

BÖLÜM 4

GEREKSİNİMLER BELGESİNİN YAZILMASI

Gereksinim Anlaşması ve Analizi

Ortaya çıkarma faaliyeti boyunca ve özellikle sistem gereksinimleri spesifikasyonunu tamamlamadan önce, SRS ham gereksinimleri problemler için analiz edilmeli ve bu problemler uzlaştırılmalıdır. Gereksinim analizi, problemler için gereksinimleri analiz etme faaliyetidir. Sorunlu gereksinimler şunları içerir:

- Kafa karıştırıcı
- Yabancı
- Yinelenen
- Çakışan
- Eksik

Bu kavramları Bölüm 5'te daha katı hale getiriyoruz.

Gereksinim analizi genellikle gayri resmidir; örneğin, müşterilere ve paydaş temsilcilerine, gereksinim mühendisinin tüm müşterilerin isteklerine ilişkin yorumunun doğru olduğundan emin olmak için anlayabilecekleri bir biçimde ortaya çıkan gereksinimler gösterilir. Resmi yöntemler, gereksinim analizi için daha titiz teknikler sağlar.

Gereksinim anlaşması, farklı kaynaklardan elde edilen aynı gereksinimdeki farklılıkların uzlaştırılması sürecidir. Gereksinim anlaşması genellikle gayri resmi bir süreçtir, ancak sistematik bir süreç kullanılmalıdır. Kullanım senaryoları ve kullanıcı hikayeleri genellikle bu amaç için etkili araçlardır (Hull ve diğerleri 2011). Ekran maketleri, özellikle işlevsel prototiplerle birlikte kullanıldığında başka bir yararlı tekniktir (Ricca ve ark. 2010). Diğerleri, gereksinim anlaşması için senaryolar, resimli taslaklar, kağıt Prototipler, komut dosyasıyla oluşturulmuş kavram oluşturucular ve işlevsel prototipler kullanılmasını önerir (Sutcliffe ve diğerleri, 2011).

Hull ve diğerleri (2011), müşteri sözleşmesi sürecinin bir parçası olarak aşağıdaki soruların sorulmasını önermektedir:

- Gereksinim tamamlandı mı?

- Gereklilik açık mı?
- Gereklilik uygulanabilir mi?
- Yeterlilik planı açık ve kabul edilebilir mi?

Bölüm 5'te bu soruların bir gereksinim belirtiminin istenen nitelikleriyle nasıl ilişkili olduğunu not edeceksiniz ve Bölüm 5 ve 6'da gereksinim mühendisliği yaşam döngüsü boyunca bu nitelikleri elde etmenin daha kesin yollarını inceleyeceğiz.

REQUIREMENTS REPRESENTATION

GEREKLİLİKLER TEMSİL

Kullanım durumları, kullanıcı öyküleri, doğal diller, biçimsel diller, stilize edilmiş (sınırlandırılmış doğal) diller ve çeşitli biçimsel, yarı biçimsel ve biçimsel olmayan diyagramlar dahil olmak üzere herhangi bir sistemdeki işlevselliği tanımlamak için çeşitli teknikler kullanılabilir. Eberlein ve Jiang (2011), spesifikasyon yaklaşımları dahil olmak üzere çeşitli gereksinim tekniklerinin seçilmesine yönelik odaklanmış bir tartışma sunar. Bu tür seçimleri yaparken aşağıdaki faktörleri göz önünde bulundurmaya önerirler:

- Uygulanmakta olan tekniğin olgunluğu ve etkinliği gibi teknik konular düşünülen
 - Tekniklerin karmaşıklığı
 - Organizasyonun değişime karşı çıkması gibi sosyal ve kültürel konular
 - Geliştiricilerin eğitim ve bilgi düzeyi

Zaman ve maliyet kısıtlamaları, projenin karmaşıklığı ve yazılım ekibinin yapısı ve yetkinliği gibi yazılım projesinin belirli özellikleri. Tek bir proje veya kuruluş çapında kullanım için doğru temsil stilini seçmek önemli bir zorluktur ve bir anlamda bu kitabın bir temasıdır.

Approaches to requirements representation

Gereksinimlerin temsiline yönelik yaklaşımlar

Genel olarak, gereksinim gösteriminin üç biçimi vardır: resmi, gayri resmi ve yarı resmi. Gereksinim belirtimleri, bu yaklaşımlardan birine veya diğerine sıkı sıkıya bağlı kalabilir, ancak genellikle bu yaklaşımlardan en az ikisinin (resmi olmayan ve bir diğeri) unsurlarını içerirler. Resmi temsillerin katı, matematiksel bir temeli vardır ve bunları Bölüm 6'da daha ayrıntılı olarak inceleyeceğiz. Ancak resmi gereksinim belirim belgelerinde bile resmi olmayan veya yarı biçimsel belirim unsurları olacaktır.

Gayri resmi gereksinimleri temsil teknikleri, kesin bir matematiksel gösterime tamamen dönüştürülemez. Resmi olmayan teknikler arasında doğal dil (yani insan dilleri), akış çizelgeleri, özel diyagramlar ve (SRS) görmeye alışık olabileceğiniz öğelerin çoğu yer alır. Aslında, tüm SRS belgelerinin bazı gayri resmi unsurları olacaktır. Bu gerçeği güvenle söyleyebiliriz, çünkü en resmi gereksinim belirtim belgeleri, yalnızca resmi bir öğeyi tanıtmak için bile olsa, doğal dili kullanmak zorundadır. Bu nedenle, dikkatimizi çoğunlukla resmi olmayan teknikleri kullanarak gereksinimlerin temsiline odaklayacağız.

Son olarak, yarı biçimsel temsil teknikleri, gayri resmi görüneler de en azından kısmi bir biçimsel temele sahip olanları içerir. Örneğin, birleşik modelleme dili (UML) veya sistem modelleme dili (SysML) metamodelleme dilleri ailesindeki, kullanım durumu diyagramı da dahil olmak üzere birçok diyagram gayri resmidir veya resmileştirilebilir. UML ve SysML genellikle yarı biçimsel modelleme teknikleri olarak kabul edilir; ancak uygun mekanizmaların eklenmesiyle tamamen resmi hale getirilebilirler (Laplante 2006).

Resmi gereksinim modellemesi Bölüm 6'da tartışılmaktadır. Endüstride kullanılan baskın yaklaşım ve en problemlisi olanı olduğundan, bu bölümün çoğunluğu doğal dil kullanılarak yapılan resmi olmayan modellemeye odaklanmaktadır.

Kasab et al. (2014), katılımcıların %60'ının gereksinimlerin doğal dil cinsinden ifade edildiğini, yani gayri resmi olarak bildirdiğini tespit etti (Şekil 4.1).

Bildirilen yarı biçimsel yaklaşımlar, UML/SysML diyagramlarını içeriyordu. Z ve VDM dahil olmak üzere rapor edilen resmi yaklaşımlar Bölüm 6'da tartışılmaktadır.

Ankete katılanların çoğunluğu endüstride resmi olmayan şartnamelerle karşılaştıklarını veya kullandıklarını ifade ederken, muhtemelen en resmi olmayan şartname belgelerinin bile bazı yarı resmi ve resmi unsurlara sahip olması muhtemeldir. Aynı şekilde, en resmi spesifikasyonlar bile bazı gayri resmi unsurlara sahip olacaktır.

Tablo

OSI IEC IEEE STANDARD 29148

2011 yılında, üç uluslararası standardizasyon kuruluşu, Uluslararası Standartlar Organizasyonu (ISO), Uluslararası Elektroteknik Komisyonu (IEC) ve Elektrik ve Elektronik Mühendisleri Enstitüsü (IEEE) birlikte ISO/IEC/IEEE Standardı 29148'i (2011) yayınladı. Sistemler ve Yazılım Mühendisliği Yaşam Döngüsü Süreçleri—Gereksinimler Mühendisliği için bu standart, birkaç farklı standardın uyumlu hale getirilmesinin bir sonucudur. Bu standartlardan biri olan IEEE Standard 830–1998, Yazılım Gereksinimleri Spesifikasyonları için Önerilen Uygulama, yazılım gereksinimlerinin spesifikasyonu için önerilen yaklaşımları açıklamıştır. Bu metnin önceki baskılarında IEEE Standard 830'a sıklıkla atıfta bulunulmuştur. Standart 29148, tam sistem yaşam döngüsünü dahil etmek, çevik ve yalın mühendislikteki gelişmeleri tanımak ve genel bir yazılım durumu olarak yazılım yoğun sistemlere odaklanmak için IEEE 830'da

geniřletildi. Kolaylık olması aısından, bundan sonra ISO/IEC/IEEE 29148, IEEE 29148 olarak anılacaktır.

IEEE 29148, ařađıdakilere yardımcı olan bir belge üreten bir modeli temel alır:

- Ürünün ne yapacağı konusunda alıcılar veya tedarikçiler arasında anlaşmanın temelini oluşturmak
- Tasarım başlamadan önce gereksinimlerin titiz bir şekilde değerlendirilmesini sağlamak ve daha sonraki yeniden tasarımı azaltmak
- Ürün maliyetlerini ve risklerini tahmin etmek için gerçekçi bir temel sağlamak ve kuruluşların doğrulama ve doğrulama planları geliřtirmesini sağlamak
- Bir ürünü yeni kullanıcılara veya yeni operasyonel ortamlara dağıtmak için bilinçli bir temel sağlamak
- Ürün geliřtirme için bir temel sağlamak

IEEE 29148 yönergelerine uymanın çeřitli faydaları vardır. İlk olarak, gereksinimler belgesinin organizasyonu için basit bir çerçeve sağlarlar. Ayrıca, IEEE 29148 taslađı, geniř bir uygulama alanı yelpazesinde yaygın olarak dağıtıldıđından, gereksinim mühendisi için özellikle faydalıdır.

Son olarak ve belki daha da önemlisi, IEEE 29148, SRS'nin işlevsel ve işlevsel olmayan gereksinimlerini organize etmek için rehberlik sağlar. Standart ayrıca Bölüm 3, Özel Gereksinimler altında işlevsel ve işlevsel olmayan gereksinimleri temsil etmenin yollarını açıklar. Bu gereksinimler bir sonraki tartışma konusudur.

TABLO

İşlevsel Olmayan Gereksinimler

IEEE , çeřitli işlevsel olmayan gereksinimleri ařađıda açıklar:

Harici arayüzler

Kullanılabilirlik

Performans

Mantıksal veritabanı konuları

Tasarım kısıtlamaları

Standartlara Uygunluk

Yazılım sisteminin öznitelik gereksinimleri

Dıř arabirim gereksinimleri, ařađıdakiler dahil olmak üzere çeřitli şekillerde düzenlenebilir:

Öğenin Adı

Amaç açıklaması

Girdi kaynağı veya çıktı hedefi

Geçerli aralık, doğruluk ve/veya tolerans

- Ölçü Birimleri
- Zamanlama
- Diğer girdiler/çıktılar ile ilişkiler
- Ekran biçimleri/organizasyonu
- Pencere biçimleri/organizasyonu
- Veri biçimleri
- Komut biçimleri

Kullanılabilirlik gereksinimleri, kullanıcı ihtiyaçlarını karşılayan gereksinimleri sağlar. Müşteri memnuniyeti açısından, kullanılabilirlik genellikle en önemli gereksinim sınıfıdır. Prototipleme, bu tür gereksinimleri keşfetmede yardımcı olabilir. Performans gereksinimleri, statik ve dinamik gereksinimlerdir. Tipik performans gereksinimleri, desteklenecek eşzamanlı kullanıcı sayısını içerebilir, işlem ve görev sayısı ve içinde işlenecek veri miktarı hem normal hem de yoğun iş yükü koşulları için belirli zaman süreleri vardır. mantıksal veritabanı gereksinimleri, aşağıdakiler gibi çeşitli işlevler tarafından kullanılan bilgi türleridir:

- Kullanım sıklığı
- Erişim Özellikleri
- Veri varlıkları ve ilişkileri
- Bütünlük Sınırlamaları
- Veri saklama gereksinimleri

Tasarım kısıtlaması gereksinimleri standartlara uygunluk ve donanım sınırlamalarıyla ilgilidir. Kısa bir süre sonra gereksinim ifadelerinde kısıtlamaların nasıl organize edileceğini göreceğiz.

Standartlara uygunluk, yürürlükteki yasalardan elde edilen gereklilikleri belirtir. Bu belgelerin ortaya çıkarken doğru ve eksiksiz bir şekilde tanımlanması çok önemlidir.

Son olarak, yazılım veya sistem öznitelikleri genellikle güvenilirlik, kullanılabilirlik, güvenlik, bakım, taşınabilirlik ve diğerlerini içerebilir.

Genel olarak, bu tür herhangi bir nitelik, bir "beceriksizlik" olarak adlandırılır, çünkü çoğu "beceriksizlik" ile biter.

Fonksiyonel gereksinimleri temsil etmeye yönelik tavsiyeler

İşlevsel gereksinimler, tüm sistem girdilerini ve her girdi olasılığı için normal ve anormal durumlara verilen işlemlerin ve yanıtların (çıktıların) tam sırasını kapsmalıdır. İşlevsel gereksinimler, duruma göre açıklamayı veya diğer genel açıklama biçimlerini kullanabilir

IEEE 29148, işlevsel Gereksinimleri.

İşlevsel gereksinimler şu şekilde düzenlenebilir:

- Sistem modu (örneğin navigasyon, savaş, teşhis)
- Kullanıcı sınıfı(örneğin kullanıcı,süpervizör)
- Nesnelere(sınıf tanımlama/nesnelere/Öznitelikler, metodlar ve mesajlar)
- Özellikler (Sistemin kullanıcılara ne sağladığını açıklar)
- Uyarıcılar (örneğin sensör 1,sensör2...)
- İşlevsel hiyerarşi (örneğin görevlerin yukarıdan aşağıya ayrıştırılmasında kullanma)

Belgeyi sistem yönetimi modunda gözden geçirirken, işletim sistemi işlevselliği aşağıdaki gibi açıklanmıştır:

İşletim Sistemi

001

İşletim sistemi, çoklu görev desteği, CPU zamanlaması gibi gerçek zamanlı sistemleri desteklemek, temel iletişim ve bellek yönetimi gibi gerekli ortak bir mekanizma seti sağlamalıdır.

001.1

İşletim sistemi çoklu görev yetenekleri sağlamalıdır.

001.2

İşletim sistemi bu işlemi olay odaklı, öncelik tabanlı sağlayacaktır.

001.2.1

Görev yürütme, bir görevin önceliği tarafından denetlenecek ve yürütülmesi için gereken kaynakların kullanılabilir.

001.3

İşletim sistemi, görevlerarası iletişim ve senkronizasyon için destek sağlayacaktır.

001.4

İşletim sistemi gerçek zamanlı saat desteği sağlayacaktır.

001.5

İşletim sistemi yazılımı, 80387 matematik ortak işlemcisi için görev düzeyi bağlam geçişi sağlayacaktır.

Ayrıca sistem işlevselliği altında komut doğrulaması, aşağıdaki gibi tanımlanır;

211 Uçuş yazılımı CCSDS(Uzay Veri Sistemleri Danışma Komitesi) komut yapısı doğrulaması gerçekleştirecektir.

211.1

Uçuş yazılımı CCSDS Komuta Operasyonlarını uygulayacaktır CCSDS aktarım çerçevelerini doğrulamak için prosedür numarası 1'e (COP-1) doğru ve sırayla alındı.

211.2

Uçuş yazılımı, sabit boyutlu bir çerçeve kabulünü destekleyecek ve 127 ve sabit bir raporlama mekanizması (ÇİFTLİK) sürgülü pencere ÇİFTLİK 63'ün negatif kenarında.

211.3

Uçuş yazılımı durumu uzaklıkölçer ve gerçek zamanlı atılacak komut paketinde aşağıdaki hatalardan biri oluşursa:

Sağlama toplamı verilmeden önce doğrulamada başarısız oluyor.

Geçersiz bir uzunluk algılandı

Geçersiz bir uygulama kimliği algılandı

211.4

uçuş yazılımı bir komut bağlantısı kontrol kelimesioluşturacak ve koruyacaktır. Her güncelleştirme yapıldığında, paket olası aşağı bağlantı için biçimlendirilir ve yönlendirilir. (NASA WIRE 1996).

Yazılım tabanlı sistemlerde de nesne yönelimli bir sistem almak çok yaygındır. Bu durum özellikle Java gibi saf nesne yönelimli bir dil kullanılarak yazılım inşa edilmesi beklenir .

Bu tekniklerin bir kombinasyonu bir SRS belgesinde kullanılabilir.

Bu tekniklerin bir kombinasyonu bir SRS belgesinde kullanılabilir. Örneğin, Ek A'da verilen akıllı ev açıklaması, işlevselliğin özellik odaklı bir açıklamasıdır. Özellik tahrikli olduğuna dair bir ipucu MANTRADIR gereksinimlerin çoğu için "sistem olacaktır".

Alternatif olarak, işlevsel mod tarafından düzenlenen bir sistemi düşünün - NASA WIRE (geniş alan kızılötesi kaşif) sistemi (1996).

Bu sistem, Küçük Kaşif programı ve bir milimetre altı dalga astronomi uydusunu tanımlar ve uzaydan karaya iletişimi içerir.

Bilgi, yere iletilen ve yerden alınan komutlar, Uzay Veri Sistemleri Danışma Komitesi (CCSDS) ve Goddard'a Uzay Uçuş Merkezi standartlarından alınmıştır. (NASA WIRE 1996).

Uçuş yazılımı gereksinimleri aşağıdaki gibi düzenlenir;

- Sistem Yönetimi
- Komut Yönetimi
- Telemetri Yönetimi
- Yararlı Yük Yönetimi
- Sağlık ve güvenlik yönetimi
- Yazılım Yönetimi
- performans gereksinimleri

Nesne yönelimli gösterimler son derece soyut sistem bileşenlerini içerir nesnelere olarak adlandırılır ve kapsüllenmiş öznitelikleri ve davranışları vardır.

Sistemlerin geleneksel yapılandırılmış açıklamaları ile sistemlerin nesne yönelimli açıklamaları arasındaki farklar Tablo 4.1'de özetlenmiştir.

Nesne yönelimli bir şekilde düzenlenmiş gereksinimlerle uğraşırken, kullanıcı hikayelerini (özellikle Bölüm 7'de tartışacağımız çevik yazılım geliştirme metodolojileriyle birlikte) kullanmak veya servis taleplerini kullanmak çok tipiktir. Bu davranışları açıklamak için örnek olay diyagramları esas alınır.

Tablo 4.1

	Yapılandırılmış	Nesne Yönelimli
Sistem bileşenleri	İşlev	Nesneler
Veri ve denetim şartnamesi	İç ayrışma ile ayrılmış	Nesne içinde <u>kapsüllenmiş</u>
Özellikler	Hiyerarşik yapı	Nesnelerin bilgi aktarımı
	İşlevsel açıklama sistemi	Davranışsal Açıklama Sistemi
	Bilgilerin işlevle <u>kapsüllenmesi</u>	Bilgilerin nesneyle <u>kapsüllenmesi</u>

UML/SysML

UML, yazılım ve sistemlerde kullanılan bir modelleme notasyonları kümesidir mühendislik gereksinimleri spesifikasyonu, tasarım, sistem analizi ve daha fazlası kullanılır. Bu, bir sonraki SysML, UML'nin genel sistemlere ve iş alanlarına odaklanan bir varyasyonudur. UML ve SysML oldukça benzerdir. Kullanım örnekleri UML ve SysML'de bulunan modellerden biridir ve gereksinim mühendisleri için çok önemlidir.

Gereksinimler belgesi

Sistem (veya yazılım) gereksinimleri spesifikasyon belgesi, sistem geliştiricilerinden nelerin istendiğinin resmi beyanıdır.

Şunu da unutmamak gerekir ki, bu koşullar altında

sponsor ve kurucuları arasında bir müşteri-satıcı ilişkisi vardır.

sistem, SRS bir sözleşmedir ve bu nedenle sivil

sözleşme hukuku (veya belirli dolandırıcılık veya ihmal türleri varsa ceza hukukunu) gösterir).

Gereksinim belgesi kullanıcıları

Her biri benzersiz bakış açısına ve ilgiye sahip bir kaç kullanıcısı var. Belgenin tipik kullanıcıları:

Müşteriler

Yöneticiler

Geliştiriciler

Bakım mühendisleri

Paydaşlar

Müşteriler gereksinimi belirler , ihtiyaçlarını karşılandıklarından yani sağladıklarından emin olmak için bunları gözden geçirirler. Müşteriler ayrıca gereksinimler de ki değişiklikleri de belirtir.

Resmi metodolojiler hariç SRS belgelerine erişilebilmelidir. Bunun nedeni müşteriler bu alana dahildir ve muhtemelen bu meslekten değillerdir bundan dolayı erişebilmelidirler.

Tüm seviyelerdeki yöneticiler, sistem için bir teklif planlar ve sistem geliştirme belgesi için ve aynı zamanda süreyi yönetmeyi de planladıkları için bu belgeyi kullanacaklardır. Bundan dolayı SRS de aradıkları kriterler güçlü maliyet ve tamamlanma süresidir.

Ve elbette, geliştiriciler gereksinim belgesini şu amaçlarla kullanır:

Hangi sistemin geliştirileceğini anlamak.

Test mühendisleri sistem için doğrulama testleri geliştirmek.

Bakım mühendisleri, sistemi ve ilişkiyi anlamaya yardımcı olmak için gereksinim raporlarını kullanacaklardır. Böylece sistem yükseltilebilir veya problemler çözülebilir.

Diğer paydaşlar SRS'yi kullanarak, söz konusu sistemin tüm doğrudan ve dolaylı yararlanıcılarını (veya düşmanlarının) ve ayrıca avukatların, yargıçların, davacıların, jürilerin, SRS'yi yasal bir belge olarak görececek olan avukatlar, hakemler, arabulucular vb. anlaşmazlıklarda bir belge olarak sunulabilir.

GEREK SINİMLER BELGE GEREK SINİMLERİ

SRS için doğru biçim nedir? Birçok olası biçim vardır. Örneğin, IEEE 29148 standardı, gereksinim belirtimi belgesi için genel bir biçim sağlar. Proje Yönetimi Enstitüsü (PMI) gibi diğer standart kuruluşların ve meslek kuruluşlarının standart gereksinim biçimleri ve şablonları vardır. Çoğu gereksinim yönetim yazılımı özelleştirilebilir formatlarda belgeler üretecektir. Ancak "doğru" format, sponsorun, müşterinin, işverenin ve diğer paydaşların neye ihtiyaç duyduğuna bağlıdır. SRS belgesinin değiştirilmesinin kolay olması gerektiği, şimdiye kadar tartıştığımız birçok nedenden dolayı açıktır. Ayrıca, SRS belgesi bakım için bir referans aracı olarak hizmet verdiğinden, sistemin yaşam döngüsü hakkında, yani değişiklikleri tahmin etmek için önceden düşünülmüş olmalıdır.

Genel örgütlenme, yazma yaklaşımı ve söylem açısından en iyi uygulamalar şunlardır:

- Spesifikasyon boyunca tutarlı modelleme yaklaşımları ve teknikleri kullanmak, örneğin yukarıdan aşağıya ayrışma, yapılandırılmış veya nesne yönelimli yaklaşımlar
- Operasyonel belirtimi açıklayıcı davranıştan ayırma
- Modellerde tutarlı soyutlama düzeylerini kullanma ve modeller arasında iyileştirme düzeyleri arasında uygunluk
- Belirli zamanlama özelliklerindeki spesifikasyon modellerinin bir parçası olarak işlev dışı gereksinimleri modelleme
- Teknik özelliklerde donanım ve yazılım atamalarını atlama (belirtim yerine tasarımın başka bir yönü)

Bu kurallara uymak her zaman daha iyi bir SRS belgesine yol açacaktır. Son olarak, IEEE 29148 gereksinim belirtimi belgeleri için bazı istenen nitelikleri açıklar ve bunlar Bölüm 5'te tartışılacaktır.

TERCİH EDİLEN YAZMA STİLİ

Mühendisler (her tür) zayıf iletişim becerileri, özellikle de yazma konusunda haksız bir üne sahiptir. Her durumda, gereklilik belgelerinin çok iyi yazılması gerektiği artık açık olmalıdır.

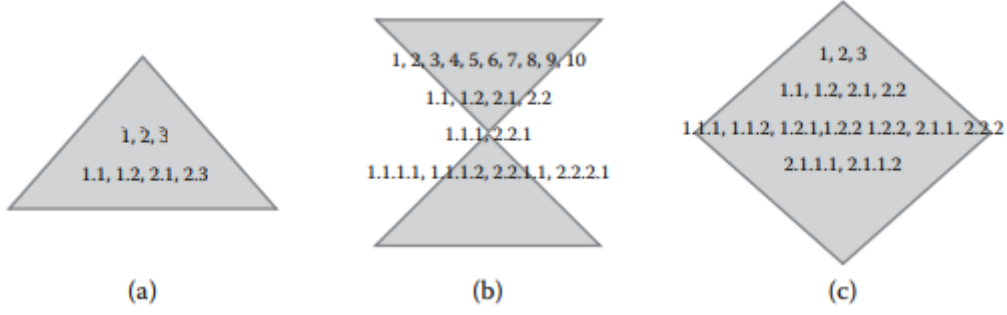
Teknik yazı üzerine her gereksinim mühendisi tarafından incelenmesi gereken mükemmel metinler vardır, örneğin, Laplante (2011). Pratik yaparak ve onlardan öğrenmek için iyi yazılmış SRS belgelerini inceleyerek yazınızı geliştirmeniz istenir. Ayrıca edebiyat, şiir ve haberleri de okumalısınız, çünkü bu yazılardan ekonomi ve sunumun netliği hakkında çok şey öğrenilebilir. Hatta bazıları, senaryo yazmayı gereksinim belgelerini (veya kullanıcı hikayeleri ve kullanım durumları için) yazmak için uygun bir paradigma olarak önermiştir (Norden 2007).

Her durumda, herhangi bir yazının tekrar tekrar yazmak ve yeniden yazmak için hazırlandığı gibi gereksinim belgesine yaklaşın. Gereksinimler belgesinin diğer birkaç paydaş tarafından incelenmesini sağlayın.

METİN YAPISI ANALİZİ

Ölçümler, temel yazma özelliklerinin asayişinde yardımcı olabilir. Çoğu sözcük işleme aracı ortalama sözcük, cümle ve paragraf uzunluğunu hesaplar ve bunlar çok basit veya anlaşılması çok zor olan yazıyı vurgulayabilecekleri için toplanması değerlidir. Bununla birlikte, standart sözcük işlemciler tarafından hesaplanmayan yazmanın önemli bir özelliği numaralandırma yapısı derinliğidir.

Yazılım ve Sistemler için Gereksinim Mühendisliği



Şekil 4.3 (a) Piramit, (b) kum saati ve (c) gereksinimler için elmas şeklindeki konfigürasyonlar metin yapısı

Numaralandırma yapısı derinliği, kaynak belgenin her düzeyinde numaralandırılmış ifadeleri sayan bir ölçümdür. Örneğin, 1.0, 2.0, 3.0 vb. numaralı birinci düzey gereksinimlerin çok üst düzey (soyut) gereksinimler olması beklenir. 1.1, 1.2, 1.3, ... 2.1, 2.2, 2.3, ... 3.1, 3.2, vb. daha düşük bir ayrıntı düzeyindeki alt gereksinimlerdir. Daha ayrıntılı gereksinimler, 1.1.1, 1.1.2 vb. olarak numaralandırılan üçüncü düzeyde bulunacaktır. Bir spesifikasyon dördüncü veya hatta beşinci seviye gereksinimlere kadar devam edebilir, ancak normalde, üçüncü veya dördüncü ayrıntı seviyeleri yeterli olmalıdır. Her durumda, düzey 1, 2, 3 vb.'deki gereksinimlerin sayımları, belgenin organizasyonunun, tutarlılığının ve ayrıntı düzeyinin bir göstergesini sağlar.

İyi organize edilmiş bir SRS belgesi tutarlı bir ayrıntı düzeyine sahip olmalıdır ve her düzeydeki gereksinimleri listeleyecek olursanız, sonuçta ortaya çıkan şekil bir piramit gibi görünmelidir, çünkü düzey 1'de birkaç numaralı ifade olmalı ve her alt düzey, üstündeki düzeyden giderek daha fazla numaralı ifadeye sahip olmalıdır (Şekil 4.3a).

Öte yandan, gereksinimleri her düzeyde sayılan kum saati şekline benzeyen gereksinim belgeleri (Şekil 4.3b) genellikle çok miktarda tanıtıcı ve idari bilgi içeren belgelerdir. Son olarak, bir piramit tarafından temsil edilen elmas şeklindeki belgeler ve ardından daha düşük seviyelerde azalan ifade sayıları (Şekil 4.3c), tutarsız bir ayrıntı gösterimi seviyesine işaret etmektedir (Hammer et al. 1998).

Bölüm 5'te tanıtilen NASA ARM aracı, belirli bir SRS belgesi için metin yapısını uygun bir biçimde hesaplar.

GEREKİNİM BİÇİMİ

Her gereksinim, dil kullanımında açık, özlü ve tutarlı bir biçimde olmalıdır. Tutarlı bir dil kullanmak, gereksinim belgelerinin kullanıcılarına yardımcı olur (Hull ve ark. 2011) ve SRS belgesinin yazılım araçları tarafından analizini çok daha kolay hale getirir.

Basitleştirilmiş standart gereksinim formu şöyledir:

[İsim cümlesi] shall (not) [fiil cümlesi]

[İsim cümlesi] gereksinimin ana konusunu açıkladığı durumlarda, shall (veya olmayacak) ifadesi gerekli veya yasaklanmış bir davranışı gösterir ve [fiil ifadesi] gereksinimin eylemlerini gösterir.

[İsim cümlesi] gereksinimin ana konusunu açıklarsa, shall ifadesi gerekli veya yasaklanmış bir davranışı ve [fiil ifadesi] gereksinimin eylemlerini gösterir.

Bagaj taşıma sistemi için şu iki gereksinimi göz önünde bulundurun:

-Sistem etiketlenmemiş bagajı reddedecektir

- Sistem hasarlı bagajları reddetmeyecektir

"Bagaj", "reddetme", "etiketli" ve "hasar" terimlerinin SRS belgesinde bir yerde tanımlandığını varsayıyoruz.

Gereksinimlerin standart olmayan biçimde yazılması nadir değildir. Örneğin, gösterilen ilk gereksinim şu şekilde yeniden yazılabilir:

-Etiketlenmemiş bagajlar sistem tarafından reddedilecektir

Bu gereksinimin her iki sürümü de eşdeğerdur, ancak standart formda gereksinimleri temsil etmek, zaten belirtilen nedenlerden dolayı arzu edilir.

Gereksinimlerde makaleyi atlamak da nadir değildir, bu nedenle iki örnek gereksinim şu olacaktır:

-Sistem etiketsiz bagajı reddedecektir.

-Sistem hasarlı bagajları reddetmeyecektir.

Gereksinimlerin bu formülasyonunda yanlış bir şey yoktur.

Genellikle hiyerarşik numaralandırma ile başvuru kolaylığı için gereksinimleri bir şekilde tanımlamak önemlidir. Bu nedenle, bir gereksinim için standart form aşağıdakilere uzanır:

-[tanımlayıcı] [İsim cümlesi] [fiil cümlesi]

Önceki iki gereksinime geri dönmek ve bazı tanımlayıcı numaraları icat etmek için aşağıdaki iki örneğimiz vardır:

1.2.1 Sistem etiketsiz bagajı reddedecektir.

1.2.2 Sistem hasarlı bagajları reddetmeyecektir.

Son olarak, mümkün olduğunda işlevsel gereksinimler için performansa ölçülebilir kısıtlamalar getirilmesi ve standart gereksinim formunun zenginleştirilmesi arzu edilir.

-[tanımlayıcı] [İsim tümcecisi] [fiil tümcecisi] [kısıtlama tümcecisi] olacaktır (olmayacaktır)

Yazılım ve Sistemler için Gereksinim Mühendisliği

Özellik: Bir tarayıcı ünitesinin bir bagaj parçasını taraması için gereken ortalama süre.

Ölçek: Saniye (tür: oran ölçüğü).

Prosedür: Bir paketi yasak içerikler için taramak için gereken süreyi ölçün: çeşitli türlerde ortalama 1.000'den fazla parça alın.

Planlanan değer: Referans değerden %50 daha az.

Kabul edilebilir en düşük değer: Referans değerden %30 daha az.

Referans değeri: Bir bagaj parçasını taramak için rakip veya benzer ürünlerin ihtiyaç duyduğu ortalama süre.

Şekil 4.4 Örnek gereksinimler, bagaj taşıma sistemi için ölçümleri ilişkilendirir.

Ölçülebilir bir hedef içeren bir gereksinime örnek olarak, bagaj taşıma sistemine geri dönelim. Gereksinim numarası 1.2.1 için, aşağıdaki gereksinimi sağlamak üzere gerçekçi bir kısıtlama dahil edilebilir:

1.2.1 Sistem, etiketlenmemiş bagajı tanımlamadan sonraki 5 saniye içinde reddedecektir.

Bir gereksinimin ölçülebilir hedefleri üzerinde farklılıklar ve toleranslar olabilir. Örneğin, bagaj taşıma sisteminde varsayımsal bir gereklilik 3.4.2'yi göz önünde bulundurun: 3.4.2 Her bagaj tarayıcı ünitesi dakikada ortalama 10 parça bagaj işleyecektir.

Bu gereksinimdeki bu hedef, Şekil 4.4'te (Glinz 2008) gösterilen biçimde belirli kısıtlama öznitelikleriyle artırılabilir.

"Nihai" bireysel gereksinim standart formunun, örneğin Hull ve ark. (2011) ve IEEE 29148 tarafından verilen diğer varyasyonları vardır. Ancak bu varyasyonların daha önce tanıtılan biçime indirgenebilir olduğu gösterilebilir.

Zorunlulukların Kullanımı

"Shall", gereksinim spesifikasyonlarında sıklıkla kullanılan bir komut kelimesi veya zorunluluktur. Sık kullanılan diğer komut kelimeleri ve cümleleri arasında "olmalı", "olmalı", "olacak", "sorumlu" bulunur ve daha fazlası vardır. Wilson ve ark. (1997), Tablo 4.2'de özetlenen bu zorunluluklardaki ince farklılıkları tanımlamaktadır.

Gereksinimler zorunlu olduğunda zorunluluk olarak "shall" kelimesinin kullanılması ve isteğe bağlı gereksinimler için olması gerektiği gibi zayıf zorunlulukların kullanılmasının ihmal edilmesi iyi bir uygulama olarak kabul edilir. Gereksinimler önem açısından sıralandığında, olması gerektiği gibi, standart formda düşük rütbeli bir gereklilik, esasen, isteğe bağlı bir gereklilik olarak kabul edilebilecek bir gerekliliktir.

Gereksinimler yazılırken negatif formdan daha fazla genellikle olumlu biçimin kullanılması tercih edilir. Yani, gereksinim "Yapılmamalı" ifadeleri yerine ifadeler "Yapılmalı" ifadeler kullanılarak yazılmalıdır.

TABLO

IEEE 29148, "Yapılmamalı" ifadelerini tamamen yazılmamasını tavsiye eder. Ancak bu mümkündür ve bazen kullanılmasına gerek duyulabilir. Özellikle hatalarla ilgili gereklilikler için istisnalar olacaktır. Aşağıdaki örneği düşünün.

ÖRNEK:

- Sistem sadece yetkili kullanıcıların erişimine izin vermelidir.

Bu ifade doğru yeniden yazılabilir.

- Sistem, yetkisiz kullanıcıların erişimine izin vermeyecektir.

Bu durumda, birinci formun anlaşılması daha kolaydır çünkü ikinci cümle ilk cümlenin negatiftir. Ancak, gereksinimin karşılandığını kanıtlamak için test senaryoları tasarlama sorununu düşünün. İlk cümlede, test senaryoları yetkili ve yetkisiz kullanıcıların alt kümelerini içermelidir. Ardından, hem yetkili hem de yetkisiz kullanıcılar için kapsamlı test mi, eşdeğerlik sınıfı testi mi yoksa ikili test vb. mi kullanacağımıza karar vermemiz gerekiyor. Tersine, ikinci cümlede, gereksinimin karşılandığını göstermek için yalnızca bir veya daha fazla yetkisiz kullanıcıyı içeren bir dizi test senaryosunun oluşturulması gerekir. Yetkisiz

kullanıcılar evreninin bir alt kümesini nasıl seçeceğimize hala karar vermemiz gerekiyor, ancak gereksinimin ikinci biçimi, testi önemli ölçüde basitleştiriyor.

Bazı durumlarda, şartın "yapılmalı" formu yerine "yapılmamalı" formunu kullanmanın paydaşlar (özellikle avukatlar) üzerinde psikolojik bir etkisi olabilir. Bu örneği düşünün:

■ Sistem insanlara zarar vermemeli.

Bu gereksinim, gereksinimden daha basit görünüyor

■ Sistem insanların zarar görmesine sebep olmamalı.

Her durumda, mümkün olduğunda ve açıklıktan ödün verilmediği durumlarda, gerekli ifadeleri kullanarak gereksinimler yazmalısınız. Ancak, "yapılmayacak" gereksinimleri bazen "yapılacak" formlarına tercih edilebilir.

Gereksinimlerde Belirsizliği Önleme

Bir spesifikasyon belgesindeki tüm gereksinimler kesin olmalıdır. Bu, ölçülebilir miktarlar kullanmak ve sayısız, biraz, yaklaşık, çok büyük, küçük, mikroskobik vb. gibi belirsiz deyişimlerden kaçınmaktır. Bu belirsiz kelimelerin ölçümlerle değiştirilmesi gerekiyor.

4.3.1 Sistem tarafından kaybolan bagaj sayısı, girilen bagajların %0,001'inden az olmalıdır. Bu gereksinim, gereksinimden daha basit görünüyor Bu gereksinim artık ölçülebilir ve dolayısıyla daha kesindir. Örneğin, bagaj taşıma sistemi için aşağıdaki gereksinimi göz önünde bulundurun: Sistem insanlara zarar vermeyecektir. hem yetkili hem de yetkisiz kullanıcılar için kapsamlı testler veya eşdeğerlik sınıfı testleri veya ikili testler vb. Tersine, ikinci biçimde, gereksinimin karşılandığını göstermek için yalnızca bir veya daha fazla yetkisiz kullanıcıyı içeren bir dizi test senaryosunun oluşturulması gerekir. Yetkisiz kullanıcılar evreninin bir alt kümesini nasıl seçeceğimize hala karar vermemiz gerekiyor, ancak gereksinimin ikinci biçimi, testi önemli ölçüde basitleştiriyor.

4.3.1 Sistem tarafından kaybedilen bagaj sayısı minimum olacaktır. Her durumda, mümkün olduğunda ve açıklıktan ödün verilmediğinde, gerekli ifadeleri kullanarak gereksinimleri yazmalısınız. Ancak "yapılmayacak" gereksinimleri, bazen "yapılacak" eşdeğer biçimlerine tercih edilebilir. Kesin olmayan kelimelerin daha fazla örneği "birkaç", "bazıları", "çok" ve "çoğu"dur. Bir gereksinim olarak, bu kelimeler kesirler veya aralıklar ile değiştirilmelidir. Örneğin, 1/100, 1/20, 1/2 ve 9/10 kesirler, sırasıyla yukarıda bahsedilen sözcüklerin yerini alabilir. Uygun aralıklar sırasıyla "<10", "10 ile 20 arası" "21 ile 50 arası" "50 ile 99 arası" olmalıdır Bulanık sözcüklerin kullanıldığı diğer belirsizlik biçimleri, gereksinimlerde belirsizliğe yol açabilir. Örneğin, bir gereksinimde "sıklıkla" olacak bir şeyin belirtilmesi bir aralık veya yaklaşık bir oran ile değiştirilmelidir. Örneğin, "sıklıkla" yerine "zamanın %75'inden fazlasını" kullanın. Sabit bir sayı veya aralık sağlanması, bilgilerin bulanık bir sayıdan (veya aralıktan) daha ince bir aralık veya tam sayıya gelecekte iyileştirilmesi için bir referans noktası sağlar. Gereksinimlerde diğer riskten korunma kelimelerinden de kaçınılmalıdır. Mümkün olduğunda bir gereksinim bildiriminde bir olasılık veya aralık ile değiştirilmesi gereken bu kısa riskten korunma sözcükleri listesini göz önünde bulundurun:

Muhtemelen

Neredeyse

Belki

Olabilir

Riskten korunma kelimelerini her zaman hassasiyetle deęiřtirezemezsiniz. Bazen sadece bir tahmin mmkndr. rneęin, nceden istatistikleriniz olmayan ve hatta tahmine dayalı istatistikler oluřturmak iin bir olasılık modeli bile olmayan bir olayın olasılıęına iliřkin bir tartiřmayı dřnn. Bu, bazı kabul kriterlerini kabul etmek iin paydařların mzakere olasılıklarının gerekli olacaęı bir durum olacaktır

Gereksinimlerde Belge Boyutu

Gereksinim sayısı, sayfa uzunluęu ve dięer ller aısından gereksinim belirtimi belge boyutunun istatistiklerini kapsayan az sayıda alıřma vardır. bir aralık veya yaklařık bir oran ile deęiřtirilmelidir. rneęin, "sıklıkla" yerine "zamanın %75'inden fazlasını" kullanın. Sabit bir sayı veya aralık saęlanması, bilgilerin bulanık bir sayıdan (veya aralıktan) daha ince bir aralık veya tam sayıya gelecekte iyileřtirilmesi iin bir referans noktası saęlar. Gereksinimlerde dięer riskten korunma kelimelerinden de kaınılmalıdır. Mmkn olduęunda bir gereksinim bildiriminde bir olasılık veya aralık ile deęiřtirilmesi gereken bu kısa riskten korunma szckleri listesini gz nnde bulundurun: Yine de řu soru sıklıkla sorulur: SRS belgesi ne kadar uzun olmalıdır? Yazılım gereksinimleri mhendislięi uygulamalarıyla ilgili yazılım uzmanlarıyla yapılan bir ankette, katılımcılara SRS belge uzunluęu soruldu (Neill ve Laplante 2003). Ankete katılanlar, belgelerin yaklařık %40'ının 25 sayfadan az olduęunu ve yaklařık %30'unun 25 ila 50 sayfa arasında olduęunu bildirdi. SRS belgelerinin kabaca %25'inin 51 ila 100 sayfa arasında, %17'sinin 101 ila 250 sayfa arasında olduęu ve bakiyenin 251 veya daha fazla sayfa uzunluęunda olduęu bildirildi.

Yukarıda belirtilen sayılar, 1990'larda 2000'lerin bařlarına kadar geliřtirilen 56 NASA gereksinimleri spesifikasyon belgesinin bir alıřmasında bulunanlarla tutarlıdır (Wilson ve dięerleri 1997). İncelenen projeler iin, SRS belge boyutları, medyan 2.200 ve ortalama 4.700 metin satırı olmak zere yaklařık 140 ila 28.000 metin satırı arasında deęiřiyordu. Sayfa bařına 60 satırlık bir metin olduęu varsayıldıęında, bu rakamlar 2 ila 470 sayfa aralıęına, ortalama 37 sayfaya ve ortalama 78 sayfaya dnřmektedir.

en Aıka, endstriyel norm, nispeten kısa gereksinimler spesifikasyon belgeleri iindir. Her bir SRS belgesinin uzunluk dıřındaki niteliklere gre deęerlendirilmesi gerektięinden, bu bulgunun iyi mi yoksa kt m olduęu konusunda bir yargıya varılamaz.

Davranıřsal zellikler

Bazı durumlarda, gereksinimler mevcut olmadıęında, eksik olduęunda, gncel olmadıęında veya yanlıř olduęunda gereksinim mhendisinden mevcut bir sistem iin gereksinimleri tersine evirmesi istenebilir. Test spesifikasyonlarının oluřturulması amacıyla aık kaynaklı yazılım (bir lisans kořulları altında kullanımı ve/veya yeniden daęıtımı cretsiz olan yazılım) iin gereksinimlerin oluřturulması da gerekli olabilir. Bu durumlarda, davranıřsal belirtim olan bir SRS biiminin oluřturulması gerekir.

Davranıřsal belirtim, gereksinim belirtimiyle tm ynleriyle aynıdır, ancak ilki, mhendislerin, kullanıcıların sistemin ne yapmayı amaladıęına dair en iyi anlayıřını temsil ederken, ikincisi, kullanıcıların sistemin ne yapması gerektięine iliřkin en iyi anlayıřını temsil eder. Davranıřsal belirtim, ek bir ıkarım katmanına sahiptir ve bu nedenle, bir gereksinim belirtiminden bile daha az eksiksiz olması beklenebilir.

Neyse ki, davranışsal belirtimi oluşturmaya yönelik bir yaklaşım var.

Teknik, sistemin amacını açıklayabilecek mümkün olduğunca çok sayıda eserden oluşan bir koleksiyon oluşturmayı içerir. Ardından, bu eserler, sistemin amaçlanan davranışını en iyi anlaşıldığı şekilde yeniden yapılandırmak için kullanılır. Yukarıdaki açıklama, Elcock ve Laplante (2006) tarafından hazırlanan orijinal makaleden uyarlanmıştır

Davranışsal bir belirtim türetmek için kullanılacak eserler şunları içerir (ancak bunlarla sınırlı olmamalıdır

Güncel olmayan, eksik veya yanlış olduğu bilinen kullanım kılavuzları ve yardım bilgileri (programı çalıştırırken mevcut) ,Uygulama forumları, İncelenen Yazılımın Sürümleri Bu yapılardan bazılarının müşteri dosyaları, e-postalar, açık kaynak topluluk havuzları veya arşivler gibi çeşitli kaynaklardan temizlenmesi gerekebilir. Spesifikasyonun oluşturulmasında bu eserlerin nasıl kullanıldığına dair kısa bir açıklama aşağıda verilmiştir.

Kullanıcı kılavuzlarından başlayarak, kullanıcı girdisine yanıt olarak bir uygulamanın davranışı hakkındaki ifadeler, doğrudan davranış belirtimi ile ilgili olabilir. Yazılımın kullanıcı uyarılarına tepkisini açıklamak onların varlığıyla ilgili olduğundan, kullanıcı kılavuzları bu amaç için özellikle uygundur. Destek web siteleri ve uygulama yardım menüleri gibi yardım bilgileri, davranışsal gereksinimleri tanımlamak için doğrudan kullanılabilen veya özetlenen bilgiler açısından da zengindir

Ardından, sürüm notlarına başvurulabilir. Sürüm notları, belirli bir sürümde hangi özelliklerin uygulandığını açıklamaya odaklanma eğiliminde olduklarından, test senaryoları geliştirme sürecine genellikle sınırlı yarar sağlar. Genellikle açıklama yoktur. desteklenen özelliklerin nasıl çalışması gerektiğine ilişkin Ancak, sürüm notları, bir uygulama kısmen uygulanan veya gelecekteki uygulamadan kaldırılan özellikler için beklendiği gibi yanıt vermediğinde ortaya çıkan çakışmayı çözmeye özellikle önemlidir.

Hata raporları, genellikle kullanıcılar tarafından yazıldığından, yazılımın sorunlu alanlarını belirlediklerinden ve genellikle bir geliştiricinin belirli davranışa yönelik niyetini açıklığa kavuşturduklarından davranışsal yanıtları çıkarmak için harika bir kaynak olabilir.

En azından açık kaynaklı sistemler için kusur raporları, Bugzilla gibi açık depolarda kolayca bulunur. Bazı durumlarda hata raporlarının davranışsal yanıtları çıkarmak için yararlı olamayacak kadar çok uygulama ayrıntısı içerdiği doğru olsa da, davranış ayrıntılarından tahmin edilemezse bunlar atılabilir.

Birçok yönden, destek isteklerinin içeriği, beklenmeyen davranışları belirlemeleri bakımından hata raporlarına benzer. Bununla birlikte, hata raporlarının aksine, destek talepleri bazen tam olarak uygulanmayan özellikleri belirlemede ve ayrıca kullanıcıların beklentilerini aydınlatan bilgiler sağlamada yardımcı olabilir. Her ikisini de araştırmak önemlidir, çünkü bahsedilen içgörüler sağlamanın yanı sıra, beklenmeyen davranışların rasyonelleştirilmesine de yardımcı olabilirler.

Birçok açık kaynaklı proje ve bazı kapalı kaynaklı projeler için projeye ilişkili Web tabanlı forumlar vardır. Bu forumlarda, çeşitli miktarlarda davranışsal bilgi çıkarılabilir. Yararsızdan alakalıya kadar, açık tartışma gönderilerinin dikkatli bir şekilde filtrelenmesi ve yalnızca diğer geliştirme eserleri eksik olduğunda uygulanması gerekir. Diğer eserlerde olduğu gibi, bu ilanlar da davranış netleştirmek için kullanılabilir.

Son olarak, başka herhangi bir yapaylığın yokluğunda, test edilen yazılımın kendisi yapısal (cam kutu) test senaryolarının geliştirilmesi için bir girdi olabilir. Bu yaklaşımın gerekli

olduğunu varsayarsak, gerçek şu ki, test senaryolarının tanımlanmasında büyük ölçüde hakim olacak olan, gerçekten de test edenin doğru davranışı karakterize etmesi olacaktır.

Keşif süreci sona erdiğinde, davranışsal belirim yazılabilir. Davranış spesifikasyonunun formatı, gereksinim spesifikasyonunun formatı ile aynıdır (Elcock ve Laplante 2006) ve tüm IEEE 29148 kuralları uygulanmalıdır.

En iyi Uygulamalar ve Öneriler

Etkili gereksinim spesifikasyonları yazmak, belirttiğimiz birçok zorluk nedeniyle önemsiz sistemler için bile çok zor olabilir. Zayıf SRS belgeleri yazmanın daha yaygın tehlikelerinden bazıları şunlardır:

- Operasyonel ve tanımlayıcı özelliklerin karıştırılması
- Düşük düzeyli donanım işlevselliği ile üst düzey sistemler ve yazılım işlevlerini aynı işlevsel düzeyde birleştirmek
- Zamanlama bilgilerinin ihmal edilmesi

Diğer kötü uygulamalar, doğrulanabilecek bir dilin kullanılmamasından kaynaklanır. Örneğin, bu gereksinimler grubunu göz önünde bulundurun:

- Sistem tamamen güvenilir olacaktır.
- Sistem modüler olacaktır.
- Sistem hızlı olacaktır.
- Hatalar %99'dan az olacaktır.

Bunların nesi var? Sorun şu ki, tamamen belirsiz ve ölçülemezler ve bu nedenle memnuniyetleri kanıtlanamıyor. Örneğin, "tamamen güvenilir" ne anlama geliyor? Herhangi bir keyfi kişi, belirli bir sistem için güvenilirlik için farklı bir anlama sahip olacaktır. Modülerliğin (yazılımda) belirli bir anlamı vardır, ancak nasıl ölçülebilir? "hızlı" ne demek Tren kadar hızlı mı? Bir mermiden daha hızlı? Bu gereksinim çok belirsizdir. Ve son olarak, "hatalar %99'dan az olacak" bir dava reçetesidir. neyin %99'u? Hangi zaman diliminde?

Yukarıdaki gereksinimler için geliştirilmiş sürüm:

- Birinci seviye eylemlerin tümü için yanıt süreleri 100 ms'den az olacaktır.
- Her modülün döngüsel karmaşıklığı 10 ila 40 aralığında olmalıdır.
- İşlemlerin %95'i 1 saniyeden daha kısa sürede işlenecektir.
- İlk arızadan önceki ortalama süre 100 saat sürekli çalışma olacaktır.

Ancak bu set bile kusurludur, çünkü bazı önemli ayrıntılar eksik olabilir - SRS belgesinin geri kalanının bağlamı dışında neyin eksik olduğunu gerçekten bilemeyiz.

Spesifikasyon belgelerinin yazılması için bazı son öneriler şunlardır:

- Standart bir format icat edin ve kullanın ve tüm gereksinimler için kullanın.

- Dili tutarlı bir şekilde kullanın.
- Zorunlu gereksinimler için "yapmalı" kullanın.
- Arzu edilen gereksinimler için "gerekir" ifadesini kullanın.
- Gereksinimin önemli kısımlarını belirlemek için metin vurgulamayı kullanın
- Garanti edilmedikçe teknik dil kullanmaktan kaçının

Dick (2010), gereksinimler belgesini yazarken aşağıdakilerden kaçınılmasını da önerir:

- Saçma sapan ifadeler
- "Gerekirse" gibi riskten korunma maddeleri
- "ve", "veya" ve "ama" gibi bağlaçlar
- tek bir ifadede birden fazla gereksinime yol açabilir
- "Belki" ve "bazen" gibi spekülâtif terimler
- "Kullanıcı dostu" ve "genel olarak" gibi muğlak terimler kullanılmayacağından bahseder.