerin ana dezavantajı, formatın çok kontrollü olması nedeniyle bazı müşterilerin bilgi vermemesidir.

Yarı yapılandırılmış görüşmeler, yapılandırılmış ve yapılandırılmamış görüşmelerin en iyilerini birleştirir. Yani, gereksinim mühendisi dikkatlice düşünülmüş bir soru listesi hazırlar, ancak daha sonra görüşme sırasında kendiliğinden yapılandırılmamış soruların içeri girmesine izin verir.

Yapılandırılmış görüşmeler tercih edilirken, hangisinin kullanılacağına seçimi çok fırsatçı bir karardır. Örneğin, müşterinin kurumsal kültürü çok resmi olmayan ve rahat olduğunda ve güven yüksek olduğunda, yapılandırılmamış görüşmeler tercih edilebilir. Daha karmaşık, süreç odaklı bir organizasyonda, yapılandırılmış ve yarı yapılandırılmış görüşmeler muhtemelen daha fazla arzu edilir.

İşte üç görüşme türünden herhangi birinde kullanılacak bazı örnek görüşme soruları:

- Sistemin önemli bir özelliğini adlandırın.
- Bu özellik neden önemlidir?
- En önemlisi beş olmak üzere birden beşe kadar bir ölçekte bu özelliği nasıl değerlendirirsiniz?
- Bu özellik diğer özelliklere göre ne kadar önemlidir?
- Başka hangi özellikler bu özelliğe bağlıdır?
- Başka hangi özellikler bu özellikten bağımsız olmalıdır?
- Bu özellik hakkında başka hangi gözlemlerde bulunabilirsiniz?

Hangi görüşme tekniği kullanılırsa kullanılsın, tüm doğru soruların sorulduğundan emin olmak için özen gösterilmelidir. Yani, önemli soruları atlamayın ve gereksiz, rahatsız edici veya gereksiz soruları dahil etmeyin. Kesinlikle gerekli olduğunda telefon, video konferans veya e-posta yoluyla görüşmeler yapılabilir, ancak bu iletişim türlerinde yanıtların bazı önemli noktayı kaybedebileceğinin farkında olun.

## **İNTROSPECTION (İçe Bakış)**

Bir gereksinim mühendisi, müşterinin ne istediğini düşündüğüne dayalı olarak gereksinimler geliştirdiğinde, iç gözlem sürecini yürütüyor demektir. Özünde, gereksinim mühendisi kendini müşterinin yerine koyar ve "müşteri olsaydım sistemin bunu yapmasını isterdim..." görüşündedir.

Gereksinim mühendisinin alan bilgisi müşterininkinden çok daha fazla olduğunda, içe dönük bir yaklaşım yararlıdır. Bazen müşteri, mühendise aşağıdakine benzer sorular soracaktır: "Ben olsan ne isterdin?" İç gözlem, gereksinim mühendisinin etkileşimlerinin her yönünü bilgilendirecek olsa da, bir müşteriye ne istemesi gerektiğini söylememe konusundaki uyarımızı unutmayın.

## **JOİNT APPLICATION DESIGN (Ortak Uygulama Geliştirme)**

Ortak uygulama geliştirme (JAD), sistem kullanıcıları, sistem sahipleri ve uzun bir süre boyunca belirli bir dizi soruna odaklanan analistler ile yüksek düzeyde yapılandırılmış grup toplantıları veya mini geri çekilmeleri içerir.

Bu toplantılar günde 4-8 saat ve 1 günden birkaç haftaya kadar süren bir süre içinde gerçekleşir. JAD, katılımcılar aynı yerde olmadığına çok bölgesel uygulama için bile uyarlanmıştır (Cleland-Huang ve Laurent 2014). Geleneksel olarak büyük hükümet sistemleriyle ilişkilendirilirken projelerde, teknik her boyuttaki sistemlerde endüstriyel ortamlarda kullanılabilir.

JAD ve JAD benzeri teknikler, problemler, hedefler ve gereksinimler üzerinde grup mutabakatı sağlamak için sistem planlama ve sistem analizi faaliyetlerinde yaygın olarak kullanılır. Spesifik olarak, gereksinim mühendisi, JAD oturumlarını aşağıdakiler için kullanabilir:

operasyon tanımı kavramı, sistem hedefi tanımı, gereksinimlerin ortaya çıkarılması, gereksinim analizi, gereksinimler belgesi incelemesi ve daha fazlası.

Bir JAD incelemesi veya denetim oturumu planlaması üç adımdan oluşur:

1. Katılımcıların seçilmesi
2. Gündemin hazırlanması
3. Bir yer seçme

Bu adımların her birini hazırlarken büyük özen gösterilmelidir.

İncelemeler ve denetimler, aşağıdaki katılımcıların bir kısmını veya tamamını içerebilir:

- ✓ Sponsorlar (ör. üst yönetim)
- ✓ Bir ekip lideri (kolaylaştırıcı, bağımsız)
- ✓ Gereksinimlerin ve iş kurallarının sahibi olan kullanıcılar ve yöneticiler
- ✓ Katipler (yani toplantı tutanakları ve not tutanlar)
- ✓ Mühendislik personeli

Sponsor, analistler ve yöneticiler bir lider seçer. Lider kurum içi veya danışman olabilir. Normalde yazılım geliştirme ekibinden bir veya daha fazla yazar (not tutucu) seçilir. Analist ve yöneticiler, kullanıcı topluluğundan bireyler seçmelidir. Bu kişiler kendi iş alanlarında bilgili ve açık sözlü olmalıdır.

Bir oturum planlamadan önce, analist ve sponsor projenin kapsamını belirlemeli ve her oturumun üst düzey gereksinimlerini ve beklentilerini belirlemelidir. Oturum lideri ayrıca sponsor insanların, zamanı ve diğer kaynakları çabalamaya istekli olduğundan emin olmalıdır. Gündem büyük ölçüde yürütülecek incelemenin türüne bağlıdır ve yeterli zamana izin verecek şekilde oluşturulmalıdır. Gündem, kod ve belgeler ayrıca tüm katılımcılara onları gözden geçirmek, yorum yapmak ve soru sormaya hazırlanmak için yeterli zamana sahip olmaları için toplantıdan çok önce gönderilmelidir.

Aşağıda yazılım gereksinimleri, tasarım denetimleri veya kod adım adım ilerlemek için bazı kurallar yer almaktadır. Oturum lideri, bu uygulamaların uygulanmasını sağlamak için her türlü çabayı göstermelidir.

- Gündeme sadık kalın.
- Programa bağlı kalın (gündem konularına belirli bir zaman ayrılır).
- Yazıcının not alabildiğinden emin olun.
- Teknik jargondan kaçının (inceleme teknik olmayan personeli içeriyorsa).
- Çatışmaları çözün (onları ertelememeye çalışın).
- Grup fikir birliğini teşvik edin.
- Kişilere izin vermeden kullanıcı ve yönetim katılımını teşvik edin

oturumu domine etmek.

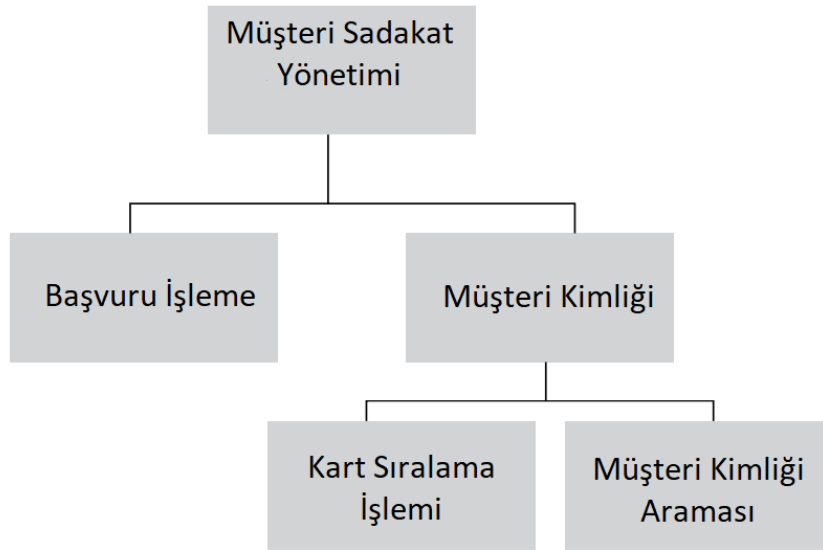
- Toplantıyı kişisellikten ayrı tutun.
- Toplantıların gerektiği kadar uzun sürmesine izin verin.

Herhangi bir gözden geçirme oturumunun son ürünü, tipik olarak, oturum sırasında üzerinde anlaşmaya varılan öğelerin (spesifikasyonlar, tasarım değişiklikleri, kod değişiklikleri ve eylem öğeleri) bir özetini sağlayan resmi bir yazılı belgedir. Belgenin içeriği ve organizasyonu, açık bir şekilde, oturumun doğasına ve amaçlarına bağlıdır. Ancak gereksinimlerin ortaya çıkması durumunda, ana eser SRS'nin ilk taslağı olabilir.

## LADDERİNG(Merdivenleme)

Merdivenlemede, gereksinim mühendisi, gereksinimleri ortaya çıkarmak için müşteriye kısa yönlendirici sorular (sondalar) sorar. Daha sonra yüzeyin altında daha derine inmek için takip eden sorular sorulur. Yanıtlardan elde edilen bilgiler daha sonra ağaç benzeri bir yapı halinde düzenlenir.

Tekniği göstermek için, evcil hayvan mağazası POS sistemi için aşağıdaki merdiven sorularını ve yanıtlarını göz önünde bulundurun. "RE", gereksinim mühendisini ifade eder.



**Figure 3.3 Evcil hayvan mağazası POS sistemi için merdiven şeması**

RE : Sistemin önemli bir özelliğini adlandırın.

Customer: Müşteri kimliği.

RE : Bir müşteriyi nasıl tanımlarsınız?

Customer : Bağlılık kartlarını kaydırabilirler.

RE : Müşteri kartını unutursa ne olur?

Customer : Telefon numarasından aranabilirler.

RE : Müşterinin telefon numarasını ne zaman alırsınız?

Customer : Müşteriler bağlılık kartı başvurusunu tamamladığında.

RE : Müşteriler başvuruları nasıl tamamlıyor?...

Ve bunun gibi...

Figure 3.3, sorulara verilen yanıtların bir merdiven veya hiyerarşik diyagramda nasıl organize edildiğini gösterir.

Merdivenleme tekniği, bilginin hiyerarşik bir şekilde düzenlenebileceğini varsayar veya en azından bilginin hiyerarşik bir şekilde düzenlenmesine neden olur.

### **PROTOCOL ANALYSIS(Protokol Analizi)**

Protokol analizi, müşterilerin gereksinim mühendisleriyle birlikte otomatikleştirecekleri prosedürleri gözden geçirdikleri bir süreçtir. Böyle bir gözden geçirme sırasında müşteriler, atılan her adımın gerekçesini açıkça belirtir.

Örneğin, Bölüm 1'deki etki alanı kelime anlama bölümünde tartışılan büyük paket teslimat şirketi için, yoğun kış tatili sezonunda mühendislerin ve diğer destek profesyonellerinin düzenli teslimat personeliyle birlikte yolculuk yapması bir uygulamaydı. Bu uygulama, yalnızca teslim edilecek paketlerdeki dalgalanmayı ele almakla kalmadı, aynı zamanda mühendisleri, şirketin hizmetleriyle ilgili süreçler ve prosedürler hakkında, gerçekte uygulandıkları şekliyle yeniden tanıştırdı. Mühendisler tarafından sahada yapılan gözlemler genellikle süreç optimizasyonuna ve diğer yeniliklere yol açtı.

Birazdan protokol analizinin çirak olarak tasarımcıya çok benzediğini göreceksiniz, ancak ince farklılıklar var. Bu farklılıklar, protokol analizinde çirak olarak tasarımcıdan daha pasif olan gereksinim mühendisinin rolünde yatmaktadır.

### **PROTOTYPING (Prototipleme)**

Prototipleme, özellikle kullanılabilirlik gereksinimleri olmak üzere yeni özellikleri keşfetmek için sistem modellerinin oluşturulmasını içerir. Prototipleme, gereksinimlerin ortaya çıkarılması için özellikle önemli bir tekniktir. Örneğin, sarmal yazılım geliştirme modelinde yaygın olarak kullanılır ve çevik metodolojiler, esas olarak, giderek daha işlevsel hale gelen bir dizi atılabilir prototipten oluşur.

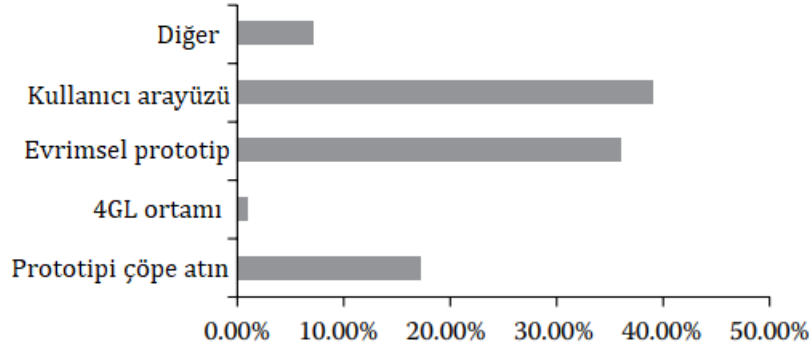
Prototipler, çalışan modelleri ve çalışmayan modelleri içerebilir. Çalışan modeller, yazılım sistemleri ve simülasyonlar durumunda yürütülebilir kodu veya yazılım olmayan sistemler için geçici veya ölçekli prototipleri içerebilir. Çalışmayan modeller, storyboard'ları ve kullanıcı arayüzlerinin maketlerini içerebilir. Bina mimarları, müşteri gereksinimlerini ortaya çıkarmaya ve onaylamaya yardımcı olmak için düzenli olarak prototipleri (örneğin ölçekli çizimler, karton modeller, 3 boyutlu bilgisayar animasyonları) kullanır. Sistem mühendisleri prototipleri aynı nedenlerle kullanır.

Çalışan yazılım prototipleri söz konusu olduğunda, kod kasıtlı olarak atılacak şekilde tasarlanabilir veya kasıtlı olarak yeniden kullanılmak üzere (yapılıp atılmaksızın) tasarlanabilir. Örneğin, grafiksel kullanıcı arayüzü kod maketleri gereksinimlerin ortaya çıkarılması için kullanılabilir ve kod yeniden kullanılabilir. Ve çevik yazılım geliştirme metodolojileri, sürekli gelişen, atılmayan prototiplerin bir sürecini içerir.

Son zamanlarda, 3 boyutlu baskı, belirli sistemlerin fiziksel modellerini oluşturmada önemli bir araç haline geldi. 3-D baskının diğer hızlı prototipleme teknolojilerine göre iki önemli avantajı vardır. Birincisi maliyettir—endüstriyel kalitede 3 boyutlu yazıcılar birkaç bin dolara satın alınabilirken, geleneksel bilgisayar sayısal kontrolünü (CNC) kullanan hızlı prototipleme makineleri birkaç yüz bin dolara mal olabilir. İkinci avantaj, 3 boyutlu yazıcıların girdi standardı olarak alabilmesidir. Yaygın olarak kullanılan bilgisayar destekli tasarım (CAD) programları tarafından üretilen dosyaları biçimlendirin. (Berman 2012).

Prototiplemeyi kullanmanın birkaç farklı yolu vardır; örneğin, dördüncü nesil bir ortamda (yani bir simülörde), kullanılıp atılan prototipleme, evrimsel prototipleme (prototipin nihai sisteme dönüştüğü yer) veya kullanıcı arayüzü prototiplemesi. Bazı kuruluşlar birden fazla prototipleme türü kullanabilir. Kasab vs. (2014), gereksinim mühendisliği ile ilgili çeşitli uygulamaların yaygınlığını belirlemek için yazılım uzmanlarıyla geniş bir anket yaptı. Bu farklı prototipleme teknikleri için seçim sıklığına ilişkin sonuçları Şekil 3.4'te gösterilmektedir.

Gereksinimlerin ortaya çıkarılması için prototiplemeyi kullanırken göz önünde bulundurulması gereken en az üç tehlike vardır.



**Figure 3.4** Yazılım geliştirme yaşam döngüsü metodolojisi boyunca prototip yöntemleri seçimi. (Adapted from Kassab, M., Neill, C & Laplante, P. State of practice in requirements engineering: contemporary data. *Innovations in Systems and Software Engineering*, 10, no. 4 (2014): 235–241. With permission.)

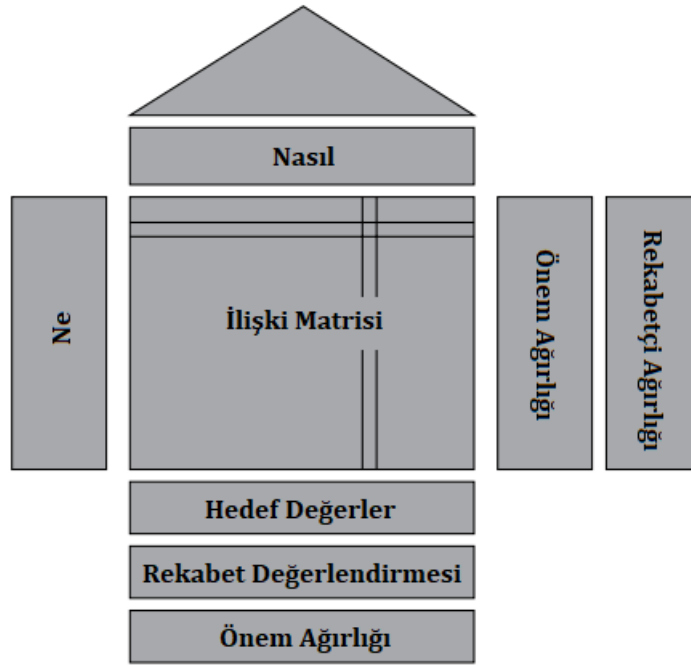
İlk olarak, bazı durumlarda, tutulması amaçlanmayan yazılım prototipleri, aslında program baskıları nedeniyle tutulur. Bu durum potansiyel olarak tehlikelidir, çünkü kod muhtemelen en titiz teknikler kullanılarak tasarlanmamıştır. Kullanılıp atılan prototiplerin istenmeyen yeniden kullanımı, endüstride sıklıkla meydana gelir. İkinci sorun, prototiplemenin belirli işlevsel olmayan gereksinimleri keşfetmede her zaman etkili olmamasıdır. Bu, özellikle yalnızca geçerli standartların ve yasaların analiziyle elde edilebilecek gereksinimler için geçerlidir. (Kassab and Ormandjieva 2014).

Son olarak, kullanıcıların sistemle etkileşim kurma yollarını keşfetmek için prototipler kullanılırken sorunlar ortaya çıkabilir. Ana endişe, kullanıcıların bir prototiple (davranışın sonuçlarının gerçek olmadığı) gerçek sistemle farklı şekilde etkileşime girmesidir. Örneğin, bir çarpışmadan kaynaklanan gerçek bir yaralanma veya hasarın olmadığı bir araç simülöründe kullanıcıların nasıl araç kullanabileceğini düşünün. Sürücüler, simülörde gerçek bir araçta olduğundan çok daha agresif davranabilir ve bu da muhtemelen hatalı gereksinimlerin keşfedilmesine yol açabilir.

#### **QUALITY FUNCTION DEPLOYMENT (Kalite Fonksiyon Yayılımı)**

Kalite fonksiyon yayılımı (QFD), müşteri gereksinimlerini keşfetmeye ve üretim aşaması boyunca kullanılacak ana kalite güvence noktalarını tanımlamaya yönelik bir tekniktir. QFD, müşterilerin ihtiyaçlarının karşılanması için bir yapı sağlar ve istekler dikkatle dinlenir, daha sonra analizden uygulamaya ve dağıtıma kadar doğrudan bir şirketin dahili teknik gereksinimlerine çevrilir. QFD'nin temel fikri, müşteri ihtiyaçları, teknik gereksinimler, öncelikler ve (gerekirse) rakip değerlendirmesi arasında ilişki matrisleri oluşturmaktır. Özünde, QFD, kart sıralama, merdivenleme ve alan analizini içerir. Bu ilişki matrisleri genellikle bir evin çatısı, tavanı ve yanları olarak temsil edildiğinden, QFD bazen "kalite evi" olarak anılır. (Figure 3.5; Akao 1990).

QFD, 1966 yılında Yoji Akao tarafından imalat, ağır sanayi ve sistem mühendisliğinde kullanılmak üzere tanıtıldı. Ayrıca IBM, DEC, HP, AT&T, Texas Instruments ve diğerleri tarafından yazılım sistemlerine uygulanmıştır. “Müşterinin sesi” derken ki kastımız; gereksinim mühendisinin müşteriden neye ihtiyaç duyduklarını empatik bir şekilde müşterinin kendi sözleriyle ifade etmesidir. Müşterinin sesi, ürünlerin yalnızca “mühendis sesinden” geliştirilmemesini sağlamak için tüm analiz, tasarım ve geliştirme faaliyetlerinin temelini oluşturur. Bu yaklaşım, gereksinimlerin ortaya çıkarılmasının özünü somutlaştırır.



**Figure 3.5 QFD's "house of quality" (Akao 1990).**

Aşağıdaki gereksinim mühendisliği süreci QFD tarafından belirlenir:

- Paydaşların niteliklerini veya gereksinimlerini belirleyin
- Gereksinimlerin teknik özelliklerini belirleyin.
- Gereksinimleri teknik özelliklerle ilişkilendirin.
- Rakip ürünlerin bir değerlendirmesini yapın.
- Teknik özellikleri değerlendirin ve her özellik için bir hedef değer belirleyin.
- Geliştirme çabası için teknik özelliklere öncelik verin.

QFD, rekabetçi analiz için yapılandırılmış bir yaklaşım kullanır. Yani, bir özellik listesi rekabetçi ürünler için tüm ilgili özelliklerin birleşiminden yaratılmıştır. Bu özellikler, bir rekabet matrisinin sütunlarını oluşturur. Satırlar, rakip ürünleri temsil eder ve karşılık gelen hücreler, her üründe bulunan özellikler için doldurulur. Matris daha sonra yeni veya revize edilmiş ürün için bir başlangıç gereksinimleri seti formüle etmek için kullanılabilir. Matris ayrıca şunu sağlamaya yardımcı olur: Temel özellikler yeni sistemden çıkarılmaz ve istenen gereksinimlerin eksiksizliği kalitesinin iyileştirilmesine katkıda bulunabilir.

Örnekleme için, evcil hayvan mağazası POS sistemi için kısmi bir rekabet analizi Tablo 3.1'de gösterilmektedir. Yalnızca çok yüksek düzeyli özelliklerin gösterildiğine dikkat edin, ancak istenen ayrıntı düzeyine inerek matrisi büyük ölçüde genişletebiliriz. Matris bize zorunlu ve isteğe bağlı

özelliklerden oluşan bir başlangıç seti sunar. Örneğin, tüm bu ürünlerde kablosuz desteğinin bulunduğunu belirtmek, yeni pet shop POS sisteminde böyle bir özelliğin zorunlu olduğunu gösterebilir.

Gereksinim mühendisliğine toplam yaşam döngüsü yaklaşımını dahil ettiğinden, QFD'nin diğer bağımsız ortaya çıkarma tekniklerine göre birçok avantajı vardır. QFD, kullanıcıların ve yöneticilerin katılımını artırır. Geliştirme yaşam döngüsünü kısaltır ve genel proje geliştirmeyi iyileştirir. QFD, iletişim süreçlerini yapılandırarak ekip katılımını destekler. Son olarak, bilgi kaybını önleyen önleyici bir araç sağlar.

Bununla birlikte, QFD'nin bazı dezavantajları vardır. Örneğin, zamansal gereksinimleri ifade etmede zorluklar olabilir. Ve tamamen yeni bir proje türü ile QFD'yi kullanmak zordur, var olmayan bir şey için müşteri gereksinimlerini nasıl keşfedersiniz ve rekabetçi ürünleri nasıl oluşturur ve analiz edersiniz? Bu durumlarda çözüm, benzer veya ilgili ürünlere bakmaktır, ancak yine de bilişsel bir boşluk olma eğilimi vardır.

**Table 3.1** Pet Shop Satış Noktası Sistemi için Kısmi Rekabet Analizi

| Özellik                             | Rakip Ürün            |               |                 |
|-------------------------------------|-----------------------|---------------|-----------------|
|                                     | Favori evcil hayvanım | Yakın arkadaş | <i>Fido-2.0</i> |
| Desteklenen Max Eşzamanlı Kullanıcı | 100                   | 250           | Unlimited       |
| Kablosuz Cihaz Desteği              | Yes                   | Yes           | Yes             |
| İş Analitiği Özellikleri            | Yes                   | Yes           | No              |
| İşletim Sistemi Desteği             | Windows/ Mac/Linux    | Windows/Linux | Windows/Mac     |
| Maliyet (Temel Sistem)(\$)          | 50                    | 110           | 75              |

Bazen belirli fonksiyonlar için ölçümler bulmak ve soyutlama seviyesini tek tip tutmak zordur. Ve ne kadar az bilirsek, o kadar az belgeliyoruz. Son olarak, özellik listesi kontrolsüz bir şekilde büyüdükçe, kalite evi bir "konak" haline gelebilir.

QFD, birincil gereksinimleri ortaya çıkarma yaklaşımı olarak kullanılmasa bile, mümkün olan her yerde rekabetçi sistem analizine yönelik yaklaşımı kullanılmalıdır. QFD rekabetçi analizinin yapılandırılmış doğası, daha eksiksiz bir gereksinimler kümesine yol açan hiçbir önemli gereksinimin eksik olmasını sağlamanın etkili bir yoludur.

### QUESTIONNAIRES/SURVEYS(Anketler)

Gereksinim mühendisleri, büyük paydaş gruplarına ulaşmak için genellikle anketleri ve diğer anket araçlarını kullanır. Anketler, genellikle kapsam sınırlarını hızlı bir şekilde tanımlamak için ortaya çıkarma sürecinin erken aşamalarında kullanılır.

Her türden anket sorusu kullanılabilir. Örneğin, sorular kapalı (örneğin, çoktan seçmeli, doğru-yanlış) veya açık uçlu olabilir, serbest biçimli yanıtlar içerebilir. Kapalı sorular, analiz için daha kolay kodlama avantajına sahiptir ve sistemin kapsamını bağlamaya yardımcı olur. Açık sorular daha fazla özgürlük ve yenilik sağlar, ancak analiz edilmesi daha zor olabilir ve kapsam kaymasını teşvik edebilir.,

Örneğin, evcil hayvan mağazası POS sistemi için olası bazı anket soruları şunlardır:

- Envanterinizde kaç tane benzersiz ürün (SKU) taşıyorsunuz?  
0–1000; (b) 1001–10,000; (c) 10.001–100.000; (d) >100.000  
(SKU: Her tür ürün için barkod numarası olarak düşünebiliriz)
- Kaç farklı depo alanınız var?
- Kaç farklı mağaza lokasyonunuz var?
- Şu anda kaç tekil müşteriniz var?

Sorular, kapalı uçlu sorular için bile yeterince çerçevesiz, gereğinden fazla ve alttan almanın bir tehlikesi vardır. Bu nedenle, anket ortaya çıkarma teknikleri, etki alanı hem paydaşlar hem de gereksinim mühendisi tarafından çok iyi anlaşılmalıdır.

Büyük ölçekli anketler gerçekleştirilmeden önce, amaçlanan anket popülasyonunun küçük bir alt kümesiyle bir pilot çalışma yürütmek önemlidir. Sonuçlar analiz edilir ve kafa karıştırıcı, eksik veya gereksiz soruları belirlemek amacıyla anket katılımcıları ile görüşülür. Daha sonra, anket daha büyük bir nüfusa uygulanmadan önce araç iyileştirilebilir.

Anket verilerini analiz ederken, özellikle katılımcılardan istenen özellikleri belirlemelerini ve sıralamalarını isterken, aşağıdaki etkiye dikkat edin.

“Gerçekleştirilmesi gerekmeyen bir dizi seçenek verildiğinde, bir kişi kararın gerçekten verilmesi durumunda çok daha fazla sayıda seçeneği arzu etme eğiliminde olacaktır.”

Aşağıdaki örnekte dolaylı olarak bu etkiye “dondurma deposu etkisi” diyoruz. Ürün araştırmasının bir parçası olarak, yanıtlayanların satın alacakları dondurma lezzetlerini kontrol ettikleri bir araçla birkaç kişiyle anket yapıyor. Ankette listelenen 30 farklı lezzetten 20'sinin, katılımcıların %50'si veya daha fazlası tarafından seçildiğini tespit etti. Böylece bu 20 çeşidi kabaca anket sonuçlarının gösterdiği talebe göre üretmeye ve stoklamaya karar veriyor. Yine de, dondurma dükkanını açtıktan 1 hafta sonra, işinin %90'ının en iyi üç aromadan (çikolata, vanilya ve çilek) kaynaklandığını keşfeder. Envanterinde tuttuğu anketteki diğer 17 lezzetten bazıları hiç satın alınmadı. Müşteriler ankette seçeceklerini söyledikleri bu tatlar sıra satın almaya geldiğinde farklı davrandılar. Bu nedenle, müşterilere özellik setleri hakkında seçenekler sunarken dondurma dükkanının etkisini unutmayın; onlar bir şey söyler ve başka bir şey yapar.

Anketler telefon, e-posta, yüz yüze ve Web tabanlı teknolojiler kullanılarak yapılabilir. Anket oluşturma ve kullanılması gereken sonuçları toplama ve analiz etme sürecini basitleştirmek için çeşitli ticari araçlar ve açık kaynaklı çözümler mevcuttur.

### **REPERTORY GRİDS(Repertuar Gridleri)**

Repertuar ızgaraları, sistemdeki farklı varlıkların çeşitli özellikleri için yapılandırılmış bir sıralama sistemi içerir ve genellikle müşteriler alan uzmanı olduğunda kullanılır. Repertuar ızgaraları, özellikle paydaş grupları içindeki anlaşma ve anlaşmazlığın tanımlanması için yararlıdır. Izzaralar, satırların sistem varlıklarını ve istenen nitelikleri temsil ettiği ve sütunların her bir paydaşa dayalı sıralamaları temsil ettiği bir özellik veya kalite matrisi gibi görünür. Izzaralar hem nitelikleri hem de özellikleri

bünyesinde barındırabilirken, genellikle şebekelerin, analiz ve uyumsuzluk çözümünün tutarlılığını sağlamak için tüm özelliklere veya tüm niteliklere sahip olması durumudur.

Tekniği göstermek için, Şekil 3.6, bagaj taşıma sisteminin çeşitli nitelikleri için bir repertuar ızgarasını temsil etmektedir. Burada, havalimanı operasyon müdürü için tüm niteliklerin esasen en yüksek öneme sahip olduğunu görüyoruz (güvenlik biraz daha düşük, 4).

|                   |   |   |   |
|-------------------|---|---|---|
| Bagaj Taşıma Hızı | 1 | 1 | 5 |
| Hata Toleransı    | 4 | 5 | 5 |
| Güvenlik          | 5 | 4 | 4 |
| Güvenilirlik      | 3 | 5 | 5 |
| Bakım Kolaylığı   | 3 | 5 | 5 |

Havaalanı Operasyon Müdürü  
Bakım Mühendisi  
Havayolu İşçileri Sendikası Temsilcisi

### Figure 3.6 Bagaj taşıma sistemi için kısmi repertuar ağı

Ancak Havayolu İşçileri Sendikası temsilcisi için güvenlik en önemli şeydir (sonuçta sendika üyeliğinin sistemle belirli bir düzeyde etkileşime girmesi gerekir).

Özünde, bu derecelendirmeler paydaşların gündemlerini veya farklı bakış açılarını yansıtır. Bu nedenle, repertuar ızgaralarının kullanılmasının, paydaş hedeflerini içeren anlaşmazlıklarla erkenden yüzleşmede neden çok yardımcı olabileceğini görmek kolaydır. Ek olarak, ızgaralar, paydaşların nitelikler ve özellikler hakkındaki tutumlarını göz ardı edilmesi zor bir şekilde yakaladıkları için, sistemin geliştirilmesinde daha sonra anlaşmazlıklarla başa çıkmak için değerli belgeler sağlayabilir. Yine de, repertuar ızgaralarını kullanırken, dondurma dükkanının etkisini hatırlayın—paydaşlar halka açık bir ortamda bir şey söyleyecek ve daha sonra farklı davranacaktır.

### SCENARIOS(SENARYOLAR)

Senaryolar, sistem çalışmasının, kullanıcı sınıflarının ve istisnai durumların üst düzey bir tanımını sağlayan, kullanılan sistemin gayri

Resmi açıklamalarıdır.

İşte evcil hayvan mağazası POS sistemi için örnek bir senaryo.

Bir müşteri evcil hayvan dükkânına girer ve sepeti çeşitli eşyalarla doldurur. Çıkış yaparken, kasiyer müşterinin sadakat kartı olup

Olmadığını sorar. Eğer öyleyse, kasiyer kartı kaydırır ve müşterinin kimliğini doğrular. Değilse, kasiyer yerinde bir tane tamamlamayı

Teklif eder.

Sadakat kartı etkinliğinden sonra kasiyer barkod okuyucu kullanarak ürünleri tarar. Her bir kalem tarandıkça satış toplamı alınır ve

Stok uygun şekilde güncellenir. Ürün taramasının tamamlanmasının ardından bir alt toplam hesaplanır. Daha sonra herhangi bir kupon ve

İndirim girilir. Yeni bir alt toplam hesaplanır ve geçerli vergiler eklenir. Bir makbuz yazdırılır ve müşteri nakit, kredi kartı, banka

Kartı veya çek kullanarak ödeme yapar. Tüm uygun toplamlar (satışlar, vergiler, indirimler, indirimler vb.) hesaplanır ve kaydedilir.

Etki alanı yeni olduğunda senaryolar oldukça kullanışlıdır (örneğin, uzay istasyonu için bir senaryo düşünün).

Kullanıcı hikâyeleri aslında bir senaryo biçimidir.

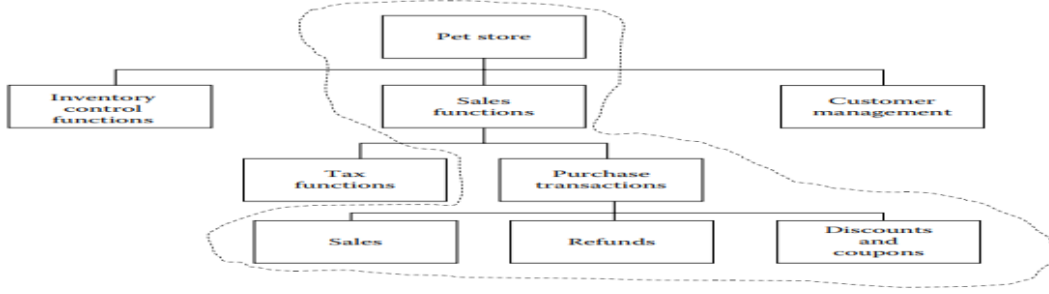
### **TASK ANALYSIS(GÖREV ANALİZİ)**

Daha önce incelediğimiz hiyerarşik yönelimli tekniklerin çoğu gibi, görev analizi de sistem tarafından gerçekleştirilecek görevlerin

İşlevsel bir ayrışmasını içerir. Yani, en üst düzeyde soyutlamadan başlayarak, tasarımcı ve müşteriler daha fazla ayrıntı düzeyi ortaya

Çıkarır. Bu ayrıntılı ayrışmayla, en düşük işlevsellik seviyesi (tek görev) elde edilir.

Örnek olarak, Şekil 3.7'de gösterilen evcil hayvan mağazası POS sistemi için kısmi görev analizini göz Önünde bulundurun.



şekil 3.7 Evcil hayvan mağazası POS sistemi için kısmi görev analizi.

Burada, kapsamlı evcil hayvan mağazası POS sisteminin üç ana görevden oluştuğu düşünülmektedir:stok kontrolü, satış ve müşteri

yönetimi.Satış fonksiyonları altında, bunların aşağıdaki görevlerden oluştuğunu görüyoruz:vergi fonksiyonları ve satın alma işlemleri

Ardından, satın alma işlemi işlevine geçin,bu görevleri satış, geri ödeme, indirim ve kupon görevlerine ayırırız.Görev analizi ve

ayrıştırma, yeterli bir ayrıntı düzeyine ulaşılan kadar devam eder (tipik olarak, bir yöntem veya ayrıştırılmaz yordam düzeyine kadar)

ve diyagram tamamlanır.

### USE CASES(KULLANIM DURUMLARI)

Kullanım durumları, daha sofistike müşterilerin ve paydaşların eksiklerini tanımlamalarının bir yoludur.Kullanım durumları, özellikle

insan kullanıcıları ve diğer sistemler olmak üzere sistem ve sistem çevresindeki etkileşimleri gösterir.Saf yazılım veya hibrit donanım

yazılım sistemlerinin davranışını modellemek için kullanılabilirler.

Kullanım durumları, sistemin işleyiş senaryolarını tasarımcının (müşterilerin aksine) perspektifinden açıklar.Kullanım durumları

genellikle sistemin dış ortamıyla etkileşimlerini gösteren bir kullanım durumu diyagramı kullanılarak temsil edilir.

\*\*\* Bu tartışma, Laplante'de (2006) bulunanlardan izin alınarak uyarlanmıştır.\*\*\*

Bir kullanım durumu diyagramında, kutu sistemin kendisini temsil eder.Çubuk figürleri, sistemle etkileşime giren dış varlıkları belirleyen

"aktörleri" temsil eder.Aktörler;insanlar,diğer sistemler veya cihaz girişleri olabilir.Dahili elipsler, aktörlerin her biri bir kullanım

etkinliğini temsil eder (Use Cases).Katı çizgiler aktörleri her kullanımda ilişkilendirir.Şekil 3.8'de bagaj muayene sistemi için bir

kullanım durumu diyagramı gösterilmektedir.

Bagajın bir görüntüsünü ("görüntü bagajı"), bir güvenlik tehdidinin algılanmasını (bu durumda çanta çevrimdışı işlem için konveyörden reddedilir),

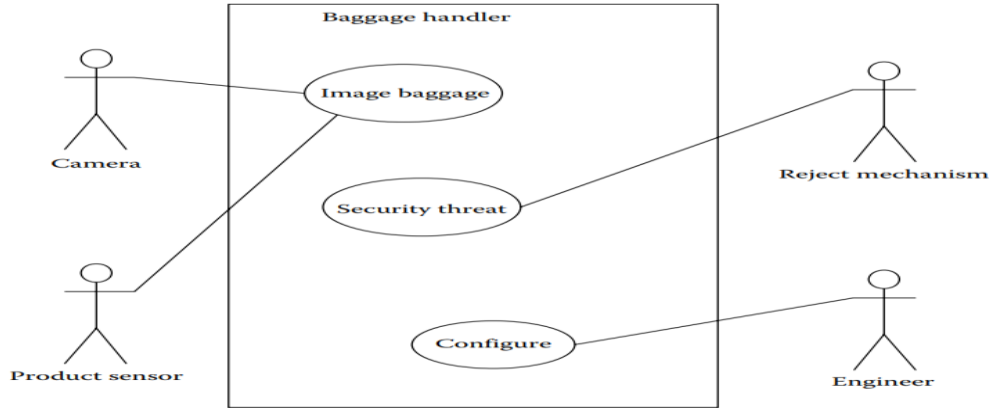
ve ardından sistem mühendisi tarafından yapılandırılan üç kullanım gösterilir.Görüntüleme kamerasının, ürün sensörünün ve reddetme mekanizmasının insan

benzeri bir çubuk figürü ile temsil edildiğine dikkat edin - bu tipiktir - çubuk figürü insan olsun ya da olmasın bir sistem "aktörünü" temsil eder.

Ek B, Şekil B.4 başka bir örnek sağlar — ıslak kuyu için bir kullanım durumu diyagramı.

Her kullanım durumu, incelenen sistemin çalışma senaryolarının yanı sıra ön ile son koşulları ve istisnaları açıklayan bir belge biçimidir.

Yinelemeli bir geliştirme yaşam döngüsünde, analiz ve tasarım iş akışları ilerledikçe bu kullanım durumları giderek daha rafine ve ayrıntılı hale gelecektir.



(Şekil 3.8 Bagaj muayene sisteminin kullanım durumu diyagramı.)

Etkileşim diyagramları daha sonra her kullanım örneği tarafından tanımlanan davranışları açıklamak için oluşturulur. İlk yinelemede, bu diyagramlar sistemi bir “kara kutu” olarak tasvir

eder, ancak etki alanı modellemesi tamamlandıktan sonra, kara kutu daha sonra görüleceği gibi nesnelere işbirliğine dönüştürülür.

İyi geliştirilirse, bazen kullanım örnekleri bir desen dili oluşturmak için kullanılabilir ve bu desenlerle türetilmiş tasarım öğeleri, ilgili sistemlerde yeniden kullanılabilir (Issa ve Al-Ali 2010). Gereksinimleri belirlerken kalıpların kullanılması daha yüksek düzeyde tutarlılık sağlar ve belirli gereksinim özelliklerinin otomatik ölçümündeki

hataları azaltabilir.

Gereksinimlerin keşfedilmesinde ve gereksinimlerin şartname belgesindeki modellenmesinde bu kadar önemli bir araç haline geldiklerinden, birçok örnekle birlikte kullanım

durumlarının kapsamlı bir tartışması Ek E'de bulunabilir.

### **USER STORIES\*(Kullanıcı Hikayeleri)**

Kullanıcı hikayeleri, ilk gereksinim keşfi ve proje planlaması için kullanılan kısa konuşma metinleridir. Kullanıcı hikayeleri çevik metodolojilerle birlikte yaygın olarak kullanılmaktadır.

Kullanıcı hikayeleri, müşteriler tarafından sistemin kendileri için yapması gerekenler açısından ve kendi "anlatımları" yazılır. Kullanıcı hikayeleri genellikle üçe beş

inçlik bir kartta yazılmış iki ila dört cümleden oluşur. Yaklaşık 80 kullanıcı hikayesi genellikle bir sistem artışı veya evrimi için uygundur, ancak uygun sayı, uygulama boyutuna,

kapsamına ve kullanılacak geliştirme metodolojisine (örneğin, çevik ve artımlı) bağlı olarak büyük ölçüde değişecektir.

Evcil hayvan mağazası POS sistemi için bir kullanıcı hikayesi örneği aşağıdaki gibidir:

- Her müşteri bir kasada kolayca check-out yapabilmelidir.
- Self servis desteklenecektir.
- Tüm kuponlar, indirimler ve iadeler bu şekilde ele alınmalıdır.

Kullanıcı hikayeleri yalnızca hikayenin ne kadar süreceğine dair makul derecede düşük riskli bir tahminde bulunmak için yeterli ayrıntı sağlamalıdır. Uygulama zamanı geldiğinde, hikaye geliştiricileri ayrıntıları ortaya çıkarmak için müşteriyle buluşacak.

Ayrıca kullanıcı hikayeleri kabul testinin temelini oluşturur. Örneğin, kullanıcı hikayesinin doğru uygulanıp uygulanmadığını doğrulamak için bir veya daha fazla otomatik kabul testi oluşturulabilir.

Kullanıcı öyküleri Bölüm 7'de ve Ek D'de daha ayrıntılı olarak ele alınmıştır.

\* Ibid.

## **VIEWPOINTS(BAKIŞ AÇISI)**

Bakış açıları, bilgiyi farklı seçim bölgelerinden (bakış açısından) organize etmenin bir yoludur. Örneğin, bagaj taşıma sisteminde, aşağıdaki paydaşların her biri için sistemin farklı bakış açıları vardır:

- Bagaj Taşıma Personeli
- Gezinler
- Bakım mühendisleri
- Havaalanı yöneticileri

- Düzenleyici kurumlar

Bu paydaşların her birinin ihtiyaçlarını ve bu bakış açılarının ortaya çıkardığı çelişkileri kabul ederek, çatışmalar çeşitli yaklaşımlar

kullanılarak uzlaştırılabilir.

Gerçek bakış açıları, işletme alanından, süreç modellerinden, işlevsel gereksinim spesifikasyonlarından, organizasyon modellerinden vb. Çeşitli

bilgileri içerir.

\*\*Sommerville ve Sawyer (1997), her bakış açısında aşağıdaki bileşenlerin olması gerektiğini öne sürdüler:

-Belirtimde kullanılan gösterimi tanımlayan bir gösterim stili.

-"Bakış açısı tarafından ele alınan endişe alanı" olarak tanımlanan bir alan.

-Tanımlanan tarzda ifade edilen bir sistemin modeli olan bir belirtim.

-Şartnamenin oluşturulmasında, denetlenmesinde ve değiştirilmesinde gerçekleştirilen eylemlerin izi olan bir çalışma kaydı.

-Bakış açısı analizi genellikle gereksinimlerin önceliklendirilmesi, mutabakatı ve sıralanması için kullanılır.

### **Workshops(Atölyeler)**

Atölyeler genel olarak, gereksinim sorunlarını çözmek için paydaşların toplanmasıdır. Atölyeleri iki tip olarak ayırt edebiliriz — resmi

ve gayri resmi.

Resmi atölye çalışmaları iyi planlanmış toplantılardır ve genellikle sözleşmeyle zorunlu kılınan "teslim edilebilir" etkinliklerdir. Örneğin, DOD-MIL-STD-2167 birden fazla gerekli ve isteğe

bağlı atölye çalışması (eleştirel incelemeler) dahil edildi. Resmi bir atölye tarzının iyi bir örneği JAD'de somutlaştırılmıştır.

Gayri resmi atölyeler genellikle yüksek yapılandırılmış toplantılardan daha az sıkıcıdır. Ancak gayri resmi toplantılar çok özensiz olma eğilimindedir, yanlış güvenlik ve bilgi

kaybına neden olabilir. Bir tür atölyeye ihtiyaç duyulursa, daha önce tartışılan başarılı toplantılar için parametreler kullanılarak resmi bir atölye yapılması önerilir.

### **Elciktin Nonfunctional Requirements\*(İşlevsel Olmayan Gereksinimlerin Ortaya Çıkarılması)**

İşlevsel olmayan gereksinim (NFR) belirleme teknikleri, işlevsel gereksinim belirleme tekniklerinden farklıdır.NFR'ler genellikle gereksinim analizi sırasında gayri resmi olarak belirtilir,

genellikle çelişkilidir ve yazılım geliştirme sürecinde uygulanması ve doğrulanması zordur. Borg ve arkadaşları (2003), farklı kuruluşlarda NFR ile ilgili sorunların köklerini

belirlemeye yönelik görüşmeler gerçekleştirdi.Sonuç olarak, NFR ile ilgili sorunların gelişim sürecinin dört aşamasında ortaya çıktığı sonucuna varılmıştır:ortaya çıkarma, dokümantasyon,

yönetim ve test;ortaya çıkarma aşamasında NFR ihmalî tüm geliştirme süreci boyunca yayıldığından,NFR ile ilgili potansiyel sorunların ana kaynağı ortaya çıkarmadır.Bunun nedenleri, örneğin,

şunlardır: (i) gereksinimler aşamasında belirli kısıtlamalar bilinmemektedir, (ii) NFR'ler birbirleriyle çatışma eğilimindedir, (iii) FRS ve NFR'LERİ ayırmak, aralarındaki bağımlılıkları

izlemeyi zorlaştırırken, tüm gereksinimler birlikte karıştırılırsa işlevsel ve işlevsel olmayan hususların ayrılması zordur.

Mevcut yöntemlerin çeşitliliği arasında NFR'LERİ ortaya çıkarmak, detaylandırmak ve belgelemek için bir yöntem seçmek kolay değildir.

NFR'leri keşfetmenin yaygın yolları, sistem niteliklerinin rekabetçi analizini içerir:NFR'ler, piyasadaki rakip ürünlerin niteliklerini analiz ederek keşfedilebilir.Örneğin, rakip

bir ürünün yanıt süresi nedir?Ve daha iyi yapmak zorunda mıyız?

\*\*\*Penn Eyaleti'nden Dr. Mohamad Kassab bu bölüme katkıda bulundu.\*\*\*

## **Ortaya Çıkarma Özet**

Bu tur birçok ortaya çıkarma tekniğini içeriyordu ve her birinin yol boyunca tartışılan avantajları ve dezavantajları vardı.Açıkçası, bu tekniklerden bazıları çok genel, bazıları çok

spesifik, bazıları paydaş bilgisine çok fazla güveniyor, bazılarıda yeterli değil, vb.Bu nedenle, gereksinimleri ortaya çıkarma zorluğunu başarılı bir şekilde ele almak için bazı

teknikler kombinasyonunun gerekli olduğu açıktır.

## Kabaca Tipine Gre eřitli Tekniklerin Dzenlenmesi

| Teknik Tipleri  | Teknikler  |
|-----------------|--|
| Domain-oriented | Card sorting<br>Designer as apprentice<br>Domain analysis<br>Laddering<br>Protocol analysis<br>Task analysis |
| Ethnography     | Ethnographic observation   |
| Goals           | Goal-based approaches<br>QFD   |
| Group work      | Brainstorming<br>Group work<br>JAD<br>Workshops  |
| Interviews      | Interviews<br>Introspection<br>Questionnaires  |
| Prototyping     | Prototyping  |
| Scenarios       | Scenarios<br>Use cases<br>User stories   |
| Viewpoints      | Viewpoints<br>Repertory grids  |

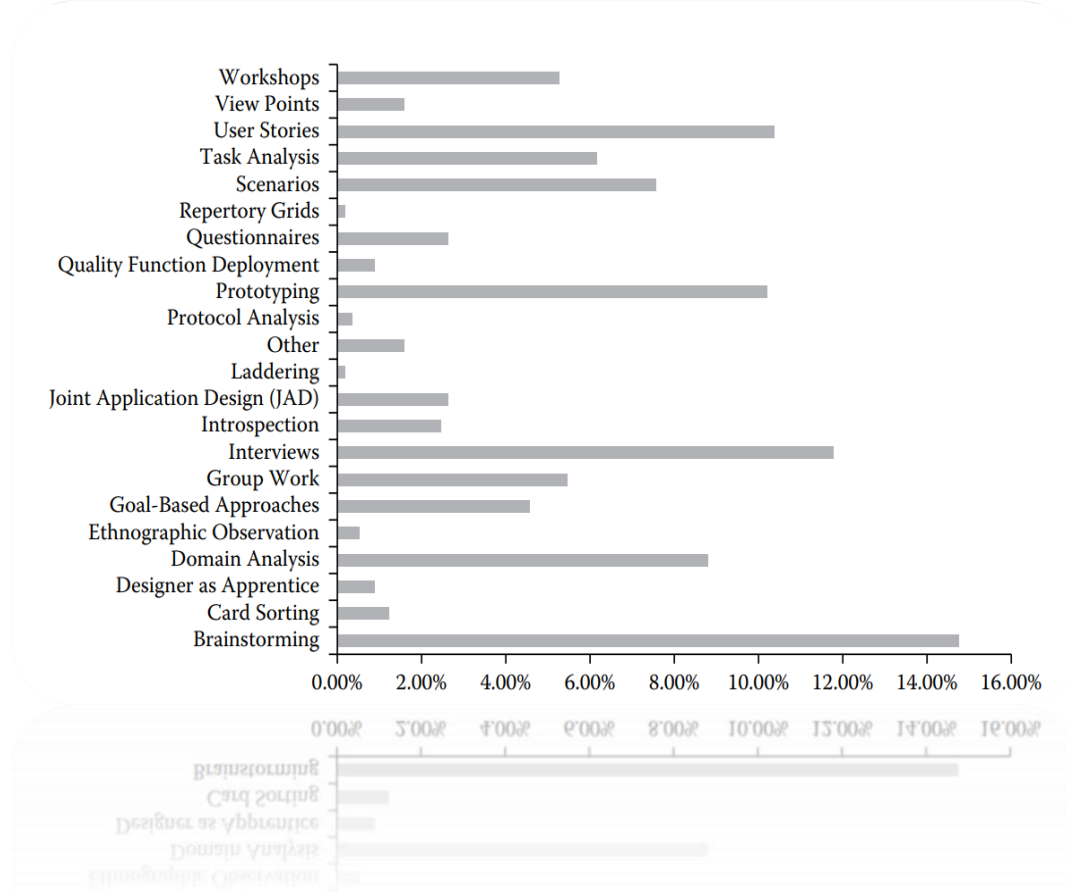


## Birbirini Tamamlayıcı veya Alternatif Teknikler

|                    | <i>Interviews</i> | <i>Domain</i> | <i>Groupwork</i> | <i>Ethnography</i> | <i>Prototyping</i> | <i>Goals</i> | <i>Scenarios</i> | <i>Viewpoints</i> |
|--------------------|-------------------|---------------|------------------|--------------------|--------------------|--------------|------------------|-------------------|
| <i>Interviews</i>  |                   | C             | A                | A                  | A                  | C            | C                | C                 |
| <i>Domain</i>      | C                 |               | C                | A                  | A                  | A            | A                | A                 |
| <i>Groupwork</i>   | A                 | C             |                  | A                  | C                  | C            | C                | C                 |
| <i>Ethnography</i> | A                 | A             | A                |                    | C                  | C            | A                | A                 |
| <i>Prototyping</i> | A                 | A             | C                | C                  |                    | C            | C                | C                 |
| <i>Goals</i>       | C                 | A             | C                | C                  | C                  |              | C                | C                 |
| <i>Scenarios</i>   | C                 | A             | C                | A                  | C                  | C            |                  | A                 |
| <i>Viewpoints</i>  | C                 | A             | C                | A                  | C                  | C            | A                |                   |

## Yaygın Kullanılan Gereksinim Belirleme Teknikleri

Konuyu noktalamadan önce , endüstride yaygın olarak kullanılan çeşitli gereksinim belirleme tekniklerinin ne sıklıkla kullanıldığına dair bir fikir elde edelim. Bunu yapmak için, yazılım uzmanları ile yapılan ankete geri dönüyoruz. Hangi gereksinim belirleme tekniklerini kullanıyorsunuz sorusunun cevapları aşağıda verilmiştir.



Yanıtlar

**Beyin Fırtınasının (Brainstorming), röportaj yapma (Interviews) ve prototip oluşturma (Prototyping) en sık kullanılan gereksinim belirleme teknikleriydi (her biri yaklaşık %10). Bu bölümde ele alınan diğer teknikler ise nispeten daha seyrek kullanılmıştır**

## Gereksinim Belirlemedeki Tehlikeler

Daha önce "shall not" davranışların istenmeyen çıktı davranışları kümesi olduğunu ve tehlikelerin ciddi veya felakete neden olma eğiliminde olan davranışların bir alt kümesi olduğunu belirttik. "Ciddi" ve "felaket" terimleri öznel, ancak genellikle can kaybı, ciddi yaralanma, büyük altyapı hasarı veya büyük finansal kayıp içerir.

*Örneğin, evcil hayvan mağazası POS'u için bazı "shall not" gereksinimleri aşağıdakileri içerir:*

- *Sistem, müşteri bilgilerini harici sistemlere açığa çıkarmayacaktır.*
- *Sistem yetkisiz erişime izin vermeyecektir.*
- *Sistem, müşterilerin mağaza kredisini fazla çekmesine izin vermeyecektir.*

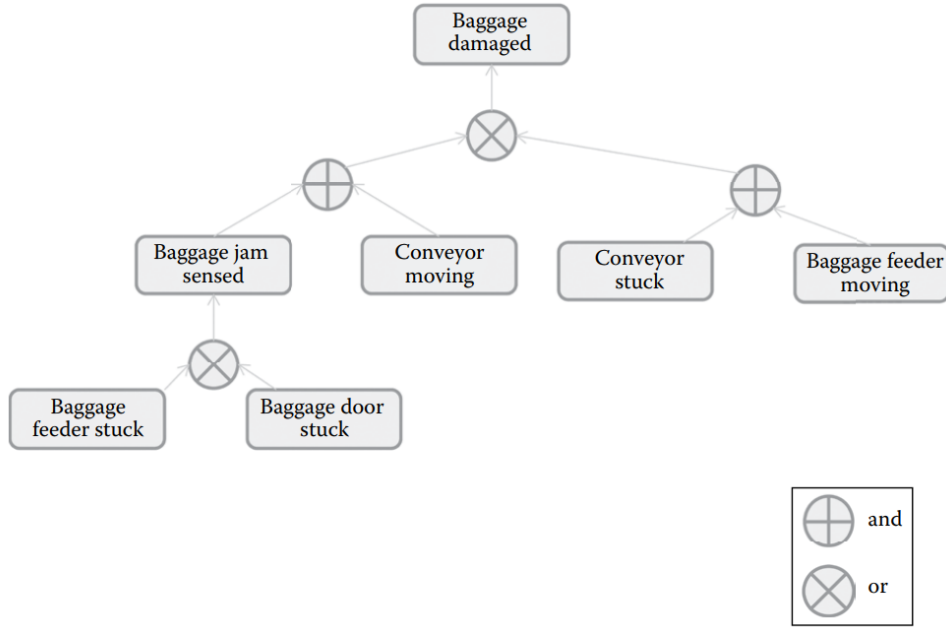
*Bu örneklerde ilk iki gereksinim, tehlike olarak kabul edilebilir çünkü şirkete maddi zarar verme potansiyeli üçüncü gereksinimden çok daha fazladır. Tehlikeler, doğal olarak meydana gelen girdi anormalliklerinin bir fonksiyonudur (Donanım Arızaları Gibi) veya yapay olarak meydana gelen (dışarıdan gelen saldırılar) gibi. Bu anormal girdi olaylarının tanımlanması ve sonuçta ortaya çıkan arızaların belirlenmesi gerekir ve sistem için uygun bir "shall not" gereksinimler kümesi geliştirmek için gereksinimlerin belirlenme aşamasında kritikliğin belirlenmesi gerekir. Diğer gereksinimlerde olduğu gibi,(shall not) gereksinimlerinin önceliklendirilmesi gerekir. Tehlike belirleme için tipik teknikler, kötüye kullanım vakalarının geleneksel gelişimini içerir. Tehlike tespiti için tipik teknikler arasında yanlış kullanım vakalarının geleneksel gelişimi, anti modelleme ve resmi yöntemler yer almaktadır.*

### *İstenmeyen Kullanım Senaryoları*

*Kullanım senaryoları, istenen davranışın yapılandırılmış kısa açıklamalarıdır. Tıpkı istenen davranışı açıklayan kullanım senaryoları olduğu gibi, istenmeyen davranışları açıklayan yanlış kullanım senaryoları vardır. Çoğu sistem için tipik suiistimaller arasında güvenlik ihlalleri ve diğer kötü amaçlı davranışların yanı sıra eğitimsiz kişiler tarafından kötüye kullanım da bulunur. Cleland-Huang (2016) kötüye kullanımı tanımlamanın çeşitli yollarını açıklar tehdit modellemesi ve beyin fırtınasına dayalı vakalar. Kullanım senaryoları oluşturmanın kolay bir yolu, istenmeyen bir kişinin rolünü üstlenmektir yani, sistemin istenmeyen bir kullanıcıını seçin ve ardından böyle bir kişinin davranışlarını modelleyin. İstenmeyen kişiler bilgisayar korsanlarını, davetsiz misafirleri, casusları içerebilir ve hatta iyi niyetli, ama beceriksiz kullanıcıları. Bu kişileri tanımlamanın olası tehlikeleri bulmada nasıl yardımcı olduğunu görmek için, aşağıdaki örnekleri göz önünde bulundurun. Evcil hayvan mağazası POS sisteminde, bir bilgisayar korsanının bu sisteme nasıl sızacağını düşünerek, bunlara engelleyecek önlemler türetin. Bagaj taşıma sisteminde, bir gereksinim mühendisi, beceriksiz veya düşüncesiz bir gezgin rolünü üstlenecek ve daha sonra bu kişilerin güvenliğini sağlayacak gereksinimleri belirleyecektir. Kötüye kullanım senaryoları türetme ihtiyacı, tüm olumsuz olasılıkları tam olarak tanımlamak için bir nedendir, çünkü bu kişilerin büyük olasılıkla istenmeyen çok sayıda kişiden oluşması muhtemeldir.*

### *Anti Modeller*

*İstenmeyen davranışları türetmenin bir başka yolu da sistem için anti modeller oluşturmaktır. Anti modeller hata ağaçlarıyla ilgilidir, yani sistem arızasına yol açan istenmeyen davranışlar için neden sonuç hiyerarşisi oluşturularak model türetilir. Ardından, sistem arızasının nedenleri yapılmaması gerekenleri oluşturmak için kullanılır. Örneğin, Şekil 3.10'da gösterilen hasarlı bagajın istenmeyen sonucunu içeren bagaj taşıma sisteminin güvenlik işlevselliğini göz önünde bulundurun.*



*Bu şekil bize aşağıdaki örnek gereksinimleri yazmamızı sağlıyor:*

- *Bir bagaj sıkışması hissedilirse, konveyör hareket etmeyecektir.*
- *Bagaj motoru sıkışmışsa, konveyör hareket etmeyecektir.*
- *Bagaj kapısı sıkışmışsa, konveyör hareket etmeyecektir.*
- *Konveyör sıkışmışsa, bagaj motoru hareket etmemelidir.*

*Bu gereksinimlerin daha fazla analize ve muhtemelen basitleştirmeye ihtiyacı var, ancak anti model bu örnek gereksinimleri sistematik bir şekilde üretmemize yardımcı oldu.*

#### *Kaynakça*

- Akao, Y. (1990). Quality Function Deployment: Integrating Customer Requirements into Product Design. Productivity Press, Cambridge, MA.
- Basili, V. R., & Weiss, D. (1984). A methodology for collecting valid software engineering data. IEEE Transactions on Software Engineering, Vol 10, 728–738.
- Berman, B. (2012). 3-D printing: The new industrial revolution. Business Horizons, 55(2): 155–162.
- Borg, A., Yong, A., Carlshamre, P., & Sandahl, K. (2003). The bad conscience of requirements engineering: An investigation in real-world treatment of non-functional requirements. In Third Conference on Software Engineering Research and Practice in Sweden (SERPS'03), Lund, pp. 1–8.
- Centers for Medicare & Medicaid Services. "The health insurance portability and accountability act of 1996 (HIPAA)." Online at <http://www.hhs.gov/hipaa> (1996) (accessed June 2017).

- Cleland-Huang, J. (2014). How well do you know your personae non gratae? *IEEE Software*, 31(4), 28–31.
- Cleland-Huang, J., Denning, T., Kohno, T., Shull, F., & Weber, S. (2016). Keeping ahead of our adversaries. *IEEE Software*, 33(3), 24–28.
- Cleland-Huang, J., & Laurent, P. (2014). Requirements in a global world. *IEEE Software*, 31(6), 34–37.,
- Eberlein, A., & Jiang, L. (2011). Selecting requirements engineering techniques. In P. Laplante (Ed.), *Encyclopedia of Software Engineering*, Published online, pp. 962–978. Boca Raton, FL: Taylor & Francis.
- Herrmann, A., Kerkow, D., & Doerr, J. (2007). Exploring the characteristics of NFR methods – A dialogue about two approaches. In *REFSQ 2007, LNCS 4542*, pp. 320–334.
- Issa, A. A., & Al-Ali, A. (2010). Use case patterns driven requirements engineering. In *Proceedings Second International Conference on Computer Research and Development*, pp. 307–313.
- Kassab, M., Neill, C., & Laplante, P. (2014). State of practice in requirements engineering: Contemporary data. *Innovations in Systems and Software Engineering*, 10(4): 235–241.
- Kassab, M., & Ormandjieva, O. (2014). Non-functional requirements in process-oriented approaches, ed., Phillip A. Laplante, In *Encyclopedia of Software Engineering*, pp. 1–11. Taylor and Francis, Boca Raton, FL.
- NASA (National Aeronautics and Space Administration). (2013). *NASA-STD 8719.13, Software Safety Standard*. <http://NASA.gov> (accessed January 2017).
- Robinson, W. N. (2010). A roadmap for comprehensive requirements modeling. *Computer*, 43(5): 64–72
- Sommerville, I., & Sawyer, P. (1997). Viewpoints for requirements engineering, *Software Quality Journal*, 3: 101–130.
- Voas, J., & Laplante, P. (2010). Effectively defining shall not requirements. *IT Professional*, 12(3): 46–53.

Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In A. Aurum & C. Wohlin (Eds.), *Engineering and Managing Software Requirements*, pp. 19–46. Springer, Berlin