

Serhat Obay
Rıfkı Burak Dinç
Pelin Altürk
Tuğçe Özelmacı

Yazılım ve Sistemler için Gereksinim Mühendisliği

Bölüm 3 **Gereksinimlerin Ortaya** **Çıkarılması**

Tartışacağımız teknikler şunlardır:

- Beyin Fırtınası
- Kart Sıralama
- Çırak Olarak Tasarımcı
- Domain(Alan) Analizi
- Etnografik Gözlem
- Hedefe Dayalı Yaklaşımlar
- Grup Çalışması
- Toplantılar
- İç Gözlem
- Ortak Uygulama Geliştirme(JAD)
- Merdiven
- Protokol Analizi
- Prototipleme
- Kalite İşlevi Dağıtımı(QFD)
- Anketler
- Repertuar Izgaraları(Şebeke, Sistem)
- Senaryolar
- Görev Analizi
- Kullanım Durumları
- Kullanıcı Hikayeleri
- Bakış Açıları
- Atölyeler

Bu liste, Zowghi ve Coulin (1998) tarafından önerilenden uyarlanmıştır.

Gereksinimlerin Ortaya Çıkarılması için Hazırlanma

- Tüm müşterilerin ve paydaşların belirlenmesi, gereksinimlerin ortaya çıkarılmasına hazırlanmanın ilk adımıdır. Ancak paydaş grupları ve özellikle müşteriler homojen olmayabilir ve bu nedenle her bir alt gruba farklı davranmanız gerekir.

Örneđin, evcil hayvan mađazası POS sistemi için farklı kullanıcı alt sınıfları şunları içerir:

- Kasiyerler
- Yöneticiler
- Sistem Bakım Personeli
- Mađaza Müşterileri
- Envanter/Depo Personeli
- Muhasebeciler (Vergi Bilgilerini Girmek İçin)
- Satış Departmanı (Fiyat ve İndirim Bilgilerini Girmek İçin)

İşte başka bir kullanıcı sınıfı bölümlenme örneđi. Bagaj taşıma sistemi dahil olmak üzere birçok farklı sistem vardır.

- Gezginler
- Sistem Bakım Personeli
- Bagaj Görevlileri
- Havayolu Zamanlayıcıları/Göndericileri
- Havaalanı Personeli
- Havaalanı Yöneticileri ve Politika Yapıcılar

Ancak her biri farklı ihtiyaları olan eřitli gezginler var. rneęin, ařaęıdaki alt sınıfları gz nnde bulundurun:

- ocuklar
- Yařlılar
- İř Adamı
- Sıradan Gezginler
- Askeri Personel
- Siviller Sık Uanlar

Beyin Fırtınası

- Beyin fırtınası, sistemler için kapsamlı hedefler oluşturmak için müşteriler ve diğer paydaşlarla yapılan resmi olmayan oturumlardan oluşur.
- JAD(Joint Application Development) tekniği beyin fırtınasını (ve çok daha fazlasını) içerir ve muhtemelen diğer grup odaklı uyarma tekniklerinin çoğu, dolaylı olarak bir tür beyin fırtınası içerir.

Kart Sıralama

- Bu teknik, paydaşların sistem/yazılım ürünü için işlevsellik hakkında önemli bilgileri içeren bir dizi kartı tamamlamasını içerir. Paydaşların/müşterilerin her bir işlevsellik için sıralama ve gerekçe içermesi de iyi bir fikirdir.
- Kart sıralamasında 2 ihtimal vardır:

- Müşterilerin ve paydaşların kartları tamamlamasına izin verecek süre önemli bir karardır. Kart sıralama alıştırmaları birkaç saat içinde tamamlanabilse de, paydaşların acele etmesi önemli, eksik işlemlere yol açacaktır.
- Diğer bir alternatif ise, müşterilerin kartları 2 saatlik bir oturumda tamamlamasını ve ardından 1 hafta sonra başka bir kart tamamlama ve inceleme oturumu için geri dönmesini sağlamaktır.
- Her durumda, her kart oluşturma oturumundan sonra, gereksinim mühendisi bu kartları bir şekilde düzenler, genellikle işlevleri mantıksal olarak kümeler.

Geri dönen
müşteriyi tanımlayın
Öncelik-Yüksek

Envanter kayıtlarını
güncelleme
Öncelik-Yüksek

Müşteri sadakati
özelliğini yönetin
Öncelik-Orta

Satış vergisi
raporları hazırlamak
Öncelik-Yüksek

Gıda dışı ürünlere
satış vergisi
uygulayın
Öncelik-Orta

Süreci açıklamak için, Şekil 3.1, müşteri tarafından evcil hayvan mağazası POS sistemi için oluşturulan ve sıralanmamış bir yığın halinde duran küçük bir kart alt kümesini göstermektedir.

Şekil 3.1

Müşteri yönetimi

Geri dönen müşteriye tanımlayın
Öncelik-Yüksek

Müşteri sadakati özelliğini yönetin
Öncelik-Orta

Vergi fonksiyonları

Satış vergisi raporları hazırlamak
Öncelik-Yüksek

Gıda dışı ürünlere satış vergisi uygulayın
Öncelik-Orta

Envanter özellikleri

Envanter kayıtlarını güncelleme
Öncelik-Yüksek

Gereksinim mühendisi bu kart yığını analiz eder ve iki kartın "müşteri yönetimi" işlevleriyle, iki kartın "vergi işlevleri" ile ve bir kartın "envanter özellikleri" veya işlevleriyle ilgili olduğuna karar verir ve kartları uygun şekilde düzenler. Şekil 3.2'de gösterildiği gibi yığınlar.

Şekil 3.2

Çırak Olarak Tasarımcı

- Çırak olarak tasarımcı, müşterinin ihtiyaçlarını anlamak için müşterinin işi hakkında yeterince bilgi edinmek için gereksinim mühendisinin müşterinin "omzunun üzerinden baktığı" bir gereksinim keşif tekniğidir.
- Müşteri ile tasarımcı arasındaki ilişki, usta ile çırak arasındaki ilişki gibidir.
- **Not: * Bu tartışma, izin alınarak Laplante'de (2006) bulunan bir tartışmadan uyarlanmıştır.**

- Bir ıarak, bir grevin birden fazla rneđini gzlemleyerek ve varyasyonları dahil ederek, kendisinin nasıl yapılacağına dair bir anlayıř oluřturarak iřin stratejilerini ve tekniklerini ğrenir.
- Bu tekniđin alıřması iin, gereksinim mhendisi ařađıdakiler dahil iřin yapısını ve anlamını anlamalıdır:
 - İři halletmek iin strateji
 - Yola ıkan kısıtlamalar
 - alıřmayı desteklediđi iin fiziksel evrenin yapısı
 - İři blnr
 - Tekrarlayan aktivite kalıpları
 - Bunların herhangi bir potansiyel sistem zerindeki etkileri

- Tasarımcı, herhangi bir yanlış anlamamanın düzeltilebilmesi için müşteriye işi anladığını göstermelidir. Son olarak, tasarımcıyı çirak yaklaşımı olarak kullanmak, gereksinimleri keşfetmenin ötesinde başka proje faydaları da sağlar. Örneğin, bu tekniği kullanmak, modellenmekte olan süreci iyileştirmeye yardımcı olabilir.
- Hem müşteri hem de tasarımcı bu süreç boyunca öğrenir; müşteri neyin mümkün olabileceğini öğrenir ve tasarımcı da çalışma anlayışını genişletir. Ancak tasarımcının süreci iyileştirmek için bir fikri varsa, bu hemen (o anda) müşteriye geri gönderilmelidir.

Domain Analizi

- Etki alanı analizi, tasarlanan sistemle ilgili ve rekabet eden uygulamaların "manzarasını" değerlendirmeye yönelik herhangi bir genel yaklaşımı içerir.
- Böyle bir yaklaşım, temel işlevlerin ve daha sonra eksik işlevlerin belirlenmesinde faydalı olabilir.
- Etki alanı analizi, daha sonra yeniden kullanılabilir bileşenleri (nihai tasarıma dahil edilebilecek açık kaynaklı yazılım öğeleri gibi) belirlemek için de kullanılabilir.

Etnografik Gözlem

- Etnografik gözlem, dolaylı ve doğrudan faktörlerin gözleminin gereksinim mühendisinin işini bilgilendirdiği herhangi bir tekniği ifade eder.
- Etnografik gözlem, sosyal bilimlerden ödünç alınan bir tekniktir; bu teknikte, insan faaliyeti ve işin gerçekleştiği çevre gözlemlerinin, bilim adamını bazı fenomenlerin çalışmasında bilgilendirmek için kullanıldığı bir tekniktir.

- Gereksinimlerin ortaya çıkarılmasına etnografik gözlem uygularken, gereksinim mühendisi, müşterinin işyeri kültürüne kendini kaptırır.
- Burada, otomatikleştirilecek işi veya faaliyeti gözlemlemeye ek olarak, gereksinim mühendisi aynı zamanda, doğrudan iletilemeyen, çevreden türetilen müşteri ihtiyaçlarının kanıtlarını toplayabilecek bir konumdadır.
- Çıracı olarak tasarımcı, etnografik gözlem etkinliğini içeren bir gereksinim ortaya çıkarma tekniğidir.
- Bu tekniği pratikte göstermek için, etnografik gözlemin meydana geldiği şu durumu göz önünde bulundurun:

- Bir müşteri için akıllı bir ev için gereksinimleri topluyorsunuz.
- Müşteriyle ne istedikleri hakkında röportaj yapmak için uzun zaman harcıyorsunuz.
- Müşteriler günlerini sürdürürken ve sorular sorarken onlarla etkileşim kurmak için zaman harcarsınız (“bulaşık makinesini neden gece çalıştırıyorsunuz, neden sabah çalıştırmıyorsunuz?”).
- İstekler ve arzular hakkında sözsüz ipuçları almak için mevcut evde müşteriye “hareket halindeyken” pasif bir şekilde gözlemleyerek uzun zaman harcıyorsunuz.
- Evin kendisinden başka bilgiler edinirsiniz - kitaplıktaki kitaplar, duvardaki tablolar, mobilya stilleri, hobilerin kanıtı, çeşitli cihazlarda aşınma ve yıpranma belirtileri vb.

Hedefe Dayalı Yaklaşımlar

- Hedefe dayalı yaklaşımlar, gereksinimlerin, gereksinimlere yol açan bir dizi hedef yoluyla misyon bildiriminden kaynaklandığı kabul edilen herhangi bir ortaya çıkarma tekniğini içerir.
 - Bu hedefler, daha düşük seviyeli hedeflere ulaşmak için bir veya daha fazla kez alt bölümlere ayrılabilir.
 - Daha sonra, alt düzey hedefler belirli üst düzey gereksinimlere ayrılır.
 - Son olarak, üst düzey gereksinimler daha düşük düzeyli gereksinimler oluşturmak için kullanılır.
- Örneğin, bagaj taşıma sistemi misyon beyanını göz önünde bulundurun
 - Amaç yolcu kalkış noktasından varış noktasına kadar bagaj taşımanın tüm yönlerini otomatikleştirmek.
 - Bu misyonun yerine getirilmesi için aşağıdaki hedefler düşünülebilir:
 - Hedef 1: Check-in'den teslim alınmasına kadar bagaj takibini tamamen otomatik hale getirmek.
 - Hedef 2: Bagajın check-in kontuarından uçağa yönlendirilmesini tamamen otomatikleştirmek
 - Hedef 3: Kayıp bagaj miktarını %1'e düşürmek

- Son olarak, bu metrikler için kabul edilebilir aralıklar belirlemek üzere müşteriyle birlikte çalışırız. Sistem oluşturulduktan sonra, gerçek kullanıcılarla test denemeleri yoluyla gereksinimlerin tatmini gösterilebilir.
- Kullanıcıların özellikleri ve sayısı, test süresi vb. gibi bu testin parametreleri daha sonra tanımlanabilir ve sistem test planına dahil edilebilir. Bu şekilde kabul edilebilir bir kullanım kolaylığı seviyesi tanımlayabiliriz.

Grup Çalışması

- Grup çalışması, ihtiyaç keşfi, analizi ve takibi süreçlerinde kullanılan her türlü grup toplantılarının genel adıdır. Gereksinimlerin ortaya çıkarılması için grup odaklı çalışmaların en ünlüsü, birazdan tartışacağımız ortak uygulama tasarımıdır (JAD).
- Grup faaliyetleri, birçok paydaşı bir araya getirme açısından çok verimli olabilir, ancak çatışma ve bölünme potansiyelini riske atabilir. Her türlü grup çalışmasında başarının anahtarı, grup toplantılarının planlanması ve yürütülmesidir. İşte grup toplantıları hakkında hatırlanması gereken en önemli şeyler.

- Ödevini yap—kuruluşun tüm yönlerini, sorunları, siyaseti, çevreyi vb. araştırın.
- Toplantı gerçekleşmeden birkaç gün önce bir gündem (her madde için ayrılan süre ile) yayınlayın.
- Toplantı boyunca gündemde kalın (toplantı kapsamı kayması yok).
- Elinizde özel bir not alıcı (yazıcı) bulundurun.
- Kişisel sorunların araya girmesine izin vermeyin.

- Herkesin sesini duyurmasına izin verin.
- İlk fırsatta fikir birliği arayın.
- Gündemdeki tüm maddeler yeterince tartışılana kadar ayrılmayınız.
- Toplantı tutanaklarını toplantı kapanışından sonraki birkaç gün içinde yayınlayın ve katılımcıların değişiklik önermesine izin verin.
- Bu ilkeler, gereksinimlerin ortaya çıkarılmasına yönelik JAD yaklaşımı için devreye girecektir.

Toplantılar

Toplantılar yoluyla ortaya çıkarma, iki kiři arasındaki yüz yüze iletişimi içerir.

Toplantılar, sistem düzeyinde gereksinimleri ortaya çıkarmak için kullanımı kolay bir tekniktir.

Ortaya çıkarma faaliyetlerinde kullanılabilir üç tür görüşme vardır ve bireylere veya odak gruplarına uygulanabilirler:

- Yapılandırılmamış
- Yapılandırılmış
- Yarı yapılandırılmış

Muhtemelen en yaygın tür olan yapılandırılmamış görüşmeler, doğası gereği konuşma ve katılımcıları rahatlatmaya hizmet eder. Gereksinimler ne zaman olursa olsun, herhangi bir zamanda ve herhangi bir yerde ortaya çıkabilir. Mühendis ve paydaş bir aradadır.

- ▶ Ancak görüşmecinin becerisine baęlı olarak, yapılandırılmamış görüşmeler başarılı veya başarısız olabilir. Bu nedenle, yapılandırılmış veya yarı yapılandırılmış toplantı tercih edilir.
- ▶ Yapılandırılmış görüşmeler, doğası gereęi çok daha resmidir ve önceden tanımlanmış özenle planlanmış sorulardır. Şablonlar, yapılandırılmış stili kullanarak toplantılarda çok faydalı olur. Yapılandırılmış görüşmelerin en büyük dezavantajı, biçim çok kontrollü olduęu için bazı müşteriler bilgileri saklayabilir.

- ▶ Yani, gereksinim mühendisi dikkatlice düşünölmüş bir liste hazırlar ancak daha sonra kendiliğinden yapılandırılmamış soruların görüşme sırasında dahil olmasına izin verir.
- ▶ Seçim bağlama göre yapılmalıdır. Örneğın, müşterinin kurumsal kültürü çok resmi olmayan ve rahat ve güven yüksekse, yapılandırılmamış görüşmeler olabilir. Daha karmaşık, süreç odaklı bir organizasyonda, yapılandırılmış ve yarı yapılandırılmış görüşmeler muhtemelen daha fazla tercih edilir.

- ▶ İşte üç görüşme türünden herhangi birinde kullanılabilecek bazı örnek görüşme soruları:
 - ▶ ■ Sistemin önemli bir özelliği nedir?
 - ▶ ■ 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 tekniđi kullanılırsa kullanılsın, tüm dođru soruların sorulduđundan emin olmak için özen gösterilmelidir. Yani, önemli soruları atlamayın ve rahatsız edici veya gereksiz soruları dahil etmeyin. Kesinlikle gerekli olduđunda, görüşmeler telefon, video konferans veya e-posta yoluyla yapılabilir.

İç Gözlem

- ▶ 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...”
- ▶ Gereksinim mühendisinin alan bilgisi müşterininkinden çok daha fazla olduğunda, içe dönük bir yaklaşım yararlıdır. İç gözlem yapılırken bilgilendirme yapılmalı ancak müşterinin isteklerine müdahale edilmemelidir.

Ortak Uygulama Geliştirme

- ▶ Ortak uygulama geliştirme (JAD), işletme kullanıcılarının ve bilgisayar mühendislerinin bir bilgisayar sistemi oluşturmak için birlikte çalıştıkları bir yazılım geliştirme yaklaşımıdır. Bu toplantılar günde 4-8 saat ve 1 günden birkaç haftaya kadar süren bir süre içinde gerçekleşir. Her boyuttaki sistemler için kullanılır.
- ▶ JAD ve JAD benzeri teknikler, problemler, hedefler ve gereksinimler üzerinde grup bütünlüğünü sağlamak için sistem planlama ve sistem analizi faaliyetlerinde yaygın olarak kullanılır.

- ▶ Bir JAD incelemesi veya denetim oturumu planlaması üç adımdan oluşur:
 - ▶ 1. Katılımcıların seçilmesi
 - ▶ 2. Gündemin hazırlanması
 - ▶ 3. Bir yer seçme

- ▶ İncelemeler ve denetimler, aşağıdaki katılımcıların bir kısmını veya tamamını içerebilir:
 - ▶ ■ Sponsorlar (ör. Üst yönetim)
 - ▶ ■ Bir ekip lideri (kolaylaştırıcı)
 - ▶ ■ Gereksinimlerin ve iş kurallarının sahibi olan kullanıcılar ve yöneticiler
 - ▶ ■ Katipler (yani toplantı tutanakları ve not tutanlar)
 - ▶ ■ Mühendislik personeli

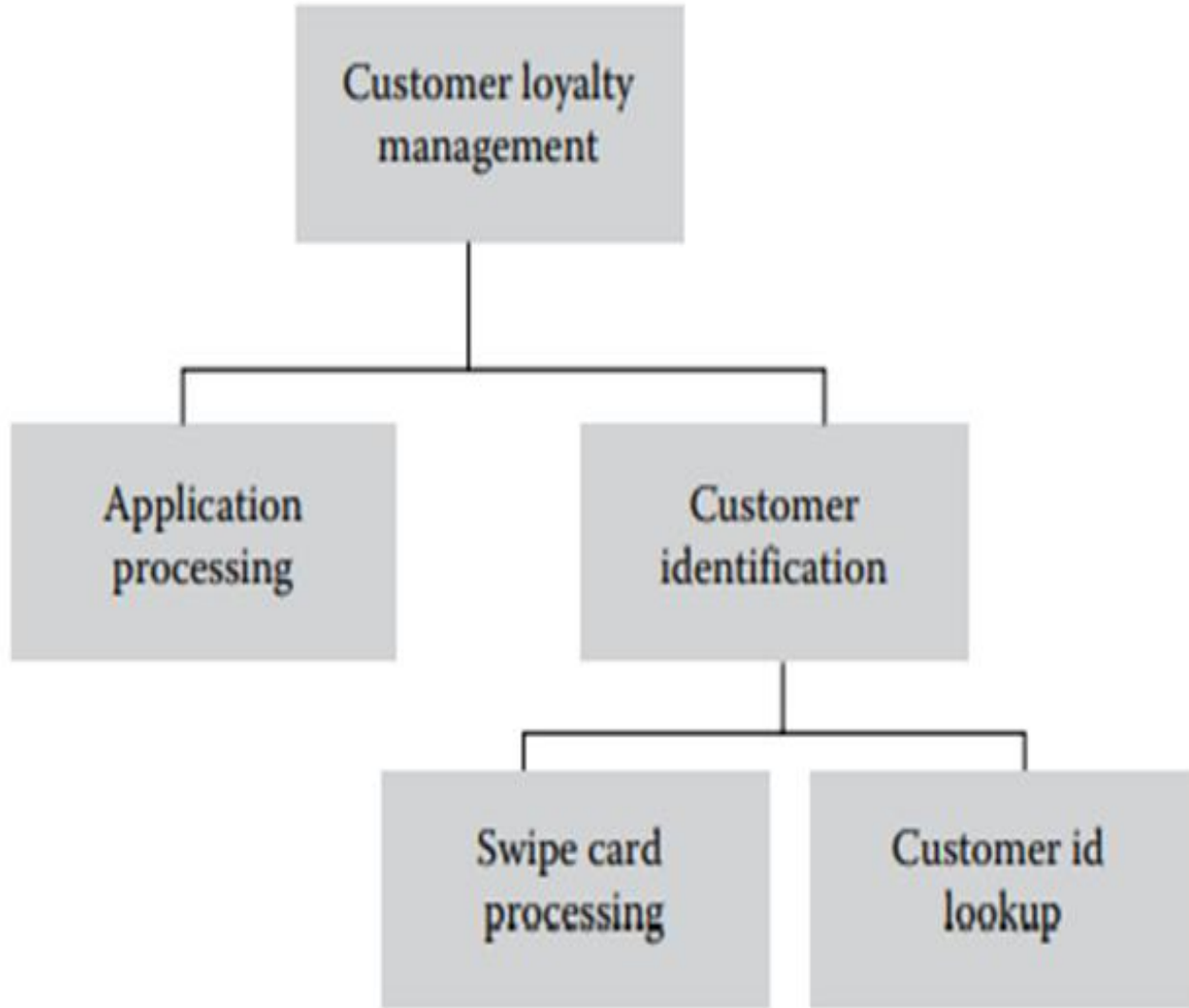
- ▶ Sponsor, analistler ve yöneticiler bir lider seçer. Lider kurum içi veya danışman olabilir. Normalde yazılım geliştirme ekibinden bir veya daha fazla yazar (not tutucu) seçilir. Analist ve yöneticiler, kullanıcı topluluğundan bireyler seçmelidir.
- ▶ Bir oturum planlamadan önce, analist ve sponsor projenin kapsamını belirlemeli ve her oturumun üst düzey gereksinimlerini ve beklentilerini belirlemelidir.
- ▶ Gündem, büyük ölçüde yürütülecek incelemenin türüne bağlıdır ve yeterli zamana izin verecek şekilde oluşturulmalıdır. Gündem, kod ve belgeler tüm katılımcılara onları gözden geçirmek, yorum yapmak ve soru sormaya hazırlanmak için yeterli zamana sahip olmaları için toplantıdan çok önce gönderilmelidir.

- ▶ Oturum lideri, bu uygulamaların uygulanmasını sağlamak için Őu maddeleri göz önüne almalıdır:
- ▶ ■ Gündeme baēlı kalın.
- ▶ ■ Programa baēlı kalın
- ▶ ■ Yazıcının not alabildiēinden emin olun.
- ▶ ■ Teknik jargondan kaçının (incelemede teknik olmayan personel yoksa).
- ▶ ■ ÇatıŐmaları çözüñ
- ▶ ■ Grup fikir birliēini teŐvik edin.
- ▶ ■ KiŐilere izin vermeden kullanıcı ve yönetim katılımını teŐvik edin.
- ▶ ■ Toplantıyı kiŐisel olmayan tutun.
- ▶ ■ Toplantıların gerektiēi 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.

Merdivenleme

- ▶ Merdivenlemede, gereksinim mühendisi, gereksinimleri ortaya çıkarmak için müşteriye kısa yönlendirici sorular sorar. Daha sonra yüzeyin altında daha derine inmek için takip eden sorular sorulur.
- ▶ Tekniği göstermek için, evcil hayvan mağazası POS sistemi için aşağıdaki merdiven sorularını ve yanıtlarını göz önünde bulundurun. “RE” gereksinim mühendisini ifade eder.
- ▶ RE: Sistemin önemli bir özelliğini adlandırın.
- ▶ Müşteri: Müşteri kimliği.
- ▶ RE: Bir müşteriyi nasıl tanımlarsınız?
- ▶ Müşteri: Sadakat kartlarını.
- ▶ RE: Bir müşteri kartını unutursa ne olur?
- ▶ Müşteri: Telefon numarasından aranabilirler.
- ▶ RE: Müşterinin telefon numarasını ne zaman alırsınız?
- ▶ Müşteri: Müşteriler sadakat kartı başvurusunu tamamladığında.
- ▶ RE: Müşteriler başvuruları nasıl tamamlıyor? ...Ve bunun gibi.



Şekil 3.3: Pet shop POS sistemi için merdiven şeması

- Şekil 3.3, sorulara verilen yanıtların bir merdiven veya hiyerarşik diyagramda nasıl organize edildiğini gösterir. Merdivenleme tekniği, bilginin hiyerarşik bir biçimde düzenlenebileceğini varsayar veya en azından bilginin hiyerarşik olarak düzenlenmesine neden olur.

Protokol Analizi

- ▶ Protokol analizi, müşterilerin gereksinim mühendisleriyle birlikte otomatikleştirecekleri prosedürleri gözden geçirdikleri bir süreçtir.
- ▶ Bölüm 1'deki bagaj uygulaması, yalnızca teslim edilecek paketlerdeki dalgalanmayı ele almakla kalmadı, aynı zamanda mühendisleri, şirketin hizmetleriyle ilgili süreçler ve prosedürler hakkında, gerçekte uygulandıkları şekliyle yeniden inşa etti.
- ▶ Birazdan protokol analizinin çırak olarak tasarımcıya çok benzediğini göreceksiniz, ancak ince farklılıklar var. Bu farklılıklar, çırak olarak tasarımcıdan ziyade protokol analizinde daha pasif olan gereksinim mühendisinin rolünde yatmaktadır.

Prototipleme

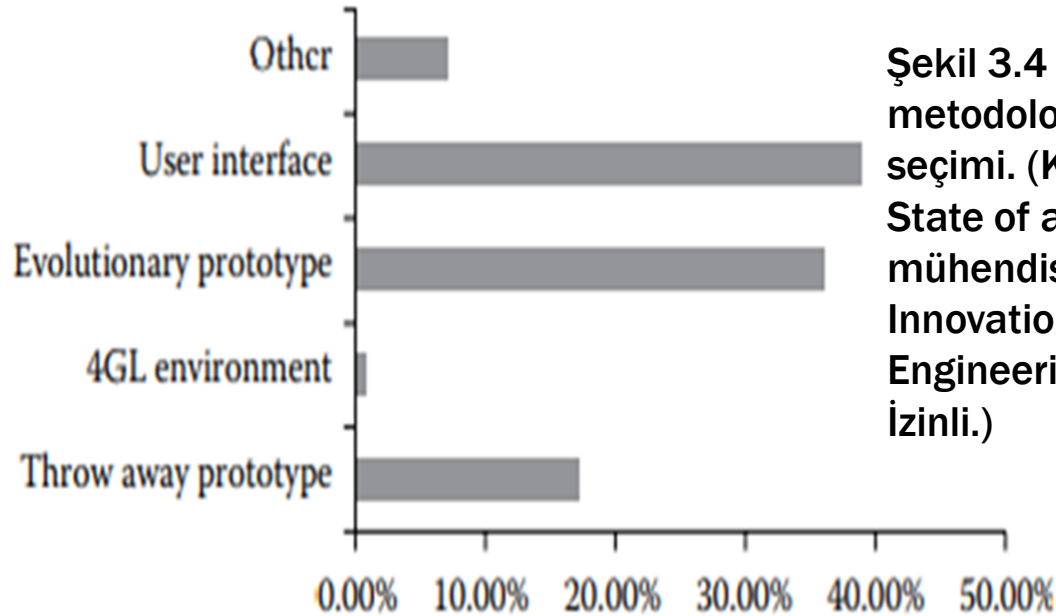
- ▶ Prototipleme, özellikle kullanılabilirlik gereksinimleri olmak üzere yeni özellikleri keşfetmek için sistem modellerinin oluşturulmasını içerir. Prototipleme, gereksinimlerin ortaya çıkarılması için özellikle önemli bir tekniktir.
- ▶ Prototipler, çalışan modelleri ve çalışmayan modelleri içerebilir. 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 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 (atılmadan) tasarlanabilir. Örneğin, grafiksel kullanıcı arayüzü kod maketleri gereksinimlerin ortaya çıkarılması için kullanılabilir ve kod yeniden kullanılabilir. Çevik yazılım geliştirme metodolojileri, atılmayan prototipleri sürekli olarak geliştirme 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, yaygın olarak kullanılan bilgisayar destekli tasarım (CAD) programları tarafından üretilen standart formattaki dosyaları girdi olarak alabilmesidir. (Berman 2012) Prototiplemeyi kullanmanın birkaç farklı yolu vardır - örneğin, dördüncü nesil bir ortamda (yani bir simülatörde), kullanılıp atılan prototipleme, evrimsel prototipleme ve kullanıcı arayüzü prototiplemesi. Bazı kuruluşlar birden fazla prototipleme türü kullanabilir. 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:
- ▶ 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.
- ▶ İkinci sorun, prototiplemenin belirli işlevsel olmayan gereksinimleri keşfetmede her zaman etkili olmamasıdır(Kassab ve Ormandjieva 2014).

- Son olarak, kullanıcıların sistemle etkileşim kurma yollarını keşfetmek için prototipler kullanılırken sorunlar ortaya çıkabilir. Ana endişe, kullanıcıların bir prototiple gerçek sistemle farklı şekilde etkileşime girmesidir. Örneğin sürücüler simülatörde gerçek bir araçta olduğundan çok daha agresif davranır ve muhtemelen hatalı gereksinimlerin keşfedilmesine yol açar.



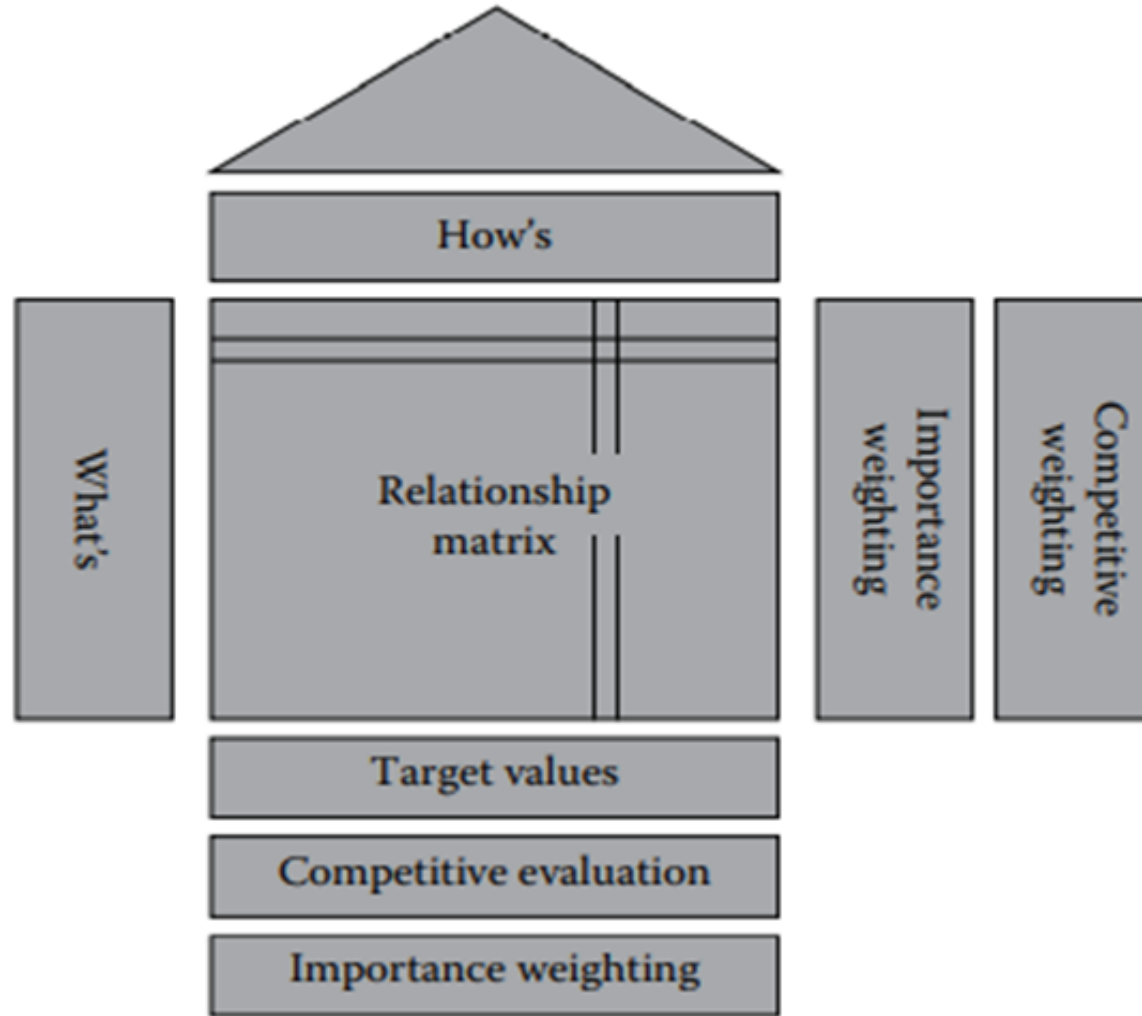
Şekil 3.4 Yazılım geliştirme yaşam döngüsü metodolojisi boyunca prototip yöntemleri seçimi. (Kassab, M., Neill, C & Laplante, M. State of application in gereksinim mühendisliği: çağdaş verilerden uyarlar. Innovations in Systems and Software Engineering, 10, no. 4 (2014): 235-244. İzinli.)

Kalite Fonksiyonu Dağıtımı

- ▶ Kalite fonksiyon dağıtımı (QFD), müşteri gereksinimlerini keşfetmeye ve üretim aşaması boyunca kullanılacak ana kalite güvence noktalarını tanımlamaya yönelik bir tekniktir.
- ▶ QFD'nin temel fikri, müşteri ihtiyaçları, teknik gereksinimler, öncelikler ve (gerekirse) rakip değerlendirmesi arasında ilişki matrisleri oluşturmaktır. Bu ilişki matrisleri genellikle bir evin çatısı, tavanı ve yanları olarak temsil edildiğinden, QFD bazen “kalite evi” olarak anılır (Şekil 3.5; Akao 1990).

- ▶ QFD, 1966 yılında Yoji Akao tarafından imalat, ağır sanayi ve sistem mühendisliğinde kullanılmak üzere tanıtıldı.
- ▶ Aşağıdaki gereksinim mühendisliği süreci QFD tarafından belirlenir.
- ▶ ■ Paydaşların niteliklerini veya gereksinimlerini tanımlayın.
- ▶ ■ 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, rakip ürünler için ilgili tüm özelliklerin birleşiminden bir özellik listesi oluşturulur. 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, yeni sistemdeki temel özelliklerin atlanmamasını sağlamaya yardımcı olur ve istenen gereksinimlerin eksiksizlik kalitesinin iyileştirilmesine katkıda bulunabilir.



Şekil 3.5 QFD'nin "kalite evi" (Akao 1990)

- ▶ 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.
- ▶ Gereksinim mühendisliğine toplam yaşam döngüsü yaklaşımını dahil ettiğinden, QFD'nin diğer bağımsız ortaya çıkarma tekniklerine göre birçok avantajı vardır. QFD, kullanıcıların ve yöneticilerin katılımını geliştirir. 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.
- ▶ 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.

<i>Feature</i>	<i>Competing Product</i>		
	<i>MyFavoritePet</i>	<i>BestFriends</i>	<i>Fido-2.0</i>
Maximum simultaneous users supported	100	250	Unlimited
Wireless device support	Yes	Yes	Yes
Business analytics features	Yes	Yes	No
Operating system support	Windows/ Mac/Linux	Windows/Linux	Windows/Mac
Cost (base system) (\$K)	50	110	75

Tablo 3.1 Pet Mağaza Satış Noktası Sistemi için Kısmi Rekabet Analizi

► ANKETLER

- Gereksinim mühendisleri genellikle anketleri ve diğer anket araçlarını geniş paydaş gruplarına ulaşmak için kullanır. Anketler genellikle kapsam sınırlarını hızlı bir şekilde tanımlamak için ortaya çıkarma sürecinin erken aşamalarında kullanılır.
- Örneğin, evcil hayvan mağazası POS sistemi için olası bazı anket soruları şunlardır:
 - ■ Envanterinizde kaç tane benzersiz ürün taşıyorsunuz?
 - ■ Kaç farklı depo siteniz var? _____
 - ■ Kaç farklı mağaza konumunuz var? _____
 - ■ Şu anda kaç benzersiz müşteriniz var? _____

- ▶ Anket alıřması teknikleri alan paydařlar ve gereksinim mhendisileri tarafından da ok iyi anlařıldığında kullanıřlıdır.
- ▶ Byk lekli anketler yapmadan nce amalanan anket poplasyonunun kk bir alt kmesi ile bir pilot alıřma yapmak nemlidir.
- ▶ Anket verilerini analiz ederken, zellikle katılımcılardan kimliklerini belirlemelerini isterken ve istenen zellikleri sıralarken, ařađıdaki etkiye dikkat edin.

- ▶ Gerçekleştirilmesi gerekmeyen bir dizi seçenek verildiğine; kişi, eğer karar verildiyse çok daha fazla sayıda seçeneği arzu etme eğiliminde olacaktır.
- ▶ Aşağıdaki örnekten dolayı bu etkiyi “dondurma marketi etkisi” olarak adlandırıyoruz. El yapımı bir dondurma dükkanı açmaya karar veren bir girişimci düşünün. Ürün araştırmasının bir parçası olarak, bir enstrümanla birkaç kişiye anket yapıyor yanıtlayanların satın alacakları dondurma lezzetlerini kontrol ediyor.
- ▶ Ankette listelenen 30 farklı lezzetten 20'sinin ankete katılanların %50'si veya daha fazlası tarafından seçildiğini öğreniyor. Böylece anket sonuçlarına göre bu 20 aromayı kabaca ankette belirtilen taleple orantılı olarak stoklayıp üretmeye karar verir. Yine de, dondurma dükkanını açtıktan 1 hafta sonra, işinin %90'ını çikolata, vanilya ve çilek aromalı satışların oluşturduğunu keşfediyor. Envanterinde tuttuğu diğer 17 lezzetten birkaçı daha önce hiç satılmamıştı. Ankette müşterilerin satın alacaklarını söylemelerine rağmen seçimlerini yapma zamanı geldiğinde, farklı davrandılar. Bu nedenle, müşterilere özellik kümeleriyle ilgili seçim sunarken dondurma dükkanının etkisini unutmayın. Bir şey söylerler ama başka bir şey yaparlar.

► Repertuar Izgaraları

- Repertuar ızgaraları, çeşitli özellikler için yapılandırılmış sistemdeki farklı varlıklardır ve genellikle müşteriler etki alanı uzmanları olduğunda kullanılan bir sıralama sistemi içerir. Repertuar ızgaraları, özellikle paydaş grupları içindeki anlaşma ve anlaşmazlığın belirlenmesi için yararlıdır.

- ▶ Sütunlar, paydaşların her birine dayalı sıralamaları temsil eder.
- ▶ Tekniği göstermek için, Şekil 3.6 çeşitli bagaj taşıma sisteminin nitelikleri için bir repertuar ızgarasını temsil eder.

Baggage handling speed	1	1	5
Fault-tolerance	4	5	5
Safety	5	4	4
Reliability	3	5	5
Ease of maintenance	3	5	5

Airline worker's union rep
Maintenance engineer
Airport operations manager

- ▶ Burada, havalimanı operasyon müdürü için tüm niteliklerin esasen en yüksek öneme sahip olduğunu görüyoruz.
- ▶ Ancak Havayolu İşçileri Sendikası temsilcisi için güvenlik en önemli şeydir (sonuçta sendika üyeliğinin sistemle günlük olarak etkileşime girmesi gerekir).
- ▶ Özünde, bu derecelendirmeler, paydaşların gündemlerini veya farklı bakış açılarını yansıtır. Bu nedenle, paydaş hedeflerini içeren anlaşmazlıklarla erken yüzleşmede repertuar ızgaralarının kullanımının neden çok yararlı olabileceğini görmek kolaydır.

- ▶ Ayrıca, ızgaralar paydaşların tutumlarını yakaladıkları için sistemin geliştirilmesi ve daha sonra nitelikler ve özellikler hakkında anlaşmazlıklarla başa çıkmak için değerli belgeler sağlayabilir.
- ▶ Yine de, repertuar ızgaralarını kullanırken, dondurma dükkanının etkisini hatırlayın—paydaşlar halka açık bir ortamda bir şey söyleyecektir ve daha sonra farklı davranacaklardır.

▶ Senaryolar

- ▶ Senaryolar, kullanımda olan sistemin üst düzey bir performans sağlayan sistem çalışmasının tanımı, kullanıcı sınıfları ve istisnai durumların resmi olmayan açıklamalarıdır.

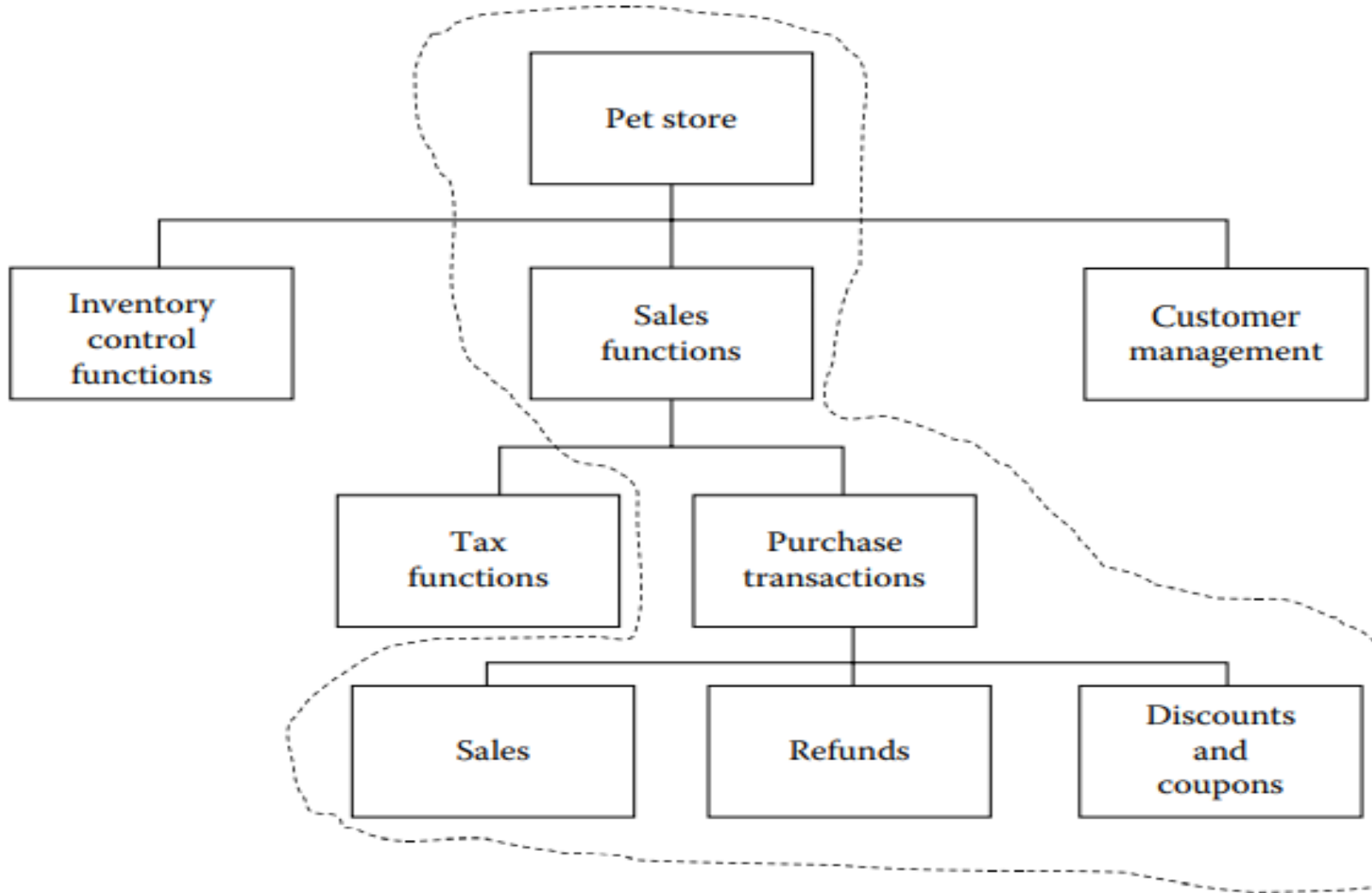
- ▶ İşte pet shop POS sistemi için örnek bir senaryo.
- ▶ Bir müşteri evcil hayvan dükkânına girer ve arabayı çeşitli ürünlerle doldurur.
- ▶ Kontrol ederken, kasiyer müşterinin sadakat kartı olup olmadığını sorar.
- ▶ Eğer öyleyse, kasiyer kartı okutarak müşterinin kimliğini doğrular.
- ▶ Değilse, kasiyer bir tanesini yerinde tamamlamayı teklif eder.
- ▶ Sadakat kartı faaliyetinden sonra, kasiyer bir kod okuyucu kullanarak ürünleri tarar.
- ▶ Her ürün tarandıkça, satış toplanır ve envanter uygun şekilde güncellenir. Ürün taramasının tamamlanmasının ardından bir ara toplam hesaplanır. Daha sonra herhangi bir kupon ve indirim girilir.

- ▶ Yeni bir ara toplam hesaplanır ve geçerli vergiler eklenir. Bir makbuz yazdırılır ve müşteri nakit, kredi kartı, banka kartı veya çek ile ödeme yapar. Tüm uygun toplamalar (satışlar, vergiler, indirimler, indirimler vb.) hesaplanır ve kaydedilir.

▶ Görev Analizi

- ▶ Halihazırda incelemiş olduğumuz hiyerarşik yönelimli tekniklerin çoğu gibi, görev analizi, sistem tarafından gerçekleştirilecek görevlerin işlevsel bir şekilde ayrıştırılmasını içerir. Yani, en yüksek soyutlama düzeyinden başlayarak, tasarımcı ve müşteriler daha ileri düzeyde ayrıntı ortaya çıkarır. Bu detaylı ayrıştırma, en düşük işlevsellik düzeyi (tek görev) elde edilinceye kadar devam eder.

- Örnek olarak, evcil hayvan mağazası POS sistemi için kısmi görev analizini düşünün.
- Şekil 3.7'de gösterilmiştir.

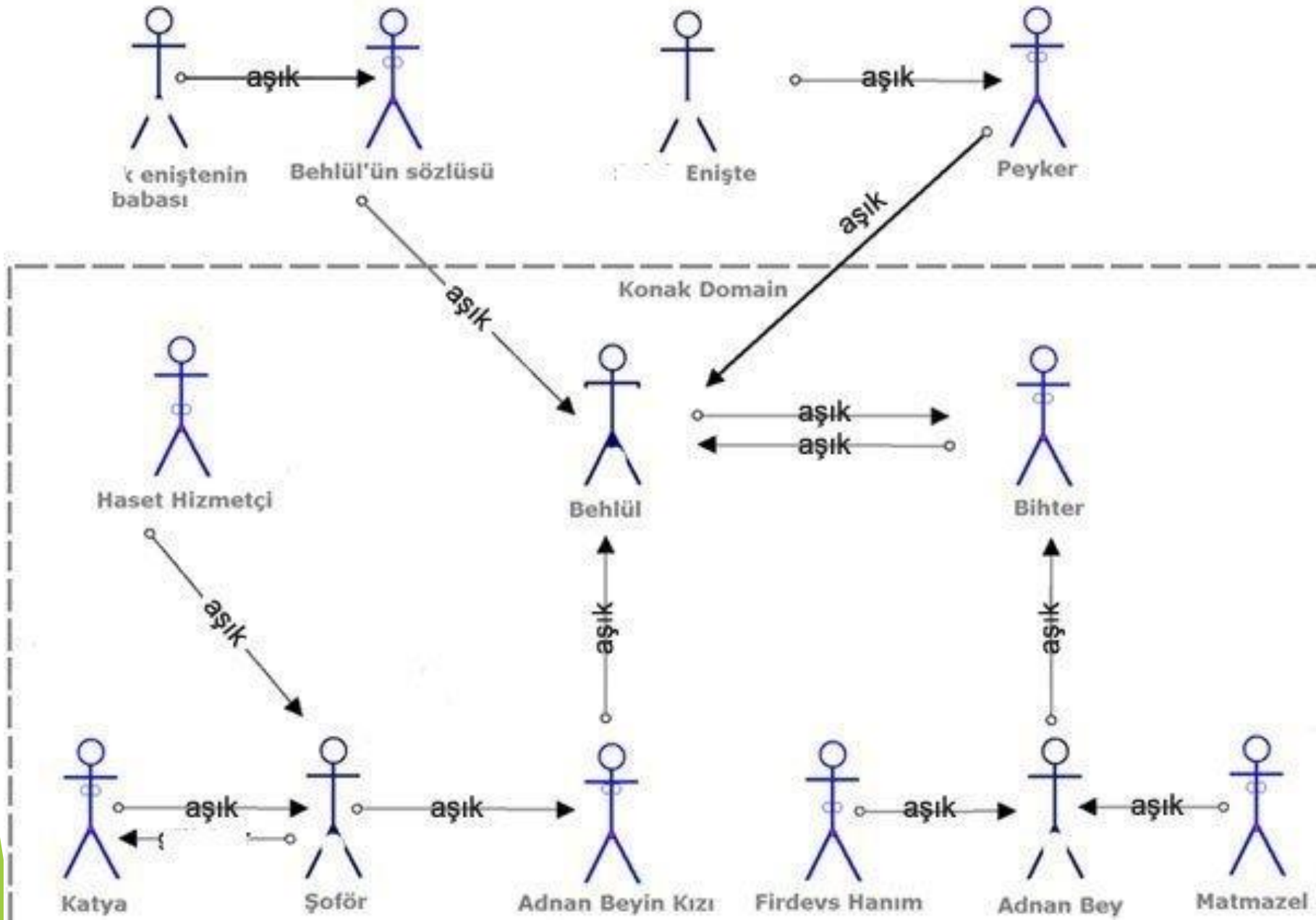


- ▶ Burada, kapsamlı evcil hayvan mağazası POS sisteminin üç ana parçadan oluştuğu kabul edilmektedir.
- ▶ görevler: envanter kontrolü, satış ve müşteri yönetimi. Altında sondaj satış fonksiyonları, bunların aşağıdaki görevlerden oluştuğunu görüyoruz: vergi fonksiyonları ve satın alma işlemleri. Ardından satın alma işlemi fonksiyonuna geçilir, bu görevleri satış, geri ödeme, indirim ve kupon görevlerine ayırıyoruz.
- ▶ Görev analizi ve ayrıştırma, yeterli bir ayrıntı düzeyine ulaşılan kadar devam eder (tipik olarak, bir yöntem veya ayrıştırılamaz prosedür düzeyine kadar) ve diyagram tamamlanır.

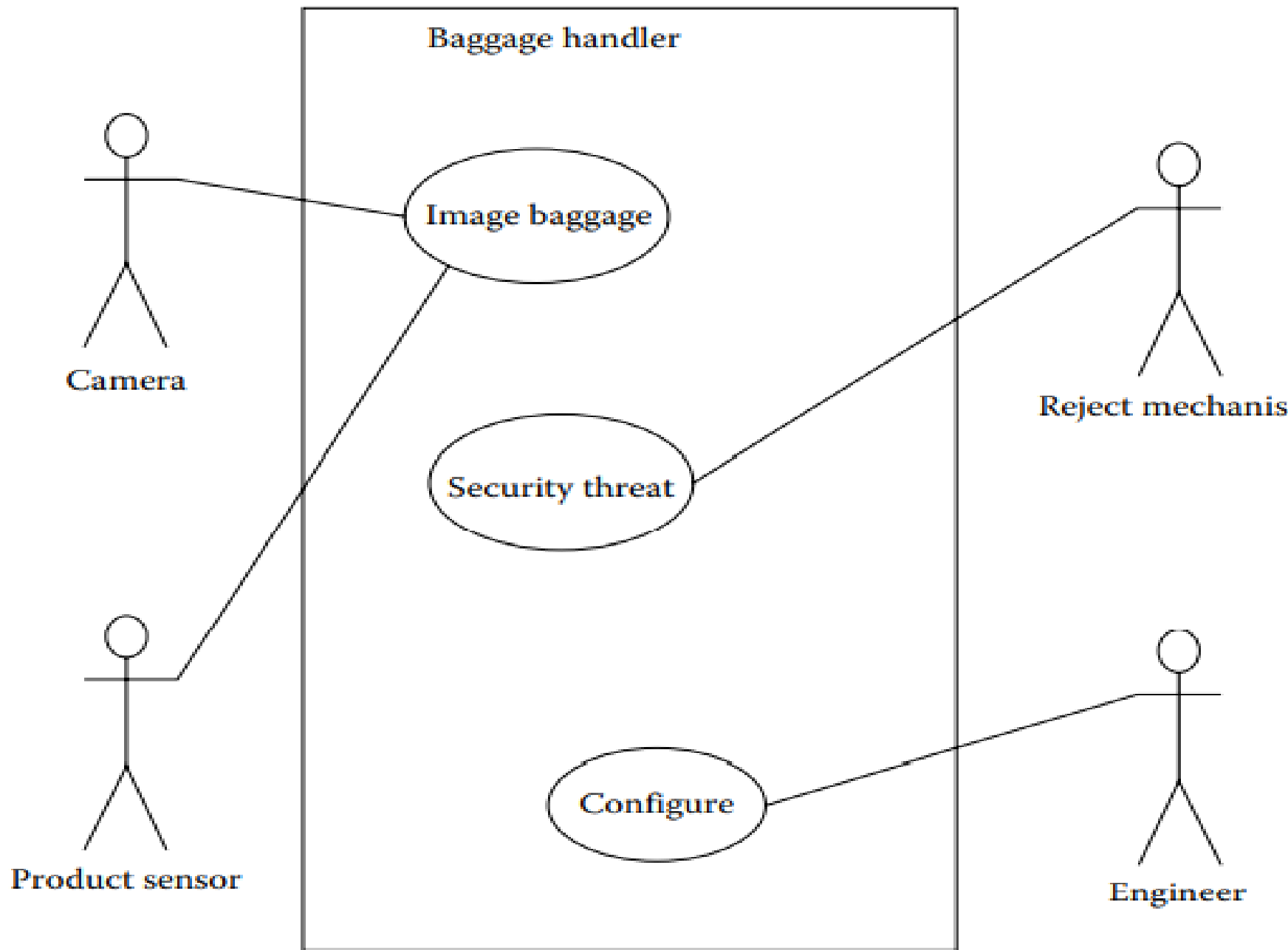
► Use Case'ler*

- Use caseler, daha karmaşık müşterilerin ve paydaşların ve onların isteklerinin tanımlaması için bir yoldur. Use caseler, sistem ve sistem arasındaki etkileşimleri, özellikle insan kullanıcılar ve diğer sistemler arasındaki etkileşimleri gösterir. Use caseler, tasarımcının bakış açısından sistemin çalışma senaryolarını açıklar.
- Use caseler, tipik olarak, sistemin dış ortamıyla etkileşimlerini gösteren bir use case diyagramı kullanılarak temsil edilir. Use case diyagramında kutu sistemin kendisini temsil eder.

AŞK-I MEMNU USE CASE



- ▶ Çubuk figürler, sistem ile etkileşime giren dış varlıkları belirleyen “aktörleri” temsil eder.
- ▶ Aktörler insanlar, diğer sistemler veya cihaz girdileri olabilir.
- ▶ İç elipsler, aktörlerin her biri için her bir kullanım etkinliğini temsil eder.
- ▶ Bu düz çizgiler, oyuncuları her kullanımla ilişkilendirir. Şekil 3.8, bagaj kontrol sistemi için bir kullanım durumu diyagramını göstermektedir.



Kullanıcı hikayeleri*

- ▶ Kullanıcı hikayeleri, ilk keşif ve projelendirme gereksinimleri için kullanılan kısa konuşma metinleridir.
- ▶ Kullanıcı hikayeleri çevik metodolojilerle birlikte yaygın olarak kullanılmaktadır.

- ▶ Kullanıcı hikayeleri, sistemin onların ihtiyacına göre müşteriler tarafından yazılır.
- ▶ Kullanıcı hikayeleri genellikle iki ila üçe beş inçlik bir karta yazılmış dört cümleden oluşur. Yaklaşık 80 kullanıcı hikayesi genellikle bir sistem artışı veya gelişimi için uygun, ancak kullanılacak metodoloji (örneğin, çevik ve artımlı) uygun sayı uygulama boyutuna ve kapsamına ve geliştirmeye bağlı olarak büyük ölçüde değişecektir.

- ▶ Pet shop POS sistemi için bir kullanıcı hikayesi örneği aşağıdaki gibidir:
- ▶ ■ Her müşteri bir kasada kolayca kontrol edebilmelidir.
- ▶ ■ Self servis desteklenecektir.
- ▶ ■ Tüm kuponlar, indirimler ve geri ödemeler bu şekilde ele alınmalıdır.
- ▶ Kullanıcı hikayeleri, yalnızca makul ölçüde düşük risk oluşturmak için yeterli ayrıntı sağlamalıdır.
- ▶ Hikayenin uygulanmasının ne kadar süreceğinin tahmini. zamanı geldiğinde uygulamak için, hikaye geliştiricileri müşteriyi buluşacak ve detaylar konuşulacaktır.

► Bakış açıları

- Bakış açıları, bilgileri farklı seçim bölgelerinden(bakış açılarından) organize etmenin bir yoludur.
- Örneğin, bagaj taşıma sisteminde
- Aşağıdaki paydaşların her biri için farklı sistemin perspektifleri:
 - ■ Bagaj taşıma personeli
 - ■ Gezginler
 - ■ Bakım mühendisleri
 - ■ Havalimanı yöneticileri
 - ■ Düzenleyici kurumlar

- ▶ Bu paydaşların her birinin ihtiyaçlarını ve çelişkileri tanıyarak bu bakış açılarıyla ortaya çıkan çatışmalar, çeşitli yaklaşımlar kullanılarak uzlaştırılabilir.
- ▶ her bakış açısında:
 - ▶ ■ Belirtimde kullanılan gösterimi tanımlayan bir temsil stili
 - ▶ ■ “Bakış açısının ele aldığı ilgi alanı” olarak tanımlanan bir alan
 - ▶ ■ Tanımlanmış tarzda ifade edilen bir sistem modeli olan bir spesifikasyon
 - ▶ ■ Sistemin nasıl oluşturulacağını ve kontrol edileceğini tanımlayan bir süreç modeline sahip bir iş planı.
 - ▶ ■ Yapım, kontrol, denetim, ve spesifikasyonun değiştirilmesi
- ▶ Bakış açısı analizi genellikle önceliklendirme, anlaşma ve gereksinimleri sıralama için kullanılır.

► Atölyeler

- En genel düzeyde, çalıştaylar, gereksinimlerin sorunlarını çözmek için herhangi bir paydaş toplantısıdır.
- Çalıştayları iki tür olarak ayırt edebiliriz: resmi ve gayri resmi.
- Resmi çalıştaylar iyi planlanmış toplantılardır ve genellikle “teslim edilebilir” etkinliklerdir. Sözleşme ile zorunlu kılınanlardır.
- Gayri resmi çalıştaylar genellikle yüksek düzeyde yapılandırılmış toplantılardan daha az sıkıcıdır.

- ▶ **İşlevsel Olmayan Gereksinimlerin Ortaya Çıkarılması***
- ▶ İşlevsel olmayan gereksinim (NFR) ortaya çıkarma teknikleri, işlevsel gereksinimleri ortaya çıkarma tekniklerinden farklıdır.
- ▶ NFR'ler genellikle gereksinim analizi sırasında ve yazılım geliştirme sürecinde gayri resmi olarak belirtilir, genellikle çelişkilidir ve uygulanması zordur.

- ▶ Geliştirme sürecinin aşamaları: ortaya çıkarma, dokümantasyon, yönetim ve ölçek
- ▶ NFR ortaya çıkarma yöntemlerinin özellikleri:
 - ▶ 1. Daha az deneyimli personel tarafından yöntem kullanımını kolaylaştırmak için rehberli bir süreç ve sonuçların tekrarlanabilirliğini desteklemek için
 - ▶ 2. Kalite güvencesini kolaylaştırmak için ölçülebilir NFR'lerin türetilmesi
 - ▶ 3. Desteklemek için türetilmiş NFR'lerin eksiksizliğini desteklemek için eserlerin yeniden kullanımını öğrenmek ve tekrar çalışmaktan kaçınmak
 - ▶ 4. Gizli gereksinimleri de yakalamak ve böylece eksiksizliği desteklemek için kalitenin sezgisel ve yaratıcı şekilde ortaya çıkarılması
 - ▶ 5. Etkin ortaya çıkarma için odaklanmış çaba ve desteklemek için NFR önceliklendirmesi
 - ▶ 6. Takas kararlarını desteklemek için NFR'ler arasındaki bağımlılıkların ele alınması

- ▶ 7. NFR'lerin fonksiyonel gereksinimlerle entegrasyonu
- ▶ NFR'leri keşfetmenin yaygın yolları, sistem niteliklerinin rekabetçi analizini içerir: NFR'ler, rekabet halindeki ürünlerin niteliklerini analiz ederek keşfedilebilir.
- ▶ Örneğin, rakip bir ürün için tepki süresi nedir? Ve daha iyisini mi yapmamız gerekiyor?
- ▶ NFS'yi keşfetmek için başka bir teknik, önceden oluşturulmuş bir gereksinim mühendisinin paydaşlardan ve geliştirme ekibinden sorulacak bir anket kullanmaktır.
- ▶ Örneğin: “Sistem aşağıdakilere nasıl yanıt vermeli? giriş hataları? Sistemin hangi bölümleri daha sonra değiştirilmeye aday olabilir?
- ▶ Sistemin hangi verileri güvenli olmalı?” Bu sorular, odaklanmak ve sormak için bir şablon veya standart izlenerek hazırlanabilir.

- ▶ 7. NFR'lerin fonksiyonel gereksinimlerle entegrasyonu
- ▶ NFR'leri keşfetmenin yaygın yolları, sistem niteliklerinin rekabetçi analizini içerir: NFR'ler, rekabet halindeki ürünlerin niteliklerini analiz ederek keşfedilebilir.
- ▶ Örneğin, rakip bir ürün için tepki süresi nedir? Ve daha iyisini mi yapmamız gerekiyor?
- ▶ NFS'yi keşfetmek için başka bir teknik, önceden oluşturulmuş bir gereksinim mühendisinin paydaşlardan ve geliştirme ekibinden sorulacak bir anket kullanmaktır.
- ▶ Örneğin: “Sistem aşağıdakilere nasıl yanıt vermeli? giriş hataları? Sistemin hangi bölümleri daha sonra değiştirilmeye aday olabilir?
- ▶ Sistemin hangi verileri güvenli olmalı?” Bu sorular, odaklanmak ve sormak için bir şablon veya standart (örneğin, ISO 9126) izlenerek hazırlanabilir.

- ▶ **Gereksinimlerin Hangi Çıkarma Teknikleri Kombinasyonu Kullanılmalı?**
- ▶ Uygun bir gereksinimleri ortaya çıkarma tekniği seçimi konusunda rehberlik sağlayacak çok az araştırma var.
- ▶ Uygun ortaya çıkarma teknikleri hakkında biraz rehberlik sağlamak için öncelikle daha önce tartışılan teknikleri kategorilere gruplandırın.
- ▶ Sınıflar (röportajlar, alan odaklı, grup çalışması, etnografi, prototip oluşturma, hedefler, senaryolar, bakış açıları) ve dahil edilen ortaya çıkarma teknikleri tablo 3.2'de gösterilmiştir.

Teknik Türü	Teknikler
Etki Alanı Odaklı	Kart sıralama, Çırac olarak tasarımcı, Alan analizi, Merdiven, Protokol analizi, Görev Analizi
Etnografya	Etnografik gözlem
Hedefler	Hedefe dayalı yaklaşımlar
Grup çalışması	Beyin fırtınası, Grup çalışması, Atölyeler
Röportajlar	Röportajlar
Prototipleme	Prototipleme
Senaryolar	Senaryolar, Use cases Kullanıcı hikayeleri
Bakış açıları	Bakış açıları Repertuar ızgaraları

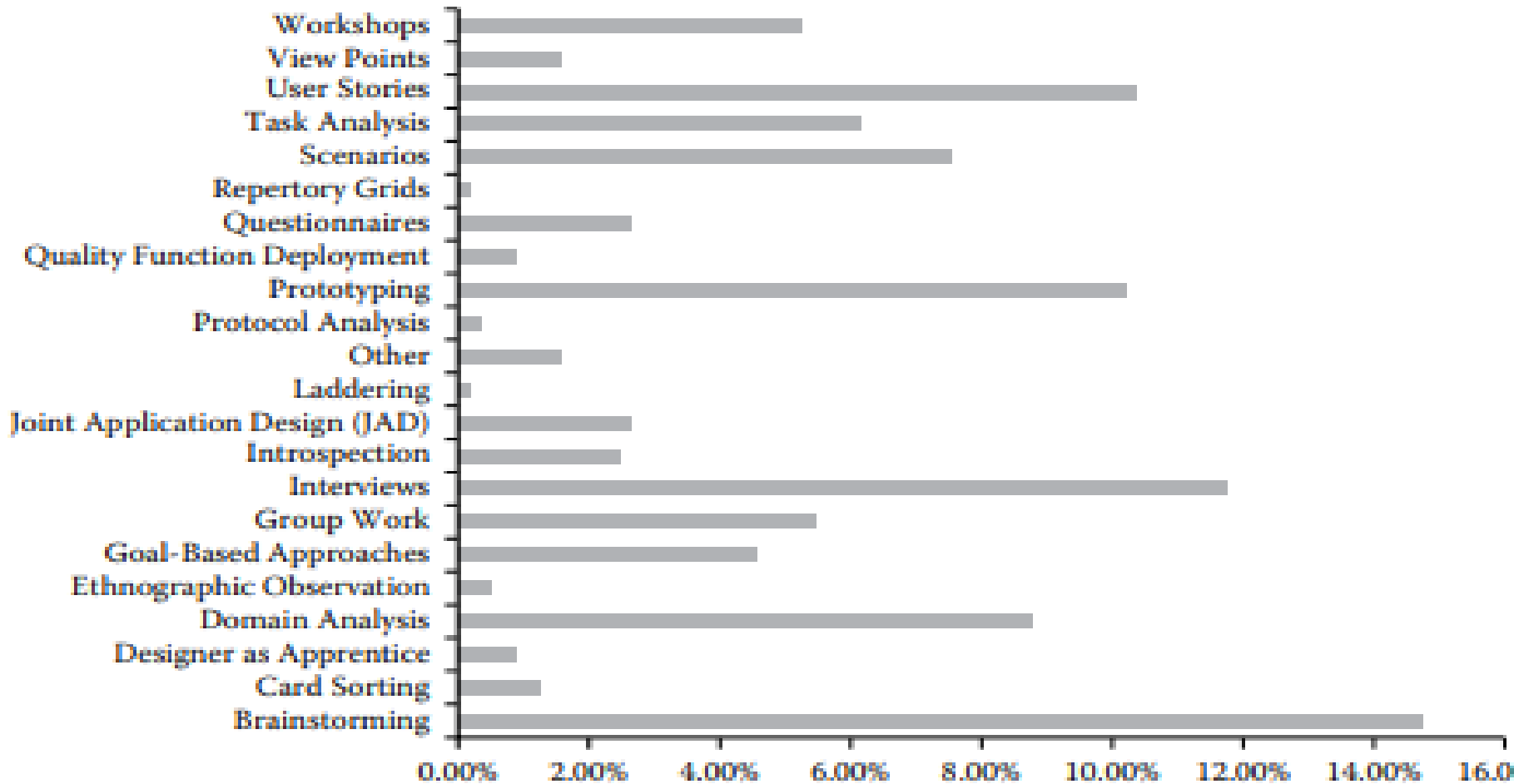
- ▶ Örneğin, görüşmeye dayalı teknikler, gereksinimlerin ortaya çıkarılmasının tüm yönleri için faydalıdır (ancak çok zaman alıcıdır). Öte yandan, prototipleme teknikleri, paydaşları analiz etmek ve gereksinimleri ortaya çıkarmak için en iyi şekilde kullanılır.
- ▶ Etnografik teknikler, problem alanını anlamak, paydaşları analiz etmek, gereksinimleri talep etmek vb. için iyidir.
- ▶ Son olarak, bu ortaya çıkarma teknikleri (kümeler) arasında bazılarının aynı şeyi başarması ve dolayısıyla birbirinin alternatifi olması bakımından açık bir örtüşme vardır.

A(Alternative) C(Complementary-Tamamlayıcı)

	röportajlar	Domain	Grup çalışması	etnografya	Prototipleme	Hedefler	senaryolar	bakış açıları
röportajlar		C	A	A	A	C	C	C
Domain	C		C	A	A	A	A	A
Grup Çalışması	A	C		A	C	C	C	C
Etnografya	A	A	A		C	C	A	A
Prototipleme	A	A	C	C		C	C	C
Hedefler	C	A	C	C	C		C	C
Senaryolar	C	A	C	A	C	C		A
Bakış Açıları	C	A	C	A	C	C	A	

- Uygun bir ortaya ıkarma teknikleri seti semenizde size rehberlik etmesi iin Tablo 3.2 ila 3.4'ü kullanabilirsiniz. Kullanılacak bir dizi teknik seerken, bir dizi tamamlayıcı teknik seersiniz. Örneğın, bir kombinasyonun bakış açısı analizi ve bir tür prototipleme istenebilir. Tersine, hem bakış açısı analizini hem de senaryo oluşturmayı kullanmak, muhtemelen aşırı derecede gereksiz bilgi verecektir.

- ▶ **Gereksinimlerin Ortaya Çıkması Tekniklerinin Yaygınlığı**
- ▶ “Hangi gereksinimleri ortaya çıkarma tekniğini/tekniklerini kullanıyorsunuz?” sorusuna verilen cevapların özeti Şekil 3.9.da gösterilmiştir.



► Tehlikeleri Ortaya Çıkarmak

- "Ciddi" ve "felaket" terimleri öznelidir, ancak genellikle can kaybını, ciddi yaralanmayı, büyük altyapı hasarını, ya da büyük mali kaybı içerir.
- Örneğin, evcil hayvan mağazası POS'u için bazı "yapılmaz" gereksinimleri şunları içerir:
 - ■ Sistem, müşteri bilgilerini harici sistemlere ifşa etmeyecektir.
 - ■ Sistem yetkisiz erişime izin vermeyecektir.
 - ■ Sistem, müşterilerin mağaza kredisini aşmasına izin vermeyecektir.

- ▶ Bu örneklerde, ilk iki gereksinim tehlike olarak kabul edilebilir, çünkü şirkete mali zarar verme potansiyeli üçüncüsünden çok daha fazladır.
- ▶ Tehlikeler, doğal olarak meydana gelen girdi anormalliklerinin (donanım arızaları gibi) veya yapay olarak meydana gelen (davetsiz girişlerden gelen saldırılar gibi) bir fonksiyonudur.
- ▶ Bu anormal giriş olaylarının tanımlanması gerekir ve bunların ortaya çıkan arıza modları ve kritikliklerinin süreç boyunca belirlenmesi gerekir.
- ▶ uygun bir “yapmamalı” seti geliştirmek için gereksinimlerin ortaya çıkarılması aşamasında diğer gerekliliklerde olduğu gibi, “yapmayacak” gerekliliklerine öncelik verilmesi gerekir.
- ▶ Tehlike belirlemeye yönelik tipik teknikler arasında, kötüye kullanım durumlarının geleneksel olarak geliştirilmesi, antimodeling ve resmi yöntemler yer alır.

- ▶ Sistemin veya ilgili sistemlerin önceki sürümlerinden oluşturulmuş istenmeyen davranışların kontrol listeleri de istenmeyen davranışların belirlenmesinde yardımcı olur. Geçerli standartlar ve düzenlemeler ayrıca belirli "değil" gereksinimleri içerebilir, örneğin Amerika Birleşik Devletleri'nde Sağlık Sigortası.

▶ **Kötüye Kullanım Durumları**

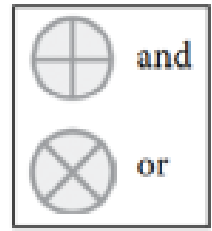
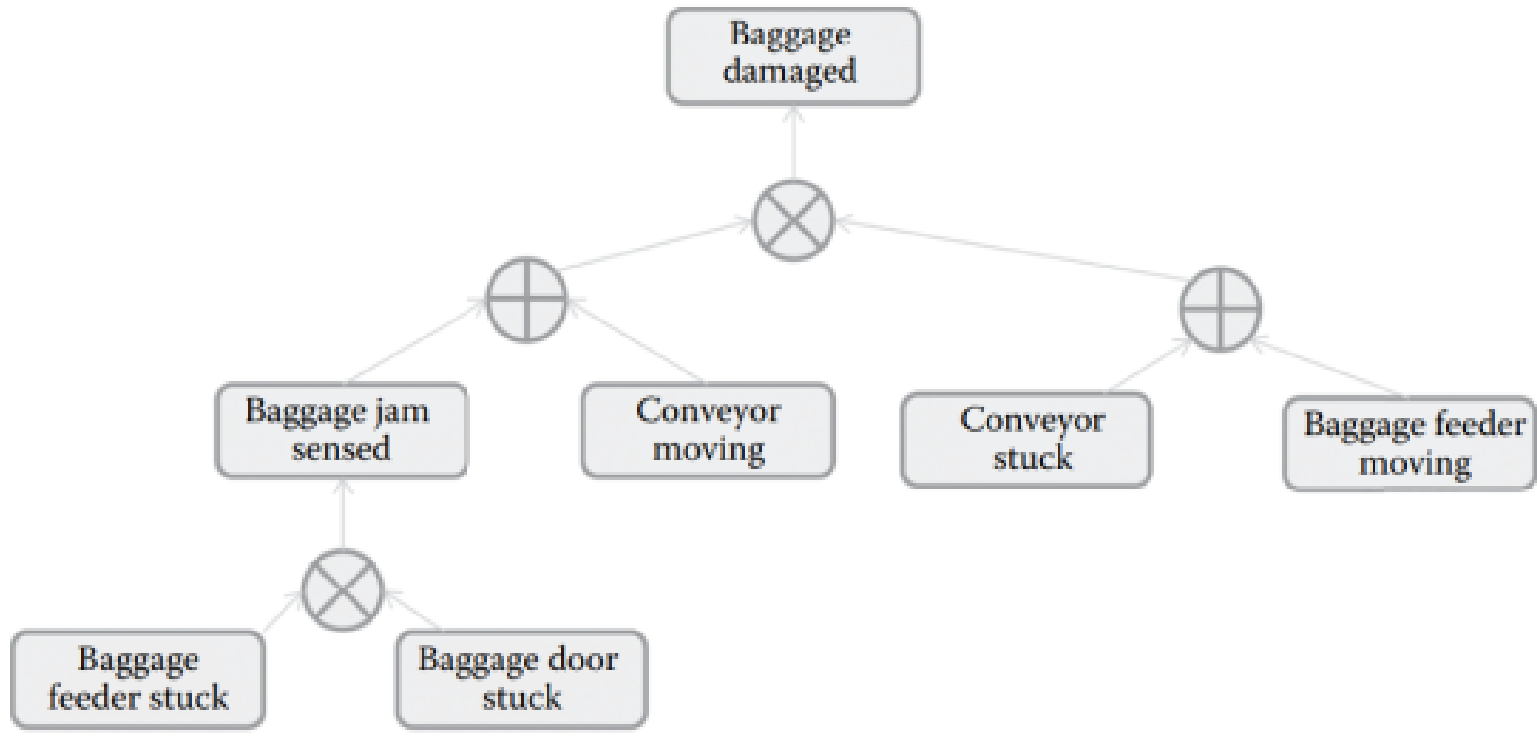
- ▶ Kullanım senaryoları, istenen davranışın yapılandırılmış, kısa açıklamalarıdır.
- ▶ İstenen davranışı tanımlayan vakalar, yanlış kullanım vakaları (veya kötüye kullanım vakaları) vardır.

- ▶ Çoğu sistem için tipik istenmeyen davranışlar suistimaller, güvenlik ihlalleri ve diğer kötü niyetli davranışların yanı sıra eğitimsiz, yönünü şaşırılmış veya beceriksiz kişilerce kötüye kullanım içerir.
- ▶ Kullanım senaryoları oluşturmanın kolay bir yolu, istenmeyen bir kişi rolünü üstlenmektir. Bu kişi sistemin istenmeyen bir kullanıcısıdır ve daha sonra böyle bir kişinin davranışlarını model alır.
- ▶ İstenmeyen kişiler, bilgisayar korsanlarını, davetsiz misafirleri, casusları, ve hatta iyi niyetli, ancak beceriksiz kullanıcıları temsil eder.

- ▶ Bu kişilerin belirlenmesinin tehlike gereksinimleri oluşturmaya nasıl yardımcı olduğunu görmek için, aşağıdaki örnekleri göz önünde bulundurun.
- ▶ Pet shop POS sisteminde, bir hacker'ın bu sisteme nasıl sızacağını düşünmek ve daha sonra hacker'ın niyetini bozacak gereksinimler yaratmak uygun olacaktır.
- ▶ Bagaj taşıma sisteminde, bir gereksinim mühendisi, beceriksiz veya dalgın bir gezgin rolünü üstlenebilir ve ardından bu tür kişilerin güvenliğini sağlayacak gereksinimleri belirleyebilir.
- ▶ Kötüye kullanım vakaları yaratma ihtiyacı, tüm olumsuz paydaşları tamamen tanımlamak için bir nedendir çünkü bu kişilerin çok sayıda olmayan kişiyi içermesi muhtemeldir.

► Antimodeller

- İstenmeyen davranışı elde etmenin başka bir yolu, sistem için antimodeller yaratmaktır.
- Antimodeller, hata ağaçlarıyla ilgilidir, yani model, bir sistem hatasına yol açan istenmeyen davranışlar için neden ve sonuç hiyerarşisidir. Sistem arızasının nedenleri “yapmamalı” gerekliliklerini oluşturmak için kullanılır.
- Örneğin, bagaj taşıma sisteminin güvenlik işlevini düşünün
- Şekil 3.10'da gösterilen hasarlı bagajın istenmeyen sonucudur.



- ▶ Şekil bizi aşağıdaki ham gereksinimleri yazmaya yönlendiriyor:
- ▶ ■ Bagaj sıkışması algılanırsa konveyör hareket etmemelidir.
- ▶ ■ Bagaj besleyici sıkışmışsa, konveyör hareket etmemelidir.
- ▶ ■ Bagaj kapısı sıkışmışsa konveyör hareket etmemelidir.
- ▶ ■ Konveyör sıkışmışsa, bagaj besleyici hareket etmemelidir.
- ▶ Bu gereksinimlerin daha fazla analize ve muhtemelen basitleştirilmesine ihtiyacı vardır, ancak antimodel, bu ham gereksinimleri sistematik bir şekilde elde etmemize yardımcı oldu.