

Bölüm 3

Gereksinimleri Ortaya Çıkarma

Giriş

Bu bölümde, gereksinimlerin bazı yollarla bulunabileceğini inceledik; keşfedilerek, ele geçirilerek veya zorlanarak. Bu bağlamda, tüm bu yollar ortaya çıkarmayla eş anlamlıdır. Ama “toplanma” eşdeğer değil. Gereksinimler örtüye alınıp yerleştirilmiş çürük meyveler değildir. Gereksinimler genellikle (en azından tamamı) bulunması kolay şeyler değildir. Zor algılananların bazıları ve karmaşık olanların birkaçı (eğer inatçı işlemler değilse) bütün titizliğiyle açıklığa kavuşturulmuş olmalıdır.

Gereksinimlerin ortaya çıkarılmasını yönetmeyi seçebileceğin bazı teknikler vardır ve muhtemelen birden fazlasını kullanmanız gerekir, farklı teknikleri farklı sınıftaki kullanıcı/paydaşlar kullanmak gibi. Bu teknikleri şöyle ele alacağız:

- Beyin Fırtınası
- Kart Sıralama
- Çırac olarak Tasarımcı
- Alan Analizi
- Etnografik Gözlem
- Hedefe Dayalı Yaklaşımlar
- Takım Çalışması
- Röportajlar
- İç Gözlem
- Ortak Geliştirme Uygulaması
- Merdivenleme
- Protokol Analizi
- Prototip Çalışması
- Kalite Fonksiyon Dağıtımı
- Soru Listesi
- Depo Kılavuzları
- Senaryolar
- Görev Analizi

- Kullanım Durumları
- Kullanıcı Hikâyeleri
- Bakış Açıları
- Atölyeler

Bu liste Zowghi ve Coulin (1998) 'in öne sürdüklerinden uyarlanmıştır.

Gereksinimlerin Ortaya Çıkması için Hazırlık

Gereksinimlerin ortaya çıkması için hazırlığın birinci adımı müşterilerin ve paydaşların tanımlanmasıdır. Ama paydaş gruplar (özellikle müşteriler) heterojen olabiliyorlar ve bu sebeple alt gruplarına farklı davranmak gerekiyor. Mesela evcil hayvan dükkânının POS sisteminin içerdiği farklı alt sınıflar şunlardır:

- Kasiyerler
- Müdürler
- Sistem Bakım Personelleri
- Mağaza Müşterileri
- Depo/Ambar Personelleri
- Muhasebeciler (Vergi bilgisi girişi için)
- Satış Departmanı (Ücretlendirme ve İndirim girişi için)

Her alt grubun kullanıcıları farklı eksikliklere sahiptir ve bunlar belirlenmelidir.

İşlem daha sonra ortaya çıkarılma için hazırlanırken:

- Müşteriler ve paydaşlar tanımlanır.
- Müşteriler ve diğer paydaş grupları; ilgilerine, kapsamlarına, yetkilerine ve diğer ayırıcı faktörlere uyularak sınıflara bölünür. (Bazı sınıflar birkaç seviyede bölümlere ayrılmaya ihtiyaç duyabilir.)
- Her kullanıcı sınıfı ve paydaş grupları için şampiyon veya temsilci grup seçilir.
- Uygun teknik seçimi için her kullanıcı sınıfı ve paydaş grubundan ilk girdiler istenir.

Kullanıcı sınıfını bölümlere ayırmanın başka bir örneği verilebilir. Bagaj taşıma sisteminin içerdiği farklı paydaşlar verilmiştir:

- Yolcular
- Sistem Bakım Personelleri

- Bagaj Görevlileri
- Havasahası Programcıları/Hareket Memurları
- Havaalanı Personeli
- Havaalanı Müdürleri ve Politikacılar

Fakat burada her birinin farklı ihtiyaçları bulunan farklı türden yolcular bulunmaktadır. Örnek olarak takip eden alt sınıfları dikkate alırsak:

- Çocuklar
- Yaşlı vatandaşlar
- İş insanları
- Gündelik seyahat edenler
- Askeri personel
- Siviller
- Sık sık yolculuk edenler

Her alt sınıfın farklı ortaya çıkarma teknikleriyle ele alınması gerekir. Örnek olarak anketler dikkate alınan gruplar askeri personelse daha az kullanışlıyken, çocuklar için uygun olmayabilir. Bazı alt sınıflar ise örtüşebilir. Örnek olarak; bir kişi iş seyahati yapan biri olabilir ve günlük seyahat eden biri de olabilir. Bu örtüşmeler ortaya çıkarma aktivitelerinde veri analizi yapılırken göz önünde bulundurulmalıdır.

Ortaya Çıkarma Teknikleri Anketi

Şimdi de ortaya çıkarma tekniklerini inceleme zamanı. Bu teknikleri alfabetik sıraya göre (herhangi bir üstünlük anlaşılmalı) sunduk. Bu bölümün sonunda bu tekniklerin farklı durumlarda tekrarlanma sıklığını ve uygunluğunu ele alacağız.

Beyin Fırtınası

Beyin fırtınası, sistemler için kapsayıcı amaçları oluşturmak amacıyla paydaşlar ve müşteriler ile resmi olmayan oturumlarda meydana gelir. Beyin fırtınası resmileştirilebilir belirlenmiş bir gündem, tutanak tutma ve resmi yapıların kullanımını içerir (Örneğin Robert's Rules of Order). Fakat Beyin fırtınası buluşmasındaki resmiyet muhtemelen buluşmada sergilenecek olan yaratma seviyesiyle ters orantılı olacaktır. Bu tür buluşmalar kendiliğinden bile resmi olmamalıdır, tek yapısı kayıtlı bazı ana kesitlerin bir bütün halinde toplanmasıyla olmalıdır.

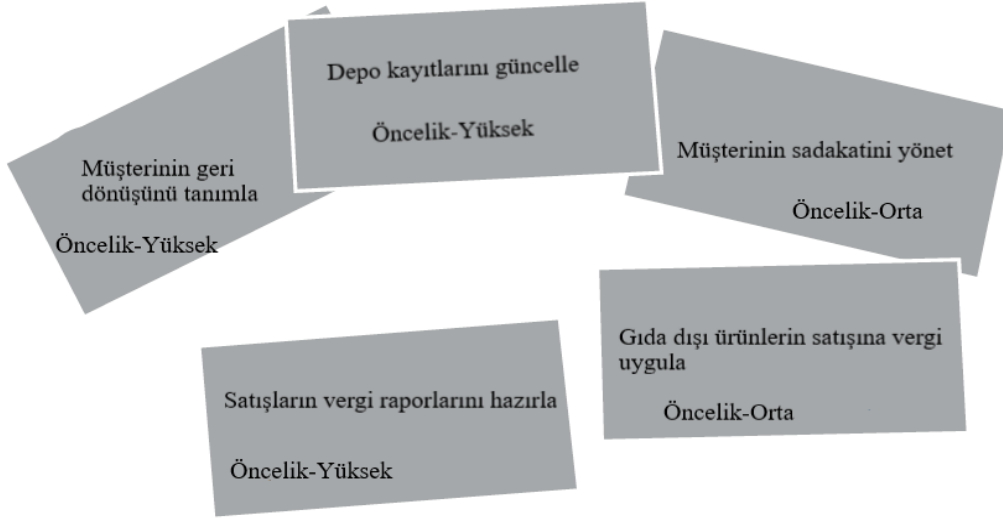
Beyin fırtınası oturumları süresince bazı öncelikli gereksinimler üretilmiş olmalıdır ancak bu işlem için ikincil durumdur. Ortak Geliştirme Uygulaması tekniği Beyin fırtınası (ve çok daha fazlasını) içerir ve muhtemelen diğer grup yönelimli ortaya çıkarma tekniklerinin çoğu örtük olarak bir tür beyin fırtınası içeriyor. Beyin fırtınası, misyon veya vizyon ifadesi oluşturma gibi genel amaç belirleme için de yararlıdır.

Kart Sıralama

Bu teknik, paydaşların sistem/yazılım ürünü için işlevsellik hakkında temel bilgileri içeren bir dizi kartı tamamlamasını kapsar. Ayrıca işlevselliklerin tasnif ve gerekçeler içermesiyle paydaşlar/müşteriler için iyi bir fikirdir.

Müşterilerin ve paydaşların kartları tamamlamasına izin veren süre önemli bir karardır. Kart Sıralama alıştırması birkaç saat içinde tamamlanabilirken paydaşları aceleye getirmek, büyük olasılıkla önemli, eksik işlevlere yol açacaktır. Paydaşlara fazla zaman vermek yine de işlemi gereksiz yere yavaşlatır. Tavsiye edilen en az 1 hafta (ve 2 haftadan fazla değil) kartların tamamlanması için izin verilir. Başka bir alternatif olarak müşterilere kartları tamamlamak için 2 saatlik seans verilir, 1 hafta sonra başka bir kart seansına tamamlamak ve gözden geçirmek için dönerler.

Her durumda her kart üretiminden sonra gereksinim mühendisi genellikle benzer yöntemle; işlevleri mantıksal olarak kümeleyerek kartları düzenler. Bu kümeler gereksinimlerin ayarlanmasının temel formudur. Sıralanan kartlar ayrıca CRC (yetenek, sorumluluk, işbirliği) kartları geliştirme sürecine girdi olarak nihai koddaki program sınıflarını belirlemek için kullanılabilir. Diğer teknik ise söz konusu olan QFD; Kart Sıralama aktivitesini içeriyor. İşlemi resimlendirecek olursak Şekil 3.1 kartların müşteriler tarafından üretilmiş evcil hayvan dükkânı POS sistemi için sınıflandırılmamış bir yığılda yatan küçük bir alt kümeyi gösteriyor.



Şekil 3.1 Evcil hayvan dükkânı POS sisteminin müşteriler tarafından üretilen sınıflandırılmamış kartlarının küçük bir kümesi.



Şekil 3.2 Evcil hayvan dükkânı POS sistemi için sınıflandırılmış kartlar.

Bu durumda her kartın içerdiği kısa tanımlara sadece fonksiyonellik ve öncelik derecesi dâhildir (özlük için gerekçeler gösterilmez).

Gereksinim mühendisi bu yığınla kartı analiz eder ve şu iki kartın “Müşteri yönetimi” fonksiyonlarına, iki kartın da “Vergi özellikleri” ve bir kartın da “Depo özellikleri” veya fonksiyonlarına ait olduğuna karar verir ve kartları Şekil 3.1 de gösterildiği gibi uygun yığınlara göre düzenler.

Düzeltilme ve eksik özellikler için bu sınıflandırılmış fonksiyonlar listesi müşteriye gösterilir. Daha sonra eğer gerekliyse yeni bir kart turu oluşturulabilir. İşlem, gereksinim mühendisi ve müşteri bu sistem özelliklerinden önemli miktarda ele geçirilmesine memnun olana kadar devam eder

Çırac olarak Tasarımcı

Çırac olarak Tasarımcı, gereksinimlerin belirlendiği bir gereksinim keşif tekniğidir. Gereksinim mühendisi, müşterinin ihtiyaçlarını anlamak için yaptığı çalışmalar hakkında yeterince öğrenmek için müşterinin “omzunun üzerinden bakar”. Tasarımcı ve müşteri arasındaki ilişki usta zanaatkâr ve çıkar arasındaki gibidir. Bu çırac ustadan bizim gereksinim mühendisinden (Tasarımcı) müşteriden müşterinin çalışmasını öğrenmek istediğimiz gibi yetenekler öğrenir. Çırac usta ne biliyorsa onu öğrenmek için oradadır (ve bu sebeple müşteriye çalışmanın bu parçaları hakkında konuştuğunda ve göstermede rehber olmak zorundadır). Tasarımcı özel ihtiyaçlara değinmek için buradadır.

Bu çalışma tekniği için müşterinin ihtiyacı olan bazı öğretme yeteneği türü görünebilirdi fakat bu doğru değil. Bazı müşteriler kendi verimli çalışmaları hakkında konuşamaz. Ancak ortaya çıktıkça konuşabilir. Dahası müşteriler onu sunmanın en iyi yolunu ya da güdülerini bulmak zorunda değiller, sadece ne yaptıklarını açıklayabilirler.

Çalışma ayrıca neyin önemli olduğunu da ortaya çıkarır. Örneğin insanlar yaptıkları her şeyden ve bazen neden yaptıklarından haberdar değil. Bazı hareketler yılların tecrübesi ve ifade etmek çok zor. Diğer hareketler ise sadece mantıklı gerekçeleri olmayan alışkanlıklardır. Çırağın varlığı ise ustaya(müşteriye) aktiviteler ve nereden geldiklerini düşünmek için sağladığı olanaktır.

İşi görmek ayrıntıları ortaya çıkarır çünkü bir görevi yerine getirmediğçe, tarif ederken detaylandırmak zordur. Son olarak, işi görmek yapıları ortaya çıkarır. Çalışmanın modelleri her zaman çalışan için açık değildir. Bir stajyer bir görevin birden fazla örneğini gözlemleyerek ve farklılıkları dâhil ederek, kendi başına nasıl yapılacağına dair bir anlayış oluşturarak işin stratejilerini ve tekniklerini öğrenir.

Bu teknikle çalışmak için sırayla, gereksinim mühendisi yapıları anlamalı ve çalışmaya dâhil etmeli. İçerdikleri:

- İşi tamamlayan stratejiler
- Yola çıkan kısıtlamalar
- Çalışmaya destek olan fiziksel çevrenin yapısı
- Farklı çalışma yolları
- Aktivitenin tekrarlanan modelleri
- Bunların herhangi bir potansiyel sistem üzerindeki etkileri

Tasarımcı, işi anladığını herhangi bir yanlış anlaşılmanın düzeltilebilir olduğunu müşteriye göstermelidir.

Son olarak Çırak olarak Tasarımcı yaklaşımını kullanmak gereksinimlerin buluşunda diğer proje faydalarından ötesini sağlar. Örneğin bu tekniği kullanmak işlemi geliştirmeye biçimlendirilmesiyle yardımcı olur.

Müşteri de tasarımcı da işlem devam ederken öğrenir. Müşteri neyin mümkün olduğunu öğrenir ve tasarımcı çalışma anlayışını genişletir. Eğer tasarımcı işlemi iyileştirmede bir fikir sahibiyse yine de müşteriden hemen (zamanında) geri bildirim almalıdır.

Alan Analizi

Gereksinim mühendisliğinde halihazırda belirtilmiş alan ilmine sahip olmak(gereksinim mühendisi ve/veya müşteri tarafından sahip olursa da olunmasa da). Alan analizi, Tasarlanan sisteme ilgili ve rakip uygulamalar tarafından “manzarayı” değerlendirmek için herhangi bir genel yaklaşımı içerir. Bu tür yaklaşım esas fonksiyonelliği ve sonra eksik fonksiyonelliği tanımlamada kullanışlı olabilir. Alan analizi ayrıca daha sonra tekrar kullanılabilir bileşende kullanıldı (son tasarım içine dâhil edilebilen açık kaynaklı yazılım öğeleri gibi). QFD ortaya çıkarma alan analizi yaklaşımını açık bir şekilde içerir ve bu tekniği kısaca tartışırız.

Etnografik Gözlem

Etnografik gözlem gereksinim mühendisine gözlemi direkt ve dolaylı bilgilendiren her tekniğe başvurur. Etnografik gözlem, sosyal bilimlerden ödünç alınan, bilim adamlarının bazı fenomênlerin çalışmasında insan faaliyeti ve işin gerçekleştiği çevre bilgisi vermek için

kullanılan bir tekniktir. En katı duyuda etnografik gözlem gözlemin uzun periyotlarını içerir (bunun sonucu olarak gereksinim ortaya çıkarma tekniklerinde itiraz da kullanılır).

Etnografik gözlemi örneklemek için, farklı bir kültürü inceleyen antropolog hayal edilir. Antropolog o kültürü incelerken içinde yaşar fakat minimal müdahaleci olarak. Yerken, uyurken, avlanırken, kutlama yaparken, yas tutarken ve daha fazlasını o kültürle yaparken, tüm türler dolaylı ve direkt olarak nasıl bu sosyal fonksiyonları ve inanç sistemlerini kazandıklarını kanıtlar. Gereksinimleri ortaya çıkarmak için etnografik gözlem uygulanırken gereksinim mühendisi kendini müşterinin kültürel çalışma alanına gömer. Burada ilaveten gözlem çalışması veya aktivitesi otomatikleştirilmiş, gereksinim mühendisi ayrıca, doğrudan iletişim kurulamayan ortamlarda müşteri ihtiyaçlarına ilişkin kanıtları toplayabilecek bir konumdadır. Çıracı olarak Tasarımcı bir gereksinim ortaya çıkarma tekniği olarak etnografik gözlem aktivitesini de içerir.

Bu tekniği pratikte anlatmak gerekirse, bu durum göz önünde bulundurulduğunda etnografik gözlemin içerdikleri:

- Gereksinimleri bir akıllı ev için bir müşteri için toplarsın.
- Müşterinin neye ihtiyacı olduğu hakkında çok zaman harcarsın.
- Müşteriler günlerini geçirirken onlarla etkileşim kurmak için zaman harcarsınız ve sorular sorarsınız. (Neden bulaşık makinesine sabah koşmuyorsun da gece koşuyorsun ?)
- İstekler ve arzular hakkında sözsüz ipuçları elde etmek için evde mevcut olan müşteriyi "iş başında" pasif bir şekilde gözlemlerken uzun zaman harcıyorsunuz.
- Evin kendisinden başka bilgiler kazanırsınız: kitaplıktaki kitaplar, duvardaki resimler, mobilya stilleri, hobilerin kanıtı, çeşitli cihazlarda aşınma ve yıpranma belirtileri vb.

Etnografik gözlem çok zaman alıcı olabilir ve gözlemcilerin önemli ölçüde eğitim almasını gerektirir. Sürecin müdahaleciliğine dayanan başka bir itiraz daha var. Fizikte, Heisenberg belirsizlik ilkesi olarak bilinen; iyi bilinen bir ilke vardır; bu sıradan kişilerin terimiyle, ölçtüğünüz şeyi etkilemeden bir şeyi tam olarak ölçemeyeceğiniz anlamına gelir. Bu nedenle, örneğin bir müşteri için çalışma ortamını gözlemlerken, herkes etkilemek için dışarı çıktığı için süreçler ve davranışlar değişir. Bu nedenle durumun yanlış görüntüsü oluşur ve hatalı kararlara yol açar.

Hedefe Dayalı Yaklaşımlar

Hedefe dayalı yaklaşımlar, gereksinimlerin, gereksinimlere yol açan bir dizi hedef aracılığıyla misyon bildiriminden kaynaklandığının kabul edildiği herhangi bir ortaya çıkarma tekniğini içerir. Yani, misyon beyanına bakıldığında, bu misyonu yerine getiren bir dizi hedef oluşturulur. 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 görev beyanını ele alalım: Bagaj taşıma işleminin yolcunun başlangıç noktasından varış noktasına kadar tüm yönlerini otomatikleştirmek.

Bu görevi yerine getirmek için aşağıdaki hedefler düşünülebilir:

Hedef 1: Check-in'den teslim alınmasına kadar bagaj takibini tamamen otomatikleştirmek.

Hedef 2: Bagajın check-in kontuarından uçağa yönlendirilmesini tamamen otomatikleştirmek.

Hedef 3: Kayıp bagaj miktarını %1'e düşürmek.

Bu hedefler daha sonra, hedef-soru-metrik (GQM) gibi yapılandırılmış bir yaklaşım kullanılarak gereksinimlere ayrıştırılabilir. GQM, gereksinim mühendisliği, mimari tasarım, sistem tasarımı ve proje yönetimi gibi birçok açıdan sistem mühendisliğinde kullanılan önemli bir tekniktir. GQM üç adımı içerir: sistemin amaçlarını veya hedeflerini belirt; her bir hedeften, hedefe ulaşıp ulaşılmadığını belirlemek için cevaplanması gereken soruları türet; soruları cevaplayabilmek için neyin ölçülmesi gerektiğine karar ver. (Basili ve Weiss 1984).

Örneğin, bagaj taşıma sistemi söz konusu olduğunda, hedef 3'ü göz önünde bulundurun. Burada ilgili soru "belirli bir (havaalanı/havayolu/uçuş/zaman aralığı/vs.) için bagajın yüzde kaçını kayboluyor?"

Bu soru şu formun bir gerekliliğini ortaya koymaktadır: Belirli bir [havaalanı/havayolu/uçuş/zaman aralığı/vb.] için kaybolan bagaj yüzdesi %1'den fazla olmayacaktır.

Bu gereklilik için ilgili ölçü, o zaman, belirli bir (havaalanı/havayolu/uçuş/zaman aralığı/vb.) için basitçe kaybolan bagaj yüzdesidir.

İlgili ölçü bu gereksinim için bir madde için (hava alanı/havayolu/uçuş/zaman dilimi/ vb.) basitçe bagaj kaybının oranıdır. Tabikide kayıp bagaj için gerçekten bir tanıma ihtiyacımız var çünkü sözde kayıp bagaj kaybolduğu ilan edildikten sonra genelde günler hatta aylar sonra bile ortaya çıkar. Ayrıca bir hava alanının bildirdiği kayıp bagajlar zamanla ya da bazı terminallerdeki belirli hava yolları ve benzerleri gibi kayıplar bakımından mantıklı tahminler çerçevelenmede yapılmalıdır. Biz burda kasten basit bir örnek alırsak- hedefler(gereksinimler) için uygun soru her zaman çok belirli olmaz ne de ilgili ölçü sorudan çok kolay türetilir. Örneğin aşağıdakilerin bir varyasyonu olan gereksinimleri görmek çok yaygındır.

Sistem kullanıcı dostu olmalı

Böyle bir gereksinimin problemi onun memnuniyetini göstermenin bir yolunun olmamasıdır- Kabul alım takımında olan herhangi biri sistemin kullanıcı dostu olmadığını ilan edebilir. Ama kullanıcı dostululuk bir sistem için mantıklı bir hedeftir. Bu yüzden, gelecek GQM, kullanıcı dostluluğun hedefleri ile ilgili olan birkaç dizi soru oluşturuyoruz, örneğin:

1. Sistemi kullanmayı öğrenmek ne kadar kolaydır?
2. Yeni bir kullanıcının ne kadar yardıma ihtiyacı vardır?
3. Kullanıcı ne kadar hata alır?

Sonraki bu sorular için bir ya da daha fazla ölçüyü tanımlarız. Hadi her biri için bir tane yapalım.

1. Bir kullanıcının belirli işlevleri nasıl gerçekleştireceğini öğrenmesi için harcadığı süre
2. Bir kullanıcının bir süre boyunca yardım özelliğini kaç kez kullanması gerektiği
3. Bir kullanıcının bir zaman boyunca bazı işlemler sırasında hata mesajı görme sayısı

Sonuç olarak biz bu ölçümler için kabul edilebilir aralıklar kurmak için müşteriyle çalışıyoruz. Sistem kurulduktan sonra gereksinimlerin karşılanması gerçek kullanıcılarla test denemeleri boyunca gösterilebilir. Test sırasında ve devamında kullanıcıların nitelikleri ve sayıları gibi bu testin değişkenleri sonra tanımlanabilir ve sistem planıyla birleştirilebilir. Böylece kullanıcı dostluluğun kabul edilebilir seviyesini tanımlayabiliriz.

Takım Çalışması

Takım çalışması gereksinimlerin keşfi, analizler ve sonraki süreçler boyunca kullanılır herhangi bir grup toplantısı için genel bir terimdir. Gereksinimlerin ortaya çıkarılması için grup odaklı çalışmaların en ünlüsü, birazdan tartışacağımız ortak uygulama tasarımıdır (JAD).

Grup aktiviteleri hissedarları ortaya getirmesi bakımından baya üretken olabilir ama anlaşmazlık ve bölücülük için potansiyel risk olabilir. Herhangi bir grup çalışmasında başarmanın anahtarı grup toplantılarının planlanması ve yapılmasıdır. Burada grup toplantıları hakkında hatırlanması gereken en önemli şeyler:

- Ev ödevinizi yapın- organizasyonun problemlerin politiklerin çevrenin ve dahasının tüm yönlerini araştırın.
- Toplantı gerçekleşmeden birkaç gün önce ajanda paylaşın (her bir maddenin zamanla hissedarlara dağıtılmasıyla)
- Ajandaya toplantı boyunca bağlı kalın (toplantının kapsamı sızmadan)
- Elinizde Adanmış not alan (yazılı) kişiye sahip olun
- Kişisel meselelerin içeri girmesine izin vermeyin
- Herkese sesinin duyulmasına izin verin
- En kısa zamanda fikir birliğini aramayın
- Ajandadaki tüm maddelerin yeterli şekilde alana kadar ayrılmayın
- Toplantının bitiminden birkaç gün sonrası içinde toplantının dakikalarını paylaşın ve katılımcılara değişiklik önermesine izin verin.

Bu ilkeler, gereksinimlerin ortaya çıkarılmasına yönelik JAD yaklaşımı için devreye girecektir.

Grup çalışmasının herhangi bir türünün birçok dezavantajı vardır. İlk olarak grup toplantılarını ayarlamak ve bu işlere odaklanan hissedarları toplamak zor olabilir. İnsanlar her zaman halka açık bir forumda gerçek duygularını ifade etmeye istekli olmadıkları için Açıklık ve samimiyet problemleri de oluşabilir. Herkesin farklı bir kişiliği olduğu için belirli bireyler toplantıyı domine edebilir (ve bu bireyler en “önemli” bireyler olmayabilir). Birkaç kişinin toplantıya sahip olmasına izin vermek, diğer katılımcıların çoğu için “dışlanma” duygularına sürükleyebilir.

Etkili bir toplantıyı yürütmek ve bu yüzden grup çalışmalarını kullanmak son derece gelişmiş liderlik, organizasyon ve kişilerarası becerileri gerektirir. Bu yüzden mühendislerin aradığı gereksinimler ne zaman mümkünse bu yetenekleri geliştirir.

Mülakatlar

Mülakatlar yoluyla çıkarım, iki bireysel hissedar veya küçük bir paydaş grubun (bazen odaklanma grubu denen) arasındaki yüz yüze iletişimi içerir. Mülakatlar, özellikle kullanılabilirlik gereksinimleri olmak üzere hissedarlardan sistem düzeyi gereksinimleri çıkarmak için kullanımı kolay bir tekniktir.

Eleme faaliyetlerinde kullanılacak üç tür mülakat vardır, ve bireylere ya da gruplara uygulanabilirler:

- Yapısız
- Yapılı
- Yarım yapılı

Muhtemelen en yaygın tür olan yapılandırılmamış mülakatlar, doğası gereği konuşma niteliğindedir ve katılımcıları rahatlatmaya hizmet eder. Spontene bir ‘itiraf’ gibi, bunlar, gereksinimler ne zaman olursa olsun, mühendis ve paydaş bir arada olduğu herhangi bir zamanda ve herhangi bir yerde ortaya çıkabilir. ve bilgileri bu şekilde yakalama fırsatı asla kaybedilmemelidir. Ancak görüşmecinin becerisine bağlı olarak, yapılandırılmamış görüşmeler başarılı veya başarısız olabilir. Bu nedenle, yapılandırılmış veya yarı yapılandırılmış mülakatlar tercih edilir.

Yapılandırılmış görüşmeler, doğası gereği çok daha resmidir ve titizlikle planlanmış önceden tanımlanmış soruları kullanırlar. Yapılandırılmış stili kullanarak röportaj yaparken şablonlar çok kullanışlıdır. Yapılandırılmış görüşmelerin en büyük dezavantajı, format çok kontrollü olduğu için bazı müşteriler bilgileri saklayabilir.

Yarı yapılandırılmış mülakatlar, yapılandırılmış ve yapılandırılmamış mülakatların en iyilerini birleştirir. Yani, gereksinim mühendisi dikkatlice düşünülmüş bir soru listesi hazırlar, ancak daha sonra mülakat sırasında kendiliğinden yapılandırılmamış soruların içeri girmesine izin verir. Yapılandırılmış mülakatlar tercih edilirken, hangisinin kullanılacağına seçimi çok fırsatçı bir karardır. Örneğin, müşterinin kurumsal kültürü çok resmi olmayan ve rahat ve güveni yüksek olduğunda, yapılandırılmamış görüşmeler tercih edilebilir. Daha yavaş hareket

eden, süreç odaklı bir organizasyonda, yapılandırılmış ve yarı yapılandırılmış mülakatlar muhtemelen daha fazla istenen şeydir.

Burda üç mülakat türünden herhangi birinde kullanılabilecek bazı örnek mülakat soruları

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

Hangi mülakat tekniği kullanılırsa kullanılsın, tüm doğru soruların sorulduğundan emin olmak için özen gösterilmelidir. Yani, önemli soruları geçmeyin ve konu dışı olan, rahatsız edici veya gereksiz soruları dahil etmeyin. Kesinlikle gerekli olduğunda; telefon, video konferans veya e-posta yoluyla mülakatlar yapılabilir, ama bu iletişim modlarında yanıtların belli önemli nüanslı yönlerinin kaybolabileceğinin farkında olun.

İç Gözlem

Bir gereksinim mühendisi, müşterinin istediğini düşündüğü şeye dayalı olarak gereksinimler geliştirdiğinde, iç gözlem sürecini yürütüyor demektir. Özünde, gereksinim mühendisi kendini müşterinin yerine koyar ve düşünür ki “müşteri olsaydım sistemin bunu yapmasını isterdim...” Gereksinim mühendisinin alan bilgisi müşterininkinden çok daha fazla olduğunda, içe dönük bir yaklaşım yararlıdır. Bazen müşteri, mühendise aşağıdakine benzer sorular soracaktır: "Benim yerimde olsan sen ne isterdin?" İç gözlem, gereksinim mühendisinin etkileşimlerinin her yönünü bilgilendirecek olsa da, müşteriye ne istemesi gerektiğini söylememe hakkındaki uyarımızı unutmayın.

Ortak Uygulama Tasarımı

Ortak uygulama cihazı (JAD), sistem kullanıcıları, sistem sahipleri ve uzun bir süre boyunca belirli bir dizi soruna odaklanan analistler ile yüksek düzeyde yapılandırılmış grup toplantıları veya küçük geri çekilmeleri içerir. Bu toplantılar günde 4-8 saat arasında ve 1 günden birkaç haftaya kadar süren bir süre boyunca gerçekleşir. JAD, katılımcılar aynı yerde olmadığına çok bölgeli uygulama için bile uyarlanmıştır (Cleland-Huang ve Laurent 2014). Geleneksel olarak büyük hükümet sistemleri projeleriyle ilişkilendirilse de teknik, endüstriyel ortamlarda her boyuttaki sistemlerde kullanılabilir.

JAD ve JAD benzeri teknikler, problemler, hedefler ve gereksinimler üzerinde grup oy birliğini sağlamak için sistem planlama ve sistem analizi faaliyetlerinde yaygın olarak kullanılır. Spesifik olarak, gereksinim mühendisi, operasyon tanımı kavramı, sistem hedefi tanımı, gereksinimlerin ortaya çıkarılması, gereksinim analizi, gereksinim belgesi incelemesi ve daha fazlası için JAD oturumlarını kullanabilir. Bir JAD incelemesi veya denetim toplantısı planlaması üç adımdan oluşur:

1. Katılımcıların seçilmesi
2. Ajandanın 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
- yazar kimse (yani toplantı tutanakları ve not tutanlar)
- Mühendislik personeli

Sponsor, analistler ve yöneticiler bir lider seçer. Lider kurum içi veya danışman olabilir. Bir veya daha fazla yazan kimse (not alıcı) seçilir, normalde yazılım geliştirme ekibinden. Analist ve yöneticiler, kullanıcı topluluğundan bireyler seçmelidir. Bu kişiler kendi iş alanlarında bilgili ve belagatli olmalıdır.

Bir oturum planlamadan önce, analist ve sponsor projenin kapsamını belirlemeli ve her oturumun üst düzey gereksinimlerini ve beklentilerini belirlemelidir. Oturum lideri ayrıca sponsorun insanları, zamanı ve diğer kaynaklar için 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 kodda adım adım ilerlemek için bazı kurallar yer almaktadır. Oturum lideri, bu maddelerin uygulanmasını sağlamak için her türlü çabayı göstermelidir.

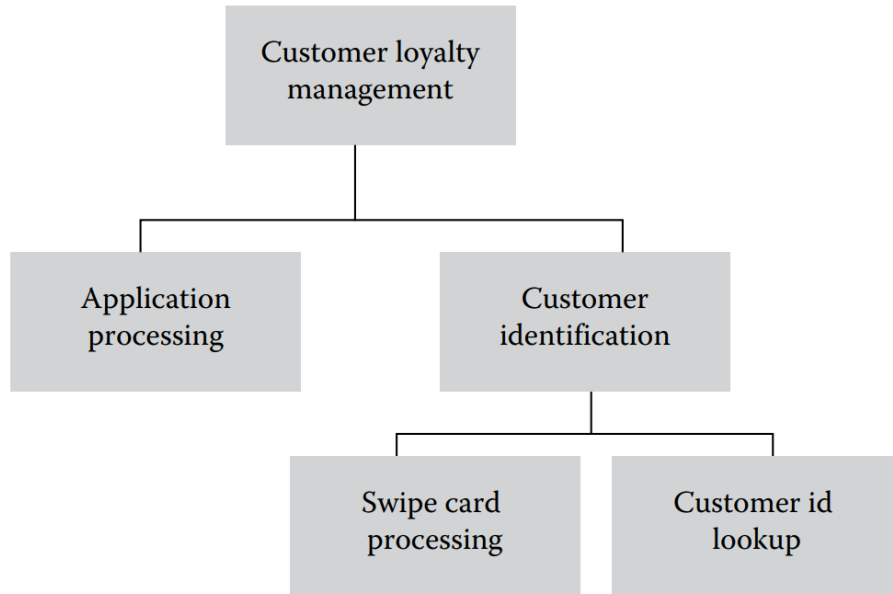
- Gündeme bağlı kalın.
- Programa bağlı kalın (gündem konularına belirli bir zaman ayrılır).
- Yazan kişinin not alabildiğinden emin olun.
- Teknik jargondan kaçının (inceleme teknik olmayan personeli içeriyorsa).
- Çatışmaları çözün (onları ertelememeye çalışın).
- Grupta fikir birliğini sağlamaya çalışın.
- Kişilerin oturumu domine etmesine izin vermeden kullanıcı ve yönetim katılımını teşvik edin.
- Toplantıyı kişisellikten uzak 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.

Merdivenleme

Merdivenlemede, gereksinim mühendisi, gereksinimleri ortaya çıkarmak için müşteriye kısa sorular sorar. Daha sonra yüzeyden 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.



Şekil 3.3 Evcil hayvan mağazası POS sistemi için merdiven şeması

RE: Bir müşteriye nasıl tanımlarsınız?

Müşteri: Sadakat kartlarını okutabilirler.

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 başvurularını nasıl tamamlar? ...

Vs.

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

Merdivenleme tekniği, bilginin hiyerarşik bir tarzda 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. 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 sürecin optimizasyonuna ve diğer yeniliklere yol açmıştır.

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.

Prototip Çalışması

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

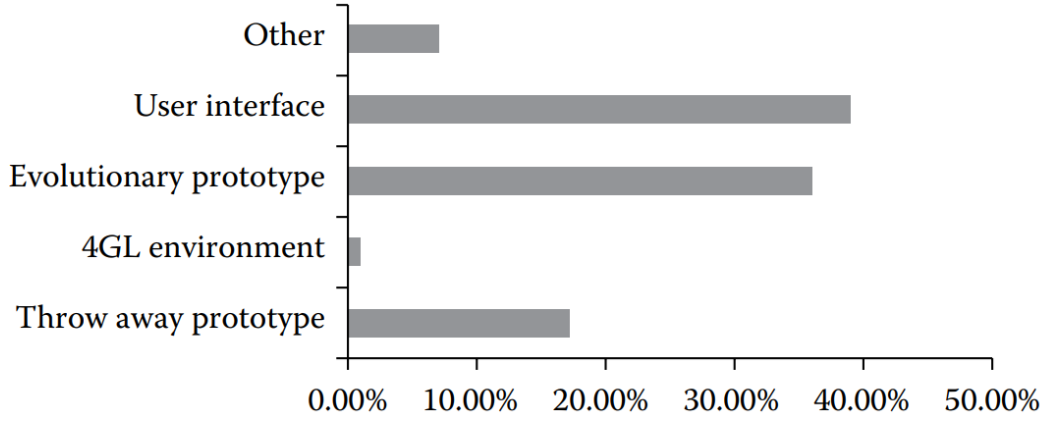
Prototipler, çalışan modelleri ve çalışmayan modelleri içerebilir. Çalışan modeller, yazılım sistemleri ve simülasyonlar söz konusu olduğunda yürütülebilir kodlar veya yazılım dışı sistemler için geçici veya ölçekli prototipler içerebilir. Çalışmayan modeller, storyboard'ları ve kullanıcı ara yüzlerinin 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 tasarlanabilir (kullanılarak atılmaz). Örneğin, grafiksel kullanıcı ara yüzü 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 yazıcı, belirli sistemlerin fiziksel modellerini oluşturmada önemli bir araç haline geldi. 3-D yazıcının diğer hızlı prototipleme teknolojilerine göre iki önemli avantajı vardır. Birincisi maliyettir—endüstriyel kalitede 3 boyutlu yazıcılar birkaç bin dolara satın alınabilirken, geleneksel bilgisayarlı sayısal kontrol (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 (prototipin nihai sisteme dönüştüğü yer) veya kullanıcı ara yüzü prototiplemesi. Bazı kuruluşlar birden fazla prototipleme türü kullanabilir. Kasab et al. (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'de gösterilmiştir.

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



Şekil 3.4 Yazılım geliştirme yaşam döngüsü metodolojisi boyunca prototip yöntemleri seçimi. (Kassab, M., Neill, C & Laplante, P. State of pratiğin gereksinim mühendisliği: güncel verilerden uyarlanmıştır. Innovations in Systems and Software Engineering, 10, no. 4 (2014): 235–241. İzinle birlikte) .)

İlk olarak, bazı durumlarda, saklanması 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 problem, prototiplemenin bazı işlevsel olmayan gereksinimleri keşfetmede her zaman etkili olmamasıdır. Bu, özellikle yalnızca geçerli standartların ve yasaların analiziyle elde edilebilecek gereksinimler için geçerlidir (Kassab ve Ormandjieva 2014).

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

Kalite Fonksiyonu Dağıtımı

Kalite fonksiyon dağıtımı (QFD), müşteri gereksinimlerini keşfetmeye ve üretim aşaması boyunca kullanılacak ana kalite güvence noktalarını tanımlamaya yönelik bir tekniktir. QFD, müşterilerin ihtiyaç ve isteklerinin dikkatlice duyulmasını ve ardından analizden uygulamaya ve dağıtıma kadar doğrudan bir şirketin dahili teknik gereksinimlerine dönüştürülmesini sağlamak için bir yapı sağlar. 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 “kaliteli ev” olarak anılır. (Şekil 3.5; Akao 1990).

QFD, 1966 yılında Yoji Akao tarafından imalat, ağır sanayi ve sistem mühendisliğinde kullanılmak üzere tanıtıldı. IBM, DEC, HP, AT&T, Texas Instruments ve diğerleri tarafından yazılım sistemlerine de uygulanmıştır.

“Müşterinin Sesi”ne atıfta bulunduğumuzda, müşterinin kendi sözleriyle ifade ettiği gibi, gereksinim mühendisinin üründen neye ihtiyaç duyduklarını anlamak için müşterileri sempatik bir şekilde dinlemesi gerektiğini kastediyoruz. Müşterinin sesi, ürünlerin yalnızca “mühendis sesinden” geliştirilmemesini sağlamak için tüm analiz, tasarım ve geliştirme faaliyetlerinin temelini oluşturur. Bu yaklaşım, gereksinimlerin ortaya çıkarılmasının özünü somutlaştırır.

Aşağıdakiler 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 çalışmaları 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 içerir.

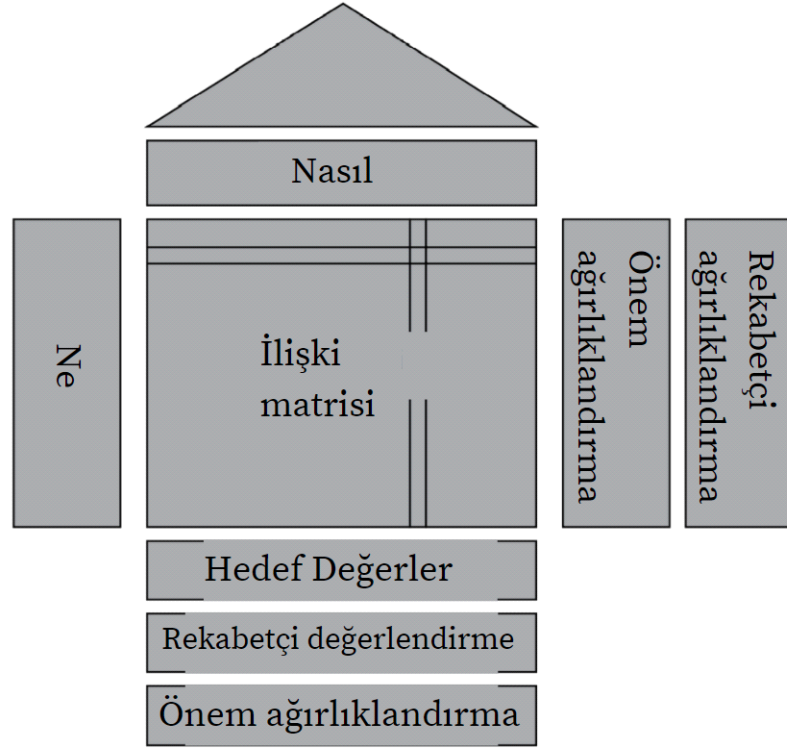


Figure 3.5 QFD'nin "kalite evi" (Aka0 1990).

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, temel özelliklerin yeni sistemden atlanmamasını sağlamaya yardımcı olur ve istenen eksiksizlik 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

bulduğunu belirtmek, yeni pet shop POS sisteminde böyle bir özelliğin zorunlu olduğunu gösterebilir.

Gereksinim mühendisliğine toplam yaşam döngüsü yaklaşımını içerdiğinden, 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 - 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		
	Benim Favori Hayvanım	En İyi Arkadaşlar	<i>Fido-2.0</i>
Desteklenen maksimum eşzamanlı kullanıcı	100	250	Sınırsız
Desteklenen kablosuz cihazlar	Evet	Evet	Evet
İş analitiği özellikleri	Evet	Evet	Hayır
Desteklenen işletim sistemi	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 yakın bir yaklaşım kullanılmalıdır. QFD rekabetçi analizinin yapılandırılmış doğası, daha eksiksiz bir gereksinim setine yol açan hiçbir önemli gereksinimin eksik olmadığından emin olmanın etkili bir yoludur.

Anketler/İncelemeler

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, çoktan seçmeli, doğru-yanlış veya açık uçlu olabilir ve 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?

(a) 0–1000; (b) 1001–10,000; (c) 10.001–100.000; (d) >100.000

- Kaç farklı depo siteniz var? _____
- Kaç farklı mağaza konumunuz var? _____
- Şu anda kaç benzersiz müşteriniz var? _____

Sorular, kapalı uçlu sorular için bile yeterince çerçevelenmemişse, gereğinden fazla ve altta kalma tehlikesi vardır. Bu nedenle, anket ortaya çıkarma teknikleri, etki alanı hem paydaşlar hem de gereksinim mühendisi tarafından çok iyi anlaşıldığında en kullanışlıdır.

Büyük ölçekli anketlere girişmeden önce, amaçlanan anket popülasyonunun küçük bir alt kümesiyle bir pilot çalışma yürütmek önemlidir. Sonuçlar analiz edilir ve kafa karıştırıcı, eksik veya konu dışı soruları belirlemek amacıyla anket katılımcıları ile görüşülür. Daha sonra, anket daha büyük bir nüfusa uygulanmadan önce araç rafine edilebilir.

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

Gerçekleştirilmesi gerekmeyen bir dizi seçenek verildiğinde, bir kişi gerçekten karar verecekse çok fazla sayıda seçeneği arzulama eğiliminde olacaktır.

Aşağıdaki örnekten dolayı bu etkiye “dondurma deposu etkisi” diyoruz. El yapımı bir dondurma dükkânı açmaya karar veren bir girişimci düşünün. 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, ankete katılanların %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 dükkânını 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. Müşterilerin ankette bu lezzetleri satın alacaklarını söylemelerine rağmen, seçim yapma zamanı geldiğinde farklı davrandıklarını fark ediyor. Bu nedenle, müşterilere özellik setleri hakkında seçenekler sunarken dondurma dükkânının etkisini unutmayın; onlar bir şey söyler ve başka bir şey yapar.

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

Repertuar İzgaraları

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 belirlenmesi için yararlıdır.

İzgaralar, satırların sistem varlıklarını ve istenen nitelikleri temsil ettiği ve sütunların her bir paydaşa dayalı sıralamaları temsil ettiği bir özellik veya kalite matrisi gibi görünür. Şebekeler hem nitelikleri hem de özellikleri içerebilirken, genellikle şebekelerin analiz tutarlılığı ve uyumsuzluk çözümü 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, 4'te biraz daha düşük olarak derecelendirildi). Ancak Havayolu İşçileri Sendikası temsilcisi için güvenlik en önemli şeydir (sonuçta sendika üyeliğinin sistemle günlük olarak etkileşime girmesi gerekir)

Bagaj taşıma hızı	1	1	5
Hata toleransı	4	5	5
Emniyet	5	4	4
Güvenilirlik	3	5	5
Bakım kolaylığı	3	5	5

Havaalanı operasyon müdürü
Bakım mühendisi
Havayolu işçileri sendika temsilcisi

Figure 3.6 Bagaj taşıma sistemi için kısmi repertuar ızgarası.

Özünde, bu derecelendirmeler paydaşların gündemlerini veya farklı bakış açılarını yansıtır. Bu nedenle, repertuar ızgaralarının kullanılmasının, paydaş hedeflerini içeren anlaşmazlıklarla erken yüzleşmede neden çok yardımcı olabileceğini görmek kolaydır. Buna ek olarak, ızgaralar paydaşların tutumlarını yakaladıkları için sistemin geliştirilmesinde daha sonra anlaşmazlıklarla başa çıkmak için değerli belgeler sağlayabilir. Nitelikler ve özellikler hakkında göz ardı edilmesi zor bir şekilde. Yine de repertuar ızgaralarını kullanırken,

dondurma dükkânının etkisini hatırlayın; paydaşlar halka açık bir ortamda bir şey söyleyecek ve daha sonra farklı davranacaktır.

Senaryolar

Senaryolar, sistem işleyişinin, kullanıcı sınıflarının ve istisnai durumların üst düzey bir tanımını sağlayan, kullanımda olan sistemin resmi olmayan açıklamalarıdır.

İşte pet shop POS sistemi için örnek bir senaryo.

Bir müşteri evcil hayvan mağazasına girer ve arabayı çeşitli ürünlerle doldurur. Kasiyer müşterinin sadakat kartı olup olmadığını sorar. Eğer müşteri sadakat kartına sahipse, kasiyer kartı okutarak müşterinin kimliğini doğrular. Değilse, kasiyer bir tanesini yerinde tamamlamayı teklif eder.

Sadakat kartı faaliyetinden sonra, kasiyer bir barkod okuyucu kullanarak ürünleri tarar. Her kalem 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 kullanarak ödeme yapar. Tüm uygun toplamalar (satışlar, vergiler, indirimler vb.) hesaplanır ve kaydedilir.

Alan yeni olduğunda senaryolar oldukça kullanışlıdır (örneğin uzay istasyonu için bir senaryo düşünün). Kullanıcı hikâyeleri aslında bir tür senaryodur.

Görev Analizi

Hâlihazırda incelemiş olduğumuz hiyerarşik yönelimli tekniklerin çoğu gibi, görev analizi de sistem tarafından gerçekleştirilecek görevlerin işlevsel bir ayrıştırmasını içerir. Yani, en yüksek soyutlama düzeyinden başlayarak, tasarımcı ve müşteriler daha ileri düzeyde ayrıntı ortaya çıkarır. Bu ayrıntılı ayrıştırma, en düşük işlevsellik düzeyine (tek görev) ulaşılan kadar devam eder.

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

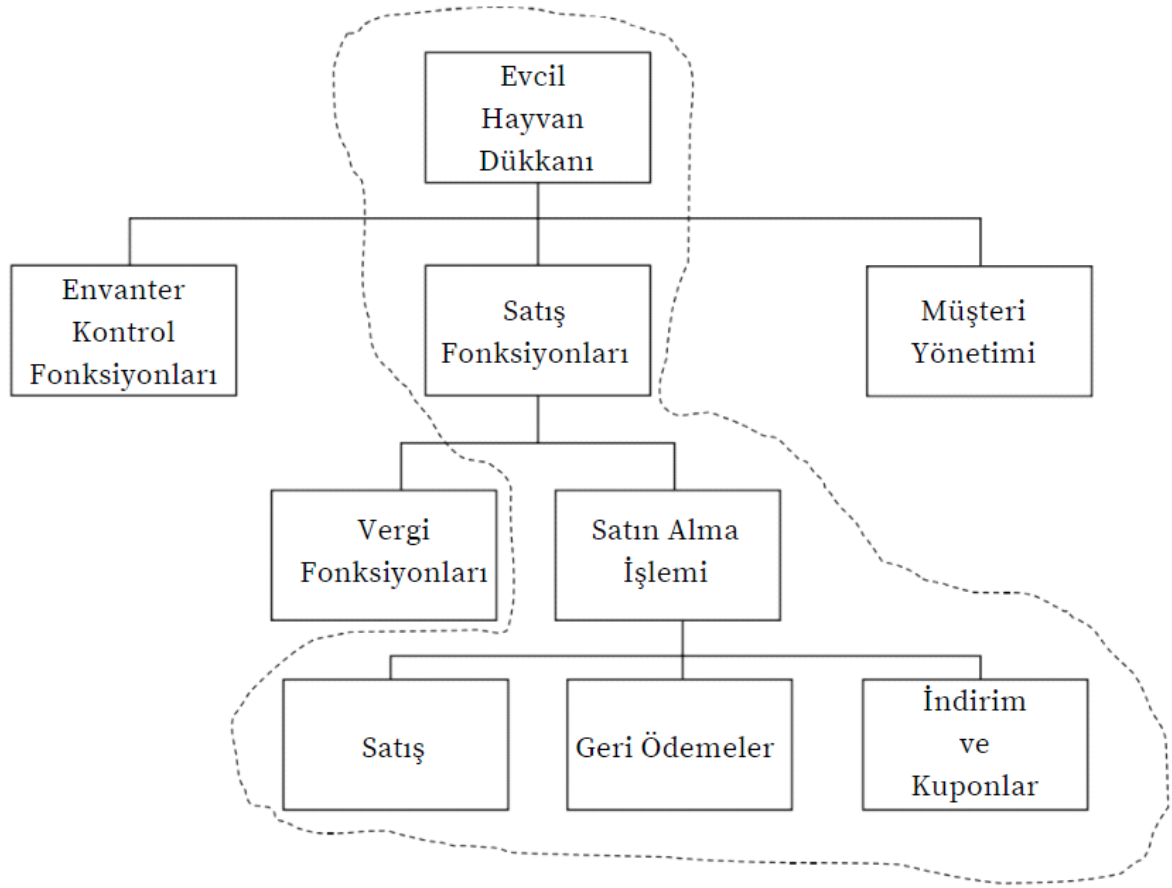


Figure 3.7 Pet shop POS sistemi için kısmi görev analizi

Burada, kapsamlı evcil hayvan mağazası POS sisteminin üç ana görevden oluştuğu kabul edilir: envanter kontrolü, satış ve müşteri yönetimi. Satış fonksiyonlarının altına indiğimizde bunların şu görevlerden oluştuğunu görüyoruz: vergi fonksiyonları ve satın alma işlemleri. Daha sonra satın alma işlemi fonksiyonuna geçerek bu görevleri satış, iade, indirim ve kupon görevlerine ayırıyoruz.

Görev analizi ve ayrıştırma, yeterli bir ayrıntı düzeyine ulaşılan (tipik olarak, bir yöntem veya ayrıştırılmaz prosedür düzeyine kadar) ve diyagram tamamlanana kadar devam eder.

Kullanım Senaryoları

Kullanım senaryoları, daha karmaşık müşterilerin ve paydaşların isteklerini tanımlamanın bir yoludur. Kullanım senaryoları, sistem ve sistem çevresindeki, özellikle insan kullanıcılar ve diğer sistemler arasındaki etkileşimleri tasvir eder. Saf yazılım veya hibrit yazılım sistemlerinin davranışını modellemek için kullanılabilirler.

Kullanım senaryoları, sistemin çalışma senaryolarını tasarımcının (müşterilerin aksine) bakış açısıyla açıklar. Kullanım senaryoları, tipik olarak, sistemin dış ortamıyla etkileşimlerini gösteren bir kullanım durumu diyagramı kullanılarak temsil edilir. Kullanım durumu diyagramında kutu sistemin kendisini temsil eder. Çubuk figürler, sistemle etkileşime giren harici 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 (kullanım durumları). Kesintisiz çizgiler, oyuncuları her kullanımla ilişkilendirir. Şekil 3.8, bagaj muayene sistemi için bir kullanım durumu şemasını göstermektedir.

Üç kullanım gösterilmiştir: bagajın bir görüntüsünün alınması ("bagajın görüntüsü"), bir güvenlik tehdidinin tespiti (bu durumda torba, çevrimdışı işleme için konveyörden reddedilir) ve ardından sistem mühendisi tarafından yapılandırma. Görüntüleme kamerasının, ürün sensörünün ve reddetme mekanizmasının insan benzeri bir çubuk figürle temsil edildiğine dikkat edin - bu tipik bir durumdur - çubuk figür, 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 bir kullanım durumu, dikkate alınan sistemin çalışma senaryolarının yanı sıra ön ve son koşullar ve istisnaları açıklayan bir belgeleme biçimidir. Yinelemeli bir geliştirme yaşam döngüsünde, analiz ve tasarım iş akışları ilerledikçe bu kullanım durumları giderek daha iyi hale getirilecek ve ayrıntılı hale gelecektir.

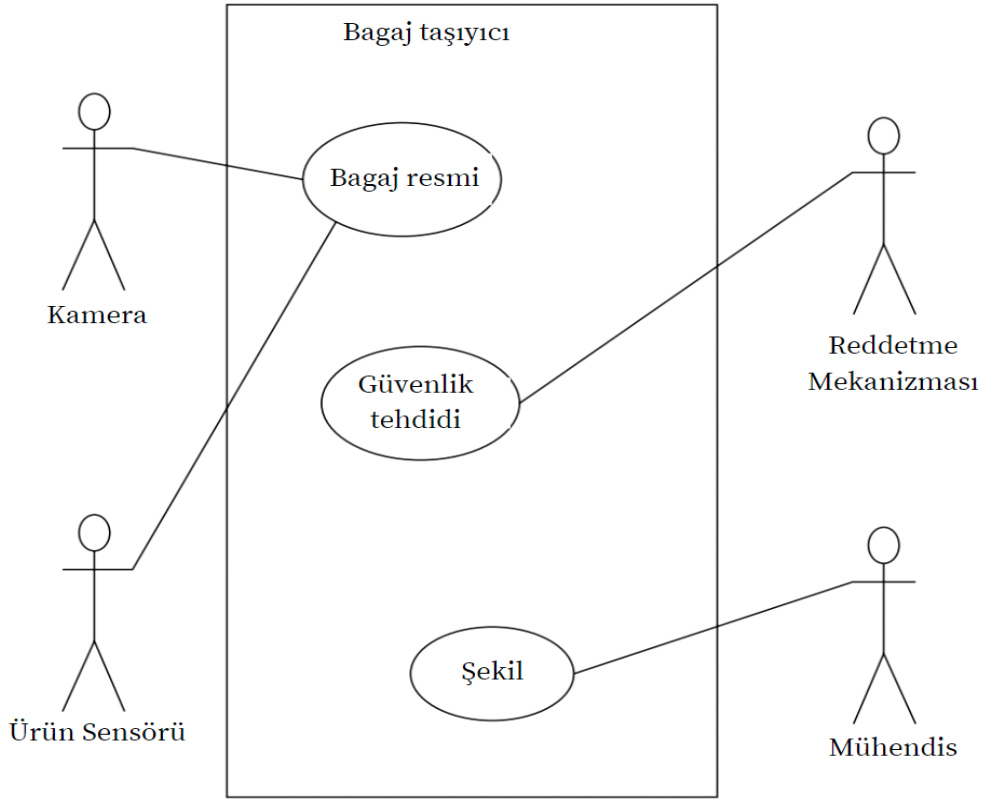


Figure 3.8 Bagaj kontrol sisteminin kullanım durum diyagramı.

Daha sonra, her bir kullanım durumu tarafından tanımlanan davranışları tanımlamak için etkileşim diyagramları oluşturulur. İlk yinelemede, bu diyagramlar sistemi bir "kara kutu" olarak tasvir eder, ancak alan modellemesi tamamlandıktan sonra, kara kutu daha sonra görüleceği gibi nesnelerin bir iş birliğine dönüştürülür.

İyi geliştirilmişse, bazen kullanım durumları bir kalıp dili oluşturmak için kullanılabilir ve bu kalıplar ve türetilen tasarım öğeleri ilgili sistemlerde yeniden kullanılabilir (Issa ve Al-Ali 2010). Gereksinimleri belirlerken kalıpları kullanmak, daha yüksek düzeyde tutarlılık sağlar ve belirli gereksinim özelliklerinin otomatik ölçümündeki hataları azaltabilir.

Gereksinim belirleme belgesinde gereksinimlerin keşfinde ve gereksinimlerin modellenmesinde çok önemli bir araç haline geldikleri için, birçok örnekle birlikte kullanım durumlarının kapsamlı bir tartışması Ek E'de bulunabilir.

Kullanıcı hikâyeleri

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

Kullanıcı hikâyeleri, sistemin kendileri için ne yapması gerektiğine göre müşteriler tarafından ve kendi “sesleri” ile yazılır. Kullanıcı hikâyeleri genellikle üçe beş inçlik bir kart üzerine yazılan iki ila dört cümleden oluşur. Yaklaşık 80 kullanıcı hikâyesi genellikle bir sistem artışı veya gelişimi için uygundur, ancak uygun sayı, uygulama boyutuna ve kapsamına ve kullanılacak geliştirme metodolojisine (örneğin, çevik veya artımlı) bağlı olarak büyük ölçüde değişecektir.

Pet shop POS sistemi için bir kullanıcı hikâyesi örneği aşağıdaki gibidir:

- Her müşteri bir kasada kolayca kontrol edebilmelidir.
- Self servis desteklenecektir.
- Tüm kuponlar, indirimler ve geri ödemeler bu şekilde ele alınmalıdır.

Kullanıcı hikâyeleri, yalnızca hikâyenin uygulanmasının ne kadar süreceğine dair makul derecede düşük riskli bir tahmin yapmak için yeterli ayrıntı sağlamalıdır. Uygulama zamanı geldiğinde, hikâye geliştiricileri detayları detaylandırmak için müşteriyle buluşacak.

Kullanıcı hikâyeleri ayrıca kabul testinin temelini oluşturur. Örneğin, kullanıcı hikâyesinin doğru şekilde uygulanıp uygulanmadığını doğrulamak için bir veya daha fazla otomatik kabul testi oluşturulabilir.

Kullanıcı hikâyeleri Bölüm 7'de ve Ek D'de daha ayrıntılı olarak tartışılmaktadır.

Bakış Açıları

Bakış açıları, farklı seçmenlerin (bakış açılarından) bilgileri organize etmenin bir yoludur. Örneğin, bagaj taşıma sisteminde, aşağıdaki paydaşların her biri için sistemin farklı bakış açıları vardır:

- Bagaj taşıma personeli
- Gezginciler
- Bakım Mühendisleri

- Havaalanı Yöneticileri
- Düzenleyici Kurumlar

Bu paydaşların her birinin ihtiyaçlarını ve çelişkilerini tanıyarak bu bakış açıları ile ortaya çıkan çatışmalar, çeşitli yaklaşımlar kullanılarak uzlaştırılabilir.

Gerçek bakış açıları, iş alanından, süreç modellerinden, işlevsel gereksinim özelliklerinden, organizasyon modellerinden vb. çeşitli bilgileri içerir.

Somerville ve Sawyer (1997), aşağıdaki bileşenlerin, her bir bakış açısında olması gerektiğini önerdi:

- Belirtimde kullanılan gösterimi tanımlayan bir temsil stili
- “Bakış açısının ele aldığı ilgi alanı” olarak tanımlanan bir alan
- Tanımlanmış tarzda ifade edilen bir sistemin modeli olan bir belirtim
- Belirtimin nasıl oluşturulacağını ve kontrol edileceğini tanımlayan bir süreç modeline sahip iş planı
- Belirtimi oluştururken, kontrol ederken ve değiştirirken gerçekleştirilen eylemlerin bir izi olan bir çalışma kaydı

Bakış açısı analizi genellikle önceliklendirme, anlaşma ve gereksinimlerin sıralanması için kullanılır.

Atölyeler

En genel düzeyde çalıştaylar, gereksinim sorunlarını çözmek için herhangi bir paydaş toplantısıdır. Çalıştayları resmi ve gayri resmi olmak üzere iki tür olarak ayırt edebiliriz.

Resmi çalıştaylar iyi planlanmış toplantılardır ve genellikle sözleşmeyle zorunlu kılınan “teslim edilebilir” etkinliklerdir. Örneğin, DOD-MIL-STD-2167, birden fazla gerekli ve isteğe bağlı atölyeyi (kritik incelemeler) içeriyordu. Resmi bir atölye tarzının iyi bir örneği JAD'de(Ortak Uygulama Geliştirme) somutlaştırılmıştır.

Gayri resmi çalıştaylar genellikle yüksek düzeyde yapılandırılmış toplantılardan daha az sıkıcıdır. Ancak gayri resmi toplantılar çok özensiz olma eğilimindedir ve yanlış güvenlik duygusuna ve bilgilerin kaybolmasına neden olabilir. Bir tür çalıştay gerekiyorsa, daha önce tartışılan başarılı toplantılar için parametreler kullanılarak resmi bir çalıştay yapılması tavsiye edilir.

İşlevsel Olmayan Gereksinimlerin Ortaya Çıkarılması

İşlevsel olmayan gereksinim (NFR) ortaya çıkarma teknikleri, işlevsel gereksinimler ortaya çıkarma tekniklerinden farklıdır. NFR'ler genellikle gereksinim analizi sırasında gayri resmi olarak belirtilir, genellikle çelişkilidir ve yazılım geliştirme sürecinde uygulanması ve doğrulanması zordur. Borg ve ark. (2003), farklı kuruluşlarda NFR ile ilgili sorunların köklerini belirlemeye yönelik görüşmeler gerçekleştirmiştir. Sonuç, NFR ile ilgili sorunların geliştirme sürecinin dört aşamasında ortaya çıkmasıydı: ortaya çıkarma, belgeleme, yönetim ve test; Ortaya çıkarma aşamasında NFR ihmalî tüm geliştirme süreci boyunca yayıldığından, ortaya çıkarma NFR ile ilgili olası sorunların ana kaynağı olarak işaretlenir. Bunun nedenleri, örneğin: G

- Gereksinimler aşamasında belirli kısıtlamaların bilinmemesi
- NFR'lerin birbiriyle çelişme eğiliminde olması
- FR'lerin ve NFR'lerin ayrılması, aralarındaki bağımlılıkların izlenmesini zorlaştırırken, fonksiyonel ve tüm gereksinimler birlikte karşılaştırılırsa işlevsel olmayan hususları ayırmak zordur.

Mevcut çeşitli yöntemler arasından NFR'leri ortaya çıkarmak, detaylandırmak ve belgelemek için bir yöntem seçmek kolay değildir. Aşağıdakiler, NFR ortaya çıkarma yöntemlerinin arzu edilen özellikleridir (Herrmann ve diğerleri, 2007):

1. Daha az deneyimli personel tarafından yöntem kullanımını kolaylaştırmak ve sonuçların tekrarlanabilirliğini desteklemek için rehberli bir süreç
2. Kalite güvencesini kolaylaştırmak için ölçülebilir NFR'lerin türetilmesi
3. Öğrenmeyi desteklemek ve yeniden çalışmayı önlemek için türetilmiş NFR'lerin eksiksizliğini desteklemek için eserlerin yeniden kullanılması (!)
4. Gizli gereksinimleri de yakalamak ve böylece eksiksizliği desteklemek için kalitenin sezgisel ve yaratıcı şekilde ortaya çıkarılması (!)
5. Dengeleme kararlarını desteklemek için verimli ortaya çıkarma ve NFR önceliklendirme için odaklanmış çaba
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, pazardaki rakip ürünlerin nitelikleri analiz edilerek keşfedilebilir. Örneğin, rakip bir ürün için tepki süresi nedir? Ve daha iyisini mi yapmamız gerekiyor?

NFR'yi keşfetmek için başka bir teknik, bir gereksinim mühendisinin paydaşlardan ve geliştirme ekibinden sorulacak bir anket geliştirdiği önceden oluşturulmuş bir anket kullanmaktır. Örneğin: “Sistem giriş hatalarına nasıl yanıt vermelidir? Sistemin hangi bölümleri daha sonra değiştirilmeye aday olabilir? Sistemin hangi verileri güvenli olmalı?” Bu sorular hazırlanırken standarttaki her bir NFR türü hakkında odaklanmak ve sorular sormak için bazı şablonları veya standartları (örneğin, ISO 9126) izlenerek hazırlanabilir.

Açıklama Özeti

Bu tur, birçok ortaya çıkarma tekniği içeriyor ve her birinin yol boyunca tartışılan avantajları ve dezavantajları var. Açıkça, bu tekniklerin bazıları çok genel, bazıları çok özel, bazıları paydaş bilgisine çok fazla güveniyor, bazıları yetersiz, vb. Bu nedenle, gereksinimlerin ortaya çıkarılması sorununu başarılı bir şekilde ele almak için bazı teknik kombinasyonlarının gerekli olduğu açıktır.

Gereksinim Ortaya Çıkarma Tekniklerinin Hangi Kombinasyonu Kullanılmalıdır?

Gereksinim belirleme tekniklerinin uygun bir karışımını seçme konusunda rehberlik sağlayacak çok az araştırma vardır. Kayda değer bir istisna, kullanıcıları bir kütüphaneden bir gösterim kombinasyonu seçmeye yönlendiren gereksinim mühendisliği tekniklerinin seçimine (KASRET) yönelik bilgiye dayalı yaklaşımdır (Eberlein ve Jiang 2011). Teknikler kütüphanesi ve atama algoritması, bir literatür incelemesine ve endüstriyel ve akademik uzmanların anketine dayanmaktadır. Bununla birlikte, teknik henüz yaygın olarak kullanılmamıştır.

Uygun ortaya çıkarma teknikleri hakkında bir miktar rehberlik sağlamak için, önce daha önce tartışılan teknikleri, tekniklerin ortaya çıkarması muhtemel bilgi türlerine dayalı olarak kategoriler veya denklik sınıfları halinde kümeleriz. Sınıflar (görüşmeler, alan odaklı, grup çalışması, etnografi, prototip oluşturma, hedefler, senaryolar, bakış açıları) ve dahil edilen ortaya çıkarma teknikleri Tablo 3.2'de gösterilmiştir.

Şimdi, Tablo 3.3'te gösterildiği gibi (Zowghi ve Coulin'in çalışmasına dayalı olarak 1998) ortaya çıkarma sürecinin çeşitli yönleriyle başa çıkmada çeşitli tekniklerin ne kadar etkili olduğunu özetleyebiliriz.

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. Diğer durumlarda, bu teknikler birbirini tamamlar. Tablo 3.4'te alternatif (A) ve tamamlayıcı (C) ortaya çıkarma gruplamaları gösterilmektedir.

Uygun bir ortaya çıkarma teknikleri seti seçmenizde size rehberlik etmesi için Tablo 3.2'den 3.4'e kadar olan tabloları kullanabilirsiniz. Kullanılacak bir dizi teknik seçerken, bir dizi tamamlayıcı teknik seçecektir.

Tablo 3.2 Çeşitli Ortaya Çıkarma Tekniklerinin Türe Göre Kabaca Düzenlenmesi

<i>Teknik Türü</i>	<i>Teknikler</i>
Etki Alanı Odaklı	Kart Sıralama Acemi Tasarımcı Alan Analizi Laddering Protokol Analizi Görev Analizi
Etnografya	Etnografik Gözlem
Hedefler	Hedefe dayalı yaklaşımlar QFD
Grup çalışması	Beyin fırtınası Grup çalışması JAD Atölyeler
Röportajlar	Röportajlar İç gözlem Anketler
Prototipleme	Prototipleme
Senaryolar	Senaryolar Kullanım durumları(Use cases) Kullanıcı hikâyeleri (Use stories)
Bakış açıları	Bakış Açıları Repertuar sistemleri

Kaynak: Zowghi, D., & Coulin, C., Gereksinimler ortaya çıkarma: Teknikler, yaklaşımlar ve araçlarla ilgili bir anket, A. Aurum & C. Wohlin (Ed.), Engineering and Managed Software Requirements, s. 19– 46. Springer,2005.

Örneğin, bakış açısı analizi ve bir tür prototiplemenin bir kombinasyonu arzu edilir. Tersine, hem bakış açısı analizini hem de senaryo oluşturmayı kullanmak muhtemelen aşırı derecede gereksiz bilgi verecektir.

Örnek olarak, bir hastanedeki insanları izlemek için bir IoT sağlık sistemi örneğini düşünün. Burada, sistemin genel hedeflerini ve istenen sonuçları tanımlamaya yönelik bir girişimle başlamak uygun olacaktır. Ardından, sistem için geçerli olan yasaları, düzenlemeleri ve standartları keşfetmek için bir alan analizi kullanılır.

	<i>Röportajlar</i>	<i>Alan</i>	<i>Grup çalışması</i>	<i>Etnografya</i>	<i>Prototip</i>	<i>Hedefler</i>	<i>Senaryolar</i>	<i>Bakış Açıları</i>
Etki Alanını Anlama	•	•	•	•		•	•	•
Gereksinim kaynaklarının belirlenmesi	•	•	•			•	•	•
Paydaşları analiz etmek	•	•	•	•	•	•	•	•
Teknikleri ve yaklaşımları seçmek	•	•	•					
Gereksinimleri ortaya çıkarmak	•	•	•	•	•	•	•	•

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

	<i>Röportajlar</i>	<i>Alan</i>	<i>Grup çalışması</i>	<i>Etnografya</i>	<i>Prototipleme</i>	<i>Hedefler</i>	<i>Senaryolar</i>	<i>Bakış açıları</i>
<i>Röportajlar</i>		C	A	A	A	C	C	C
<i>Alan</i>	C		C	A	A	A	A	A
<i>Grup çalışması</i>	A	C		A	C	C	C	C

<i>Etnografya</i>	A	A	A		C	C	A	A
<i>Prototipleme</i>	A	A	C	C		C	C	C
<i>Hedefler</i>	C	A	C	C	C		C	C
<i>Senaryolar</i>	C	A	C	A	C	C		A
<i>Bakış açıları</i>	C	A	C	A	C	C	A	

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

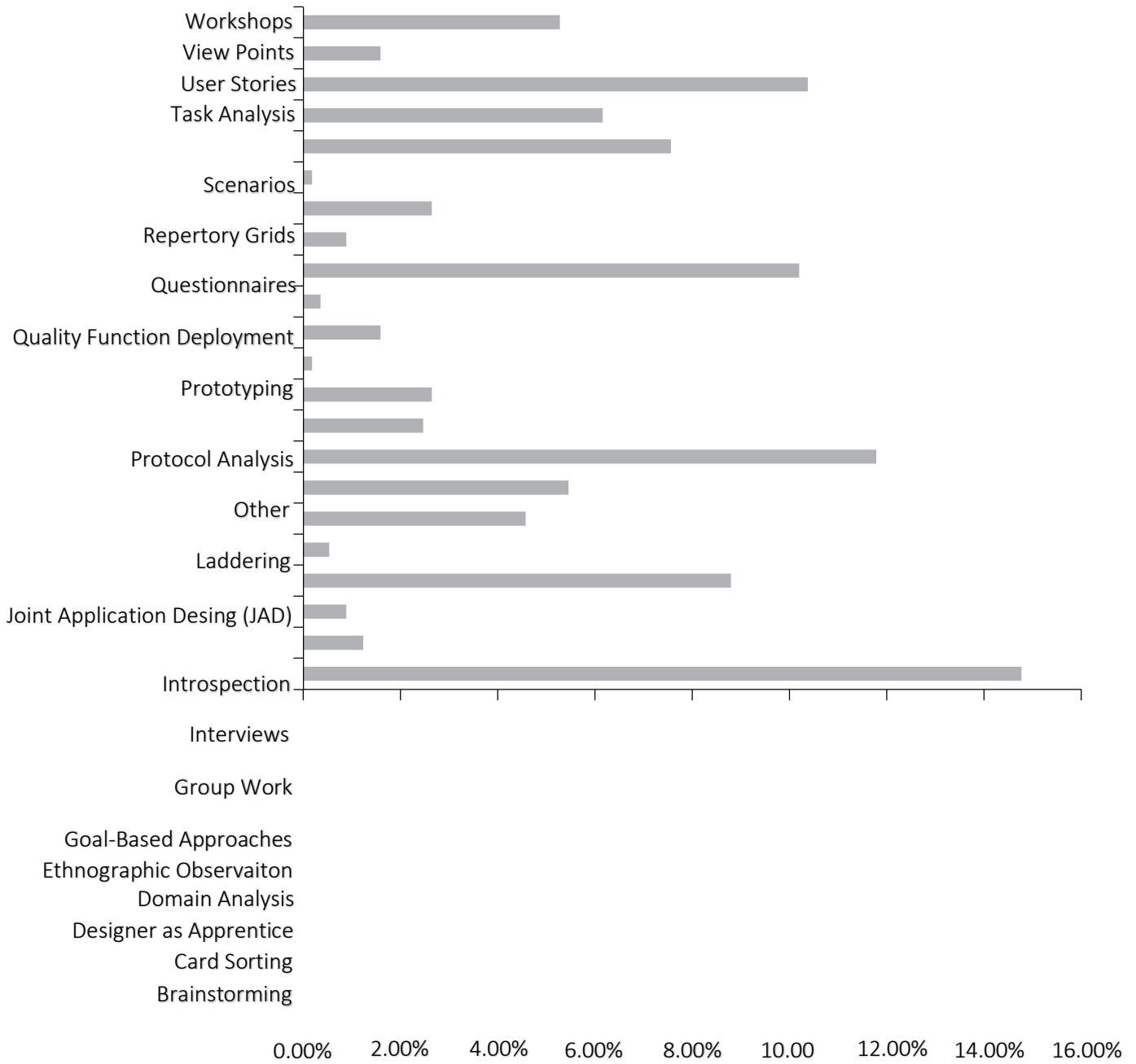
Kullanıcı hikâyeleri, senaryolar ve görüşmeler, sistem kullanıcılarından gereksinimlerin ortaya çıkarılması için uygun olacaktır. Gereksinim bulma ve iyileştirme süreci boyunca kuşkusuz bazı prototipleme olacaktır. Gereksinim keşfi ilerledikçe bu tekniklerin her biri resmi olmayandan daha titiz hale gelecektir. Bu yaklaşım genellikle federal, eyalet ve belediye yönetimleri ve hatta büyük endüstriyel projeler için kullanılır.

Bununla birlikte, ortaya çıkarma tekniklerinin "gümüş kurşun" kombinasyonu yoktur. Doğru karışım, uygulama alanına, müşteri organizasyonunun ve gereksinim mühendisinin kültürüne, projenin boyutuna ve diğer birçok faktöre bağlı olacaktır. Bu konuda deneyimlerden ders almak çok önemlidir.

Gereksinimlerin Ortaya Çıkması Tekniklerinin Yaygınlığı

Bu tartışmayı bitirmeden önce, endüstride çeşitli ortaya çıkarma tekniklerinin nasıl yaygın olarak kullanıldığı hakkında bir fikir edinelim. Bunu yapmak için yazılım uzmanlarının anketine dönüyoruz (Kassab ve diğerleri 2014). Hangi gereksinim belirleme teknik(ler)ini kullanıyorsunuz sorusuna verilen cevapların bir özeti. Şekil 3.9'da gösterilmiştir.

Yanıtlar beyin fırtınası, görüşmeler ve prototip oluşturmanın en sık kullanılan bilgi toplama teknikleri olduğunu ortaya çıkardı (her biri yaklaşık %10). Bu bölümde tartışılan diğer teknikler nispeten seyrek kullanılmıştır.



Şekil 3.9 “Hangi gereksinim belirleme tekniklerini kullanıyorsunuz?” sorusuna verilen yanıtların özeti. (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.)

Tehlikeleri Ortaya Çıkarmak

Daha önce "yapmamalı" davranışlarının istenmeyen çıktı davranışları kümesi olduğunu ve tehlikelerin ciddi veya yıkıcı başarısızlıklara neden olma eğiliminde olan bu davranışların bir alt kümesi olduğunu belirtmiştik. "Ciddi" ve "felaket" terimleri öznel ancak genellikle can kaybını, ciddi yaralanmayı, büyük altyapı hasarını veya 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, şirkete mali zarar verme potansiyeli üçüncü gereksinimden çok daha büyük olduğundan, ilk iki gereksinim tehlike olarak kabul edilebilir.

Tehlikeler, ya doğal olarak meydana gelen (donanım arızaları gibi) ya da yapay olarak meydana gelen (dışarıdan yapılan saldırılar gibi) girdi anormalliklerinin bir fonksiyonudur (Voas ve Laplante 2010). Sistem için uygun bir "yapmayacak" gereksinimleri seti geliştirmek için, gereksinimlerin ortaya çıkarılması aşamasında bu anormal girdi olaylarının tanımlanması ve bunların ortaya çıkan arıza modlarının ve kritikliklerinin belirlenmesi gerekir. Diğer gereksinimlerde olduğu gibi, "yapmayacak" gereksinimlerine de öncelik verilmesi gerekir.

Tehlike belirlemeye yönelik tipik teknikler arasında, kötüye kullanım durumlarının geleneksel olarak geliştirilmesi, antimodelleme ve resmi yöntemler yer alır (Robinson 2010). Sistemin veya ilgili sistemlerin önceki sürümlerinden oluşturulan istenmeyen davranışların kontrol listeleri de istenmeyen davranışların belirlenmesinde yardımcı olur. Geçerli standartlar ve yönetmelikler ayrıca belirli "yapılmaması gereken" gereksinimleri de içerebilir, örneğin, Amerika Birleşik Devletleri'nde, Sağlık Sigortası Taşınabilirlik ve Sorumluluk Yasası (HIPPA 1996), belirli kişisel bilgilerin yetkisiz taraflara verilmesini yasaklar ve tüm yargı alanlarındaki standart inşaat kuralları, belirli inşaat uygulamalarına karşı çok sayıda yasak içerir.

Kötüye Kullanım Durumları

Kullanım senaryoları kısaca istenen davranışın yapılandırılmış açıklamalarıdır. İstenen davranışı tanımlayan kullanım durumları olduğu gibi, istenmeyen davranışları tanımlayan kötüye kullanım durumları (Misuse Cases ya da Abuse Cases) vardır. Çoğu sistem için tipik kötüye kullanımlar, güvenlik ihlallerini ve diğer kötü niyetli davranışların yanı sıra eğitimsiz, yönünü şaşırılmış veya yeteneksiz kullanıcılar tarafından kötüye kullanımı içerir. Cleland-Huang ve ark. (2016), tehdit modelleme ve beyin fırtınasına dayalı olarak kötüye kullanım vakalarını belirlemenin çeşitli yollarını açıklar.

Kullanım senaryoları oluşturmanın kolay bir yolu, istenmeyen bir sistem kullanıcısı olan istenmeyen bir kişi rolünü üstlenmek ve ardından böyle bir kişinin davranışlarını modellemektir (Cleland-Huang 2014). İstenmeyen kişiler, bilgisayar korsanlarını, davetsiz misafirleri, casusları ve hatta iyi niyetli, ancak beceriksiz kullanıcıları içerebilir.

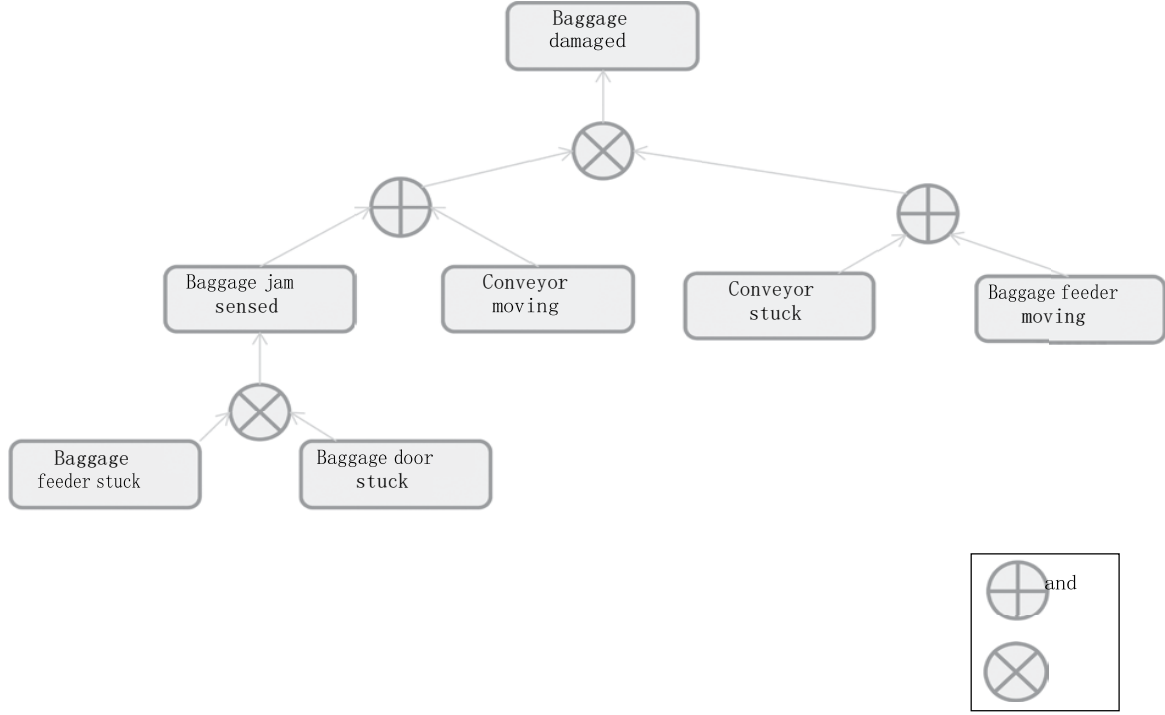
Bu kişilerin belirlenmesinin tehlike gereksinimleri oluşturmaya nasıl yardımcı olduğunu görmek için aşağıdaki örnekleri inceleyin. Pet shop POS sisteminde, bir hacker'ın bu sisteme nasıl sızacağını düşünmek ve daha sonra hacker'ın niyetini bozacak gereksinimler yaratmak uygun olacaktır. Bagaj taşıma sisteminde, bir gereksinim mühendisi beceriksiz veya dalgın bir traveler rolünü üstlenebilir ve ardından bu kişilerin güvenliğini sağlayacak gereksinimleri belirleyebilir.

Kötüye kullanım vakaları yaratma ihtiyacı, tüm olumsuz paydaşları tam olarak belirlemek için bir nedendir, çünkü bu kişiler büyük olasılıkla istenmeyen kişilerden oluşacaktır.

Antimodeller

İstenmeyen davranışı elde etmenin başka bir yolu, sistem için antimodeller yaratmaktır. Antimodeller hata ağaçlarıyla ilgilidir, yani model, sistem hatasına yol açan istenmeyen davranışlar için bir neden-sonuç hiyerarşisi yaratılarak türetilir. Ardından, sistem arızasının nedenleri “yapmayacak” gereksinimleri oluşturmak için kullanılır.

Örneğin, Şekil 3.10'da gösterilen hasarlı bagajın istenmeyen sonucunu içeren bagaj taşıma sistemi için güvenlik işlevselliğini düşünün.



Şekil 3.10 Bagaj taşıma sistemi için kısmi antimodel.

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 Bagaj sıkışması algılanırsa konveyör hareket etmemelidir. antimodel, bu ham gereksinimleri sistematik bir şekilde türetmemize yardımcı oldu.

Resmi Yöntemler

Bölüm 6'da tartışılacağı üzere, amaçları, işlemleri, gereksinimleri ve kısıtlamaları ile ilgili olarak sistem ve çevresinin bir modelini oluşturmak için matematiksel formalizmler kullanılabilir. Bu formalizmler daha sonra sistemin çeşitli özelliklerini incelemek ve istenmeyenlerin bulunmadığından emin olmak için otomatik model denetleyicileri ile birlikte

kullanılabilir. Örneğin, davranışı modellemek için UML/SysML kullanılıyorsa, resmi etkinlik diyagramlarına güvenlik endişeleri eklenebilir. Aktivite diyagramları Ek C, UML'de açıklanmıştır.

VIGNETTE 3.1 Güvenlik Açısından Kritik Sistemler için Gereksinimler Mühendisliği

Güvenlik açısından kritik sistemler, arızanın insan yaşamının kaybına veya önemli yaralanmalara neden olabileceği sistemlerdir. Bu tür sistemler, elektrik ve su için kamu altyapısını, tıbbi sistemleri, ulaşım sistemlerini ve çok daha fazlasını içerir. Akıllı otoyollar, sürücüsüz araçlar ve robotik cerrahi gibi akıllı ve otonom sistemlerin büyümesiyle birlikte güvenlik gereksinimlerine odaklanmak çok önemli hale geldi.

Güvenlik açısından kritik sistemler için gereksinimlerin geliştirilmesi için rehberlik sağlayan bazı alana özgü standartlar (örneğin nükleer, tıbbi cihazlar, havacılık) ve genel standartlar vardır. Genel bir standardın bir örneği, yazılım yoğun sistemlerde güvenlik hususları için rehberlik sağlayan NASA-STD-8719.13'tür. (NASA 2)

Bu standart, belirli bir tehlikenin olasılığını ve ciddiyetini göstermek için bir risk değerlendirme matrisi tablosu kullanır (Şekil 3.11).

Sistem önem seviyeleri	Sistem tehlikesi oluşma olasılığı				
	İmkansız	Olası Olmayan	Olası	Muhtemel	Büyük İhtimal
Felaket	4	3	2	1	1
Kritik	5	4	3	2	1
Orta	6	5	4	3	2
İhmal edilebilir	7	6	5	4	3

Standart, güvenlik gereksinimlerinin, gereksinimlerdeki etkilerin ve deęişikliklerin deęerlendirilmesini saęlamak için ihtiya duyulan mühendislik yaşam döngüsü boyunca izlenebilirlik için etiketlenmesini şart koşar.

1 =En yüksek öncelik (en yüksek sistem riski), 7 = En düşük öncelik (en düşük sistem riski).

Şekil 3.11 NASA-STD-8719.13C'nin sistem risk indeksi matrisi. (NASA'dan [Ulusal Havacılık ve Uzay Dairesi], NASA-STD 8719.13, Yazılım Güvenlięi Standardı, 2013, <http://NASA.gov> [Ocak 2017'ye erişildi].)

Benzersiz olarak tanımlanan tehlikeler, gereksinimler belgesindeki özel bir bölümde listelenebilir veya gereksinimin yanında bir bayrakla belirtilebilir veya bir gereksinim yönetimi aracı içinde etiketlenebilir.

Yazılım güvenlięi gereksinimleri, güvenli olmayan sistem davranışına karşı koruma saęlamaktan fazlasını yapabilir. Yazılım, sistemi izlemek, kritik verileri analiz etmek, eğilimleri aramak ve tehlikeli bir durumun habercisi olabilecek olaylar meydana geldiğinde sinyal vermek veya harekete geçmek için proaktif olarak kullanılabilir. Böyle bir gösterge tespit edildiğinde, bu tehlikenin etkilerini önlemek veya azaltmak için yazılım kullanılabilir. Önleme veya azaltma, sistemin tamamen veya kısmen geri yüklenmesini veya sistemin güvenli bir duruma getirilmesini içerebilir.

2. sınıf 1. Öğretim A şubesi Chapter3

Hazırlayanlar:

Emin Duran

Ahmet Eren Çelik

Barış Yesari

Muharrem Candan