

Bölüm 8

Araç Desteği Gereksinim Mühendisliği

Tanıtım

NASA ARM aracılığıyla otomatikleştirilmiş gereksinim analizini ve ilgili gereksinim doğrulama ve doğrulama araçlarını Bölüm 5'te zaten tartışmıştı. Kelime işleme, veritabanı yöneticileri, elektronik tablolar, içerik analizörleri, kavram haritalama programları, otomatik gereksinim denetleyicileri vb. gereksinim mühendisi. Tablo 8.1, gereksinim yönetimi için kullanılan geleneksel ofis araçlarının özelliklerini, zengin özelliklere sahip bir gereksinim yönetim aracıyla karşılaştırır.

Ancak en önemli gereksinim araçları, yüksek düzeyde entegre işlevsellik sağlayan bu tür ticari veya açık kaynaklı paketlerdir. Bu gereksinim yönetimi araçlarının başlıca işlevi, gereksinim mühendisliği yaşam döngüsü boyunca gereksinimler, gereksinimler, kullanım senaryoları, senaryolar ve kullanıcı yükleri için kaynaklar gibi tüm "tipik" gereksinim mühendisliği nesnelerini temsil etmek ve düzenlemektir. Kullanıcı tanımlı varlıklar için de destek genellikle sağlanır. Bu tür ticari gereksinim mühendisliği araçları için diğer tipik işlevler şunları içerir:

- Çoklu kullanıcı desteği ve sürüm kontrolü

- Çevrimiçi işbirliği desteği

- Özelleştirilebilir kullanıcı arayüzleri

- Standart şablonlar için yerleşik destek (IEEE 29148 gibi)

- Doğrulama ve doğrulama araçları

- Programlanabilir bir arayüz aracılığıyla özelleştirilebilir işlevsellik

- İzlenebilirlik desteği

- Kullanıcı tanımlı sözlük desteği (Heindl et al. 2006)

Tablo 8.1 Gereksinimler Deposu Metrik Yetenekleri Hammer ve ark. (1998)

	Kelime İşlemci Elektronik Tablosu		ilişki Veri tabanı	Gereksinimler Yönetmek Alet
Belge Boyutu	Evet	Numara	Numara	Değişilmeden biçimlendirilmiş durum
Dinamik Değişiklikler Mesai	Mümkün ile karmaşık değişiklik izleme etkin	Numara	Numara	Evet
Yayın Boyutu	Evet	Evet	Evet	Evet
Gereklilik Genişletme Profili	Numara	Numara	Evet	Evet
Gereksinimler Doğrulama	Numara	Numara	Mümkün	Mümkün
Gereksinimler oyunaklık	Evet	Evet	Evet	Evet
Test kapsamı	Numara	Karmaşık denklem mantığıyla mümkün	Evet	Evet
Test Aralığı		Karmaşık denklem mantığıyla mümkün	Evet	Evet
Test Türleri	Evet	Evet	Evet	Evet

Doğrulama ve doğrulama özellikleri, herhangi bir otomatikleştirilmiş gereksinim mühendisliği aracının önemli bir bileşenidir. Gerçekten de, daha karmaşık ticari gereksinim mühendisliği araçları, diğer gereksinimleri kontrol etme, izleme ve araştırma özellikleri sağlar. Bunlar Tablo 8.2'de gösterilmiştir.

Bu özellikler özellikle önemlidir çünkü zamanla artefaktların doğru bir şekilde izlenmesini sağlarlar. İzlenebilirlik, SRS belgesinin önemli bir özelliğidir ve daha fazla tartışmayı gerektirir.

Tablo 8.2 Otomatikleştirilmiş Gereksinimler Mühendislik Aracı Özellikleri (Heindl 2006)

Araç Özelliği	Tanım
İçin iş akışının tanımı Gereksinimler	Gereksinimler için bir iş akışı (durumlar, roller, durum geçişleri) yapılandırılabilir.
İşlerin çift yönlü olduğu otomatik olarak oluşturulması	Kullanıcı artefakt arasında bir ilişki olduğu durumda A ve artefakt B, otomatik olarak B'den A'ya geriye doğru bir ilişki oluşturur.
Kullanıcıya özel izleme türlerinin tanımı	Yetkili bir kullanıcı, izleme türleri tanımlayabilir ve adlar atayabilir.
Şüpheli izler	Bir gereksinim değişikliğinde araç, izlemeleri kontrol etmek ve güncellemek için bu gereksinimle ilgili tüm izleri otomatik olarak vurgular.
Uzun süreli arşivleme işlevi	Araçtaki tüm veriler, gerektiğinde araç olmadan erişilebilen bir formatta arşivlenebilir.

İzlenebilirlik Desteği

İzlenebilirlik, gereksinimler, bunların kaynakları ve çok sayıda diğer eserler arasındaki ilişkilerle ilgilidir. Üç tür izlenebilirlik tanımlıyoruz—gereksinimleri bu gereksinimleri öneren paydaşlara bağlayan kaynak izlenebilirliği; bağımlı gereksinimler arasında bağlantı kuran gereksinim izlenebilirliği; ve gereksinimlerden tasarıma bağlantılar sağlayan tasarım izlenebilirliği.

Örneğin, gereksinimlerden kaynak koduna, kaynak koddan test senaryolarına ve test senaryolarından gereksinimlere kadar bağlanabilirlik türleri vardır. Bu tür bir gidiş-dönüş izlenebilirliği, özellikle kritik görev sistemlerinde önemlidir. Örneğin, ABD, Kanada ve diğer ülkelerdeki federal havacılık düzenleme kurumları tarafından kullanılan Standart DO-178C, Havadaki Sistemlerde Yazılım Hususları ve Ekipman Sertifikasyonu, eser izleme yeteneği için kurallar içerir. DO-178C, tüm düşük seviyeli gereksinimler ile bunların ana yüksek seviyeli gereksinimleri arasında izlenebilirlik gerektirir. Kaynak kodu öğeleri, gereksinimler ve test senaryoları arasında da bağlantılar zorunludur. Tüm yazılım bileşenleri bir gereksinime bağlı olmalıdır, yani tüm öğelerin gerekli bir amacı olmalıdır (yani altın kaplama yok). Daha sonra bu kurallara uyulmasını sağlamak için sertifikasyon faaliyetleri yürütülür (RTCA 2011). Ancak amaçlarımız için, yalnızca gereksinim belirtimi belgesinde (veya yardımcı belgelerde) bulunan izlenebilirlik yapılarıyla ilgileniyoruz.

202 Yazılım ve Sistemler için Gereksinim Mühendisliği

SRS belgesinde izlenebilirlik, gereksinimler ile bunların kaynakları, paydaşları, standartları, yönetmelikleri vb. arasındaki karşılıklı ilişkiyi odaklanır.

Bu amaçlar için, izlenebilir eserler şunları içerir:

Gereksinim

Bir gereklilikle ilişkili paydaş(lar)

Gerekliliği zorunlu kılan standart, yönetmelik veya yasa

Gerekçe (gereksinim için)

Anahtar kelimeler (arama için)

Bunların ve diğer eserlerin çeşitli kombinasyonları, birkaç kısa süre içinde sunulacak olan birçok izlenebilirlik matrisi formatına yol açar. Yu ve ark. (2011) ayrıca gereksinim öğeleri ve analiz modeli öğeleri arasındaki bağlantıları koruyan izlenebilirlik matrisleri oluşturmayı önerir. Gereksinim izleme yaklaşımları ve uygulamaları hakkında kapsamlı bir tartışma Lee ve ark. (2011).

Gereksinimler Bağlantılı İzlenebilirlik Matrisi

Bir gereksinimin, başka bir gereksinime "kullanımlar" veya "gösterir" ilişkisi olarak açık bir bağ vuruda bulunması alışılmadık bir durum değildir. Aşağıdaki gereksinimi göz önünde bulundurun:

2.1.1 Sistem, ısıtma sisteminin kontrolünü sağlamak için zaman programlama fonksiyonuna göre programlanabilir olmalıdır (ref. gereksinim 3.2.2).

bu, bir "gösterir" ilişkisini gösterir. "Kullanımlar" ilişkisi aşağıdaki gereksinimde gösterilmektedir:

2.1.1 Sistem, gereksinim 3.2.2'de gösterilen zaman çizelgesine ve gereksinim 4.1'de açıklanan güvenlik özelliklerine göre ısıtma sisteminin kontrolünü sağlamalıdır.

"Kullanımlar" ve "gösterdiği" arasındaki temel ayrım, "kullanımların" daha güçlü ve doğrudan bir bağlantıyı temsil etmesidir.

İzlenebilirliğin birincil eseri, gereksinim izlenebilirlik matrisidir.

Gereksinim izlenebilirlik matrisi, SRS belgesinde tablo biçiminde, bağlantısız bir izlenebilirlik belgesinde görünür veya değil işlevlik, görünümleme ve çıktı için bir araç içinde dahili olarak temsil edilebilir. Hangi eserlerin dahil edileceğine bağlı olarak birçok izlenebilirlik matrisi varyasyonu vardır.

Bir matris formu, gereksinimler arasındaki karşılıklı ilişkiyi gösterir. Burada, gereksinim izlenebilirlik matrisindeki girişler aşağıdaki gibi tanımlanır:

$R_{ij} = R$ öğeri gereklilik gereksinim j 'ye atıfta bulunuyorsa (bilgi amaçlı olarak "belirtilen" anlamına gelir).

Rij = U , eğer gereksinim i , gereksinim j'yi kullanırsa (doğrudan "bağlıdır" anlamına gelir).

Satır = aksi takdirde boş .

Bir gereksinim baş ka bir gereksinimi hem kullandığı ında hem de referans gösterdiği ında, “U” giri ş i “R” nin yerini alır. Kendi kendine referanslar dahil edilmediğ i inden, matrisin kö ş egeni her zaman boş giri ş ler içerir. $R_{ij} = U$ veya $R_{ij} = R$ ise, R_{ji} ’nin boş olmasını bekleyeceğ imiz ş ekilde, gereksinimlerde dairesel referanslar beklemeyiz. Aslında, bir gereksinim mü hendisliğ i aracındaki otomatik doğ rulama özelliğ i, bu tü r döngü sel referansları iş aretlemelidir.

Tipik bir gereksinim izlenebilirlik matrisinin formatı Tablo 8.3'te gösterilmektedir.

Örnek giriş leri olan varsayımsal bir sistem için kısmi izlenebilirlik matrisi Tablo 8.4'te gösterilmiş tir.

R genellikle seyrek olduğu undan , yalnızca boş olmayan satırları ve sütunları listelemek uygundur. Örneğin, ekte bulunan akıllı ev için SRS belgesi için, gereksinimlerden açıkça türetilen izlenebilirlik matrisi Tablo 8.5'te gösterilmektedir.

Bu t  rden bir seyrek izlenebilirlik matrisi, gereksinimler arasında d  Ő   k d  zeyde a ık bir baĒ lantı olduĒ unu g sterir. Gereksinimler belgesinde d  Ő   k d  zeyde bir baĒ lantı arzu edilir - gereksinimler ne kadar baĒ lantılıysa, bir gereksinimde o kadar fazla deĒ iŒ iklik diĒ erlerine yayılır. Aslında, a ık gereksinimler baĒ lantıları,

Tablo 8.3 Gereksinim Spesifikasyonu için İzlenebilirlik Matrisi Formatı (R)

[illegible]

Tablo 8.4 Hayali Bir Sistem Spesifikasyonu için Kısmi İmlenebilirlik Matrisi

Gereksinim Kimliği	1.1	1.1.1	1.1.1.1	1.2	1.2.1	1.2.2	2	2.1	2.1.1	3					
1													RR		
1.1															
1.1.1	R										R				
1.1.2								R							
1.2															
1.2.1									R						
1.2.2													R		
2												R			
2.1															
2.1.1															R
3							R								

Tablo 8.5 Ekte Gösterilen Akıllı Ev Gereksinimleri için İmlenebilirlik Matrisi Spesifikasyonu

Gereksinim Kimliği	5.11	9.11
9.1.1	R	
9.10.7		R

endişelerin ayrılması ilkesi, yazılım mühendisliğinin temel bir ilkesidir.

Bu nedenle, gereksinimleri yalnızca kesinlikle gerekli olduğu anda bağlayın.

Genel olarak, her gereksinimin birden fazla test senaryosu tarafından test edilmesini isteriz. Aynı zamanda, her bir test senaryosunun birden fazla aşırı yerine getirmesini istiyoruz. "Test süresi" metrikleri, test planını karakterize etmek ve yetersiz veya aşırı testi belirlemek için kullanılır:

Test başına gereksinimler
İhtiyaca göre testler

Bu metrikler için uygun istatistikleri belirlemek için araştırmalar halen devam etmektedir. Ancak en azından tutarsızlıkları ve tek biçimli olmayan test kapsamını aramak için bu ölçümleri kullanabilirsiniz. Elbette, testin kapsamılığı ile testin süresi ve maliyeti arasında her zaman bir denge vardır. Ancak test bu kitabın konusu değil.

Gereksinimler Kaynak İzlenebilirlik Matrisi

Yine bir başka tür izlenebilirlik matrisi, gereksinimleri kaynaklarına bağlar. Doğrudan kullanıcılardan gelenlerin yanı sıra, birçok gereksinim hükümet düzenlemelerinden ve standartlardan türetilmiştir. Gereksinimleri bu kaynaklara bağlamak, kaynaklar değiştiğinde çok yardımcı olabilir. Tablo 8.6, böyle bir izlenebilirlik matrisi için tipik formatı göstermektedir.

Bu tür izlenebilirlik matrisi, özellikle işlevsel olmayan gereksinimlerin izlenmesi için kullanışlıdır. Proje yaşam döngüsü boyunca işlevsel olmayan gereksinimlerin izlenememesi önemli bir sorun olabilir (Kassab ve Ormandjieva 2014).

Gereklilikler Paydaş İzlenebilirlik Matrisi

Başka bir izlenebilirlik matrisi türü, paydaşları sundukları gereksinimlere bağlar; bunun bir örneği Tablo 8.7'de gösterilmektedir.

Tablo 8.7 ayrıca, gereksinimleri büyüklüğü de kolaylaştırarak bir sıralama işlemi içermektedir. Müzakere ve takas analizi.

Önceki bölümlerde yalnızca üç tür izlenebilirlik matrisi gösterilmiştir, ancak bunların herhangi bir varyasyonu veya kombinasyonu oluşturulabilir ve bunlar

Tablo 8.6 Gereksinimleri ve Kaynaklarını Gösteren İzlenebilirlik Matrisi

Gereklilik ID	Federal Yönetmelik #1	Federal Yönetmelik #2	<small>Batırmak, kırmak</small> Yönetmelik #1	<small>Batırmak, kırmak</small> Yönetmelik #1	Uluslararası Standart #1
3.1.1.3	X				
3.1.2.9	X	X			
3.2.1.8			X	X	
3.2.2.5			X		
3.2.2.6					X
3.3.1		X			
3.3.2		X			
3.4.1		X			
3.4.3		X			
3.4.4		X			
3.6.5.1				X	
3.6.6.4					X

Tablo 8.7 Gereksinimleri, Sıralamalarını ve Paydaş Kaynaklarını Gösteren İzlenebilirlik Matrisi

Gereklilik	Rütbe 1 (en düşük önem) – 5 (en yüksek önem)	Paydaş Kaynak D – Doktor, N – Hemşire, A – İdari Destek Personel, P – Hasta, R – Regülatör
3.1.1.1	5	R
3.1.1.2	4	R
3.1.2.1	5	DNA
3.1.2.2	3	DNA

bahsedilen diğer izlenebilirlik yapılarını dahil edin. Ticari gereksinim mühendisliği araçları, kullanıcının her türlü izlenebilirlik matrisini özelleştirmesine olanak tanır.

Gereksinim Yönetimi Araçları

Gereksinim araçları, bir projenin yaşam döngüsü boyunca gereksinim bilgilerinin yönetilmesinde son derece önemlidir. Örneğin, en yaygın olarak kullanılan gereksinim yönetimi araçlarından biri olan IBM'in DOORS'unu (dinamik nesne yönelimli gereksinimler sistemi) ele alalım. DOORS, temel olarak gereksinim yönetimi için bir nesne veritabanıdır. Yani her özellik, bir özellik tanımı, özellik grafiği ve kullanım durumu diyagramını içeren bir nesne olarak temsil edilir. Veritabanı, klasörler ve projeler oluşturularak daha da düzenlenir ve tüm modül ve nesne düzeyindeki eylemlerin geçmişi korunur. DOORS'u, UML modellerinin DOORS içine gömülmesine ve izlenmesine izin veren başlıca bir araçla entegre eden Analyst adlı bir uzantı var (Hull ve diğerleri 2011).

Özellik öznitelikleri zorunlu, isteğe bağlı, tek bağımlı ve birden çok bağımlı içerir ve kullanıcı, gereksinimlerin tamamlayıcı ve destekleyici yapısını bünyesinde barındıran bir özniteliğin gerekli mi yoksa hariç mi olduğunu belirleyebilir. Diğer nitelikler, kullanım senaryosu paketleri ve ünlü örneklerine dayanmaktadır (Eriksson ve diğerleri, 2005).

DOORS, tam izlenebilirlik ve eksik bağlantı analizi için bir projedeki tüm nesneler arasında bağlantı sağlar. Araç, C benzeri bir programlama dili aracılığıyla özelleştirilebilir. Ve ISO 12207, ISO 6592 ve diğer standartlarla uyumlu olarak gereksinimleri yapılandırmak için standart şablonlar mevcuttur (Volare 2013).

DOORS'un en son sürümünde daha birçok gereksinim analizi ve yönetimi özelliği bulunmaktadır. DOORS'un (yazım tarihi itibarıyla) DOORS Yeni Nesil (NG) adlı güncel sürümü hakkında daha fazla bilgi için lütfen Ek F, DOORS NG'ye başvurun.

Araç Değ erlendirmesi

Birçok ticari ve açık kaynaklı araç vardır ve herhangi bir aracın özelliklerini iş letmede benimsemeden önce dikkatlice değ erlendirmek önemlidir. ISO/IEC TR 24766 (2009) Bilgi Teknolojisi—Sistemler ve Yazılım Mü hendisliğı i—

Gereksinim Mü hendisliğı i Araç Yetenekleri Kılavuzu, gereksinim mü hendisliğı inin yeteneklerini değ erlendirmek için bir çerçevedir. Bu yetenekler ş u alanlarda organize edilmiş tir: ortaya çıkarma, analiz, belirtim, doğ rulama ve geçerli kılma, yönetim ve diğ er yetenekler. ISO/IEC TR 24766 çerçevesi kullanılarak ticari araçlarla ilgili çeş itli çalış malar yapılmış tır (örneğ in, Carrillo de Gea ve diğ erleri 2011, 2015; Daud ve diğ erleri 2014). Bu çalış malar genellikle alet pazarının hızla değ iş tiğ ini ve aletlerin giderek daha karmaş ık ve kullanımı zor hale geldiğ ini ortaya koymuş tur. Pahalı ticari araçların karmaş ık lığı ı daha sonra ucuz araçların ortaya çıkması için fırsatlar yaratır, ancak karmaş ık özellikler sunmaz. Ayrıca, bu çalış malar, araçların çoğ unda tutarlılık, doğ ruluk ve eksiksizlik gibi doğ rulama iş levlerinin hala eksik olduğ unu göstermiş tir.

Sud ve Arthur (2003), aş ağı ıdaki boyutları kullanarak bir dizi gereksinim yönetimi aracını değ erlendirdi:

- Gereksinim izlenebilirlik mekanizması
- Gereksinim analiz mekanizması
- Gü venlik ve eriş ilabilirlik mekanizması
- Taş ınabilirlik ve arka uç uyumluluğ u
- Konfigü rasyon yönetimi yaklaş ımı
- İ letiş im ve iş birliğı i mekanizması
- Değ iş im yönetimi desteğ i
- Çevrimiçi yayın desteğ i
- Kelime iş lemcisi uyumluluğ u gibi kullanılabilirlik özellikleri
- SRS belge formatı

Çalış malarının bulguları artık bir ş ekilde modası geçmiş olsa da, bu değ erlendirme boyutları mü hendisler tarafından benimsenmeden önce çeş itli ticari ve açık kaynak gereksinim yönetimi araçlarını karş ılaşt ırma için kullanılabilir. Örneğ in, bu boyutlar kullanılarak basit bir kontrol listesi oluş turulabilir veya Likert ölçeğ ine dayalı bir değ erlendirme yapılabilir.

Açık Kaynak Gereksinimleri Mü hendislik Araçları

Birçoğ u tam özellikli gereksinim yönetimi araçları olan yüz binlerce açık kaynak projesi vardır ve bunları satın almadan veya sıfırdan gelişt irmeye çalış madan önce araçları aramak için ilk olarak açık kaynak havuzlarına dönmeniz teş vik edilir. Gereksinim mü hendisliğı i için yardımcı programlar veya kaynaklar da vardır.

208 Yazılım ve Sistemler için Gereksinim Mühendisliği

Bu bölümde, kullanılacak bu tür iki açık kaynaklı yardımcı programa bakacağız. gereksinim mühendisi.

Özgür aklın

Zihin haritalama araçları, kullanıcının düşünceleri ve fikirleri arasındaki ilişkiyi tasvir eden ve beyin fırtınası, görev analizi, merdivenleme, izlenebilirlik, JAD, odak grupları ve diğer birçok gereksinim mühendisliği faaliyetleri için gereksinim mühendisi için yararlı olan kavram haritaları oluşturmasına olanak tanır. Kavram haritalama araçları, kavram haritalarının kolayca oluşturulmasına izin verir ve yukarıdaki tartışma, açıklayıcı amaçlar için belirli bir araca dayalıyken, gereksinim mühendisi tarafından herhangi bir kavram haritalama aracı kullanılabilir.

FreeMind, Java ile yazılmış açık kaynaklı bir zihin haritalama aracıdır. (FreeMind 2017). Araç, kullanımı kolay bir grafik kullanıcı arayüzüne sahiptir. Kavramlar, birincil bir kavrama ve bu kavrama bağlı fikirlerle dayalı hiyerarşik bir yapı içinde haritalanır. Her fikir için öncelikleri, önemli kriterleri veya tehlikeleri belirtmek için kullanılan simgeler kullanılabilir. Zihin haritasının içeriğini zenginleştirmek için başka birçok simge mevcuttur.

Gereksinim mühendisinin zihin haritalarının kullanımını göstermek için, Phil'in akıllı ev kontrol sistemi konseptine bakalım. Temel bir akıllı ev konseptiyle yola çıkarak Phil, Şekil 8.1'de gösterildiği gibi bir ebeveyn düğümünü (bir balonla temsil edilir) oluşturur.

Ardından Phil, sisteme bir özellik (güvenlik) eklemek ve bu özelliğe bir öncelik atamak için aracı kullanır. Yedi seviyeye kadar sıralama mevcut olmasına rağmen, Phil üç seviyeli basit bir sistem seçti.

1. Zorunlu
2. İsteğe bağlı
3. Sahip olmak güzel

Ortaya çıkan sıralı özellik ve güncellenmiş zihin haritası Şekil 8.2'de gösterilmektedir.

Phil daha sonra özelliğe ayrıntılar ekler. Örneğin akıllı evin güvenlik sisteminin evdeki mevcut güvenlik sistemini kullanmasını ve onunla çalışmasını istiyor.



Phil's Smart Home

Arzu ettiğini bir diğer özellik ise HVAC (ısıtma, havalandırma ve iklimlendirme) sisteminin akıllı ev sistemi ile entegre olmasıdır. Ek olarak, Phil'in evinde hassas bitkiler, evcil hayvanlar ve koleksiyon üyeleri bulunduğundan, sıcaklığın asla 100°F'yi aşmaması önemlidir, bu nedenle bu tehlike bir bomba simgesiyle işaretlenmiştir. Gözden geçirilmiş zihin haritası Şekil 8.3'te gösterilmektedir.

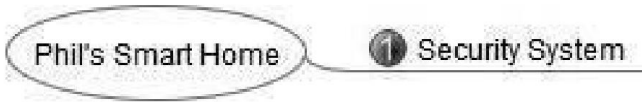
Phil, müzik yönetimi, çim bakımı ve telefon yanıtlama sistemi ekleyerek özelliklerle ilgili beyin fırtınasına devam ediyor. Bazı özellikler, uygun sembollerle işaretlenmiş önemli detayları içerir (Şekil 8.4).

Sistemin görsel anlamda gerçekleştirildiğini gören Phil, daha fazla özellik düşünebilir ve uygun önceliklendirme kararlarını verebilir. Phil, Şekil 8.5'te görüldüğü gibi zihin haritasına buna göre ekler.

Şekil 8.5'te gösterilen zihin haritası tamamlanmamıştır. Ancak bu zihin haritası, gereksinim mühendisleriyle yapılan açıklama tartışmaları sırasında kullanılabilir ve zaman içinde kolayca geliştirilebilir.

FreeMind, zihin haritasını metin içinde hiyerarşik yapıya dönüştürmenize olanak tanıyan güzel bir özellik e sahiptir. Böylece, tüm diyagramı seçip kopyalayıp bir Word işlemciye yapıştırırsanız, iç içe bir metin listesi elde edersiniz. Daha sonra, bir gereksinim belirtimi belgesinde kullanılmak üzere hiyerarşik numaralandırılmış bir liste elde etmek için sayılar ekleyebilirsiniz.

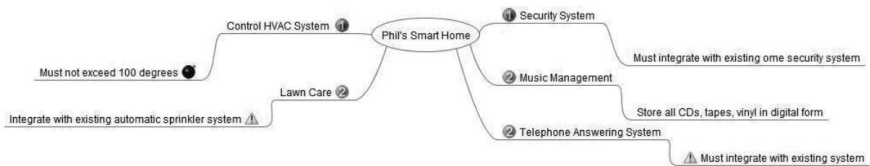
FreeMind, fikirleri yazmanıza ve bunları çok hızlı bir şekilde yeniden düzenlemenize olanak tanır ve bir gereksinim belirtimi belgesini başlatmak için harika bir araçtır. Ayrıca birçok taşınabilir cihazda çalışan FreeMindPDA adlı bir sürümü de bulunmaktadır.



Şekil 8.2 Phil'in akıllı ev sistemine ilk özelliğin eklenmesi.

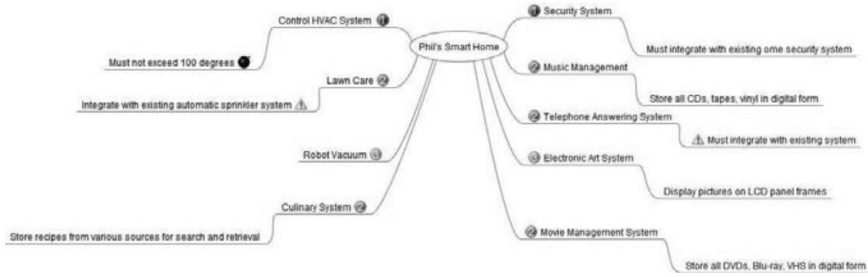


Şekil 8.3 Phil'in akıllı ev sistemine ilk özelliğe bir detay ekleme ve bir tehlike ekleme.



Şekil 8.4 Phil'in akıllı ev sistemine daha fazla özellik ekleme.

210 Yazılım ve Sistemler için Gereksinim Mühendisliği



Şekil 8.5 Phil'in bir akıllı ev sistemine ilişkin kısmi zihin haritası.

Tabii ki, FreeMind'a benzer işlevselliklere sahip başkaca açık kaynaklı ve ticari zihin haritalama araçları da var. Bu araçlardan herhangi biri, gereksinimlerin belirlenmesinde ve gereksinimlerin ortaya çıkarılmasında çok değerli olabilir.

FitNesse

FitNesse, etkileşimli test senaryoları oluşturmak için bir çerçeve sağlayan açık kaynaklı, wiki tabanlı bir yazılım işbirliği aracıdır. FitNesse başlangıçta bir test aracı olarak geliştirilmiş olsa da, bizim ilgi alanımız, örneğin çevik bir ortamda entegre gereksinimler/test spesifikasyonları için kullanmaktır.

FitNesse'nin nasıl çalıştığını kısaca açıklamadan önce, bir FitNesse test senaryosunun neye benzediğini incelemek uygun olur. Test durumu, esasen, test edilecek fonksiyonun (veya yöntemin) adından, girdi parametreleri setinden ve karşılaşılan gelen beklenen sonuç parametrelerinden ve ardından tablonun satırları boyunca bir dizi tipik test senaryosundan oluşan bir tablodur. (bkz. Tablo 8.8).

FitNesse tablosu yürtülebilir; gerçek kod geliştirildiğinde, tıklandığında birdüğmeye basıldığında, karşılaşılan gelen işlevin sonuçları hesaplanacaktır.

Akıllı ev sistemi için SRS kullanarak durumu örnekleyelim.
ek. Bu SRS'de Bölüm 9.2.2'şunları belirtir:

9.2.2 Sistem, su mevcut olduğunda, kahve çekirdeği seviyeleri yeterli olduğunda üniteye enerji verildiğinde sürece, kullanıcı tanımlı herhangi bir zamanda kahve makinesini başlatacaktır.

Tablo 8.8 Tipik UyumNesse Gereksinimi/Test Senaryosu Formatı

Adına İşlev							
Giriş 1	Giriş 2	...	Giriş n	Beklenen sonuç 1	Beklenen sonuç 2	...	Beklenen sonuç

Ön koşulların karşılandığını varsayarsak (kahve çekirdekleri ve su mevcuttur), kahve yapmak için makul bir program Tablo 8.9'da verildiği gibi olabilir.

Tablo 8.9'da gösterilen program, ev sahibinin hafta sonları geç uyumayı sevdiğini kabul etmektedir. Bu tabloyu farklı şekillerde doldurabiliriz. Artık istenen işlevselliğin dinamik zamanlama için olduğu doğrudur, böylece örneğin bir tatil haftasında ev sahibi her günün geç saatlere kadar uyuyabilir. Ancak buradaki tablo, bir cihaz programlama senaryosunu belirtmek için kullanılır. Aslında, FitNesse çerçevesi aracılığıyla uygun şekilde yapılandırılırsa, Tablo 8.9, tamamlanmış sistem için yürütülebilir bir test durumu oluşur.

Çok fazla ayrıntıya girmeden, FitNesse'nin nasıl çalıştığını kabaca burada bulabilirsiniz. FitNesse, Fit çerçevesinin üzerine inşa edilmiş bir wiki GUI'dir. Fit, test tablosuna karşılık gelen fişti kodunu (bir Java veya C# sınıfı) çalıştırır. Örneğin, Tablo 8.9'un en üst satırında Activate_Coffee_Pot, çağrılacak asıl sınıfı belirtir. FitNesse, test başlangıçsız olursa, geçerse veya bir istisna atılırsa, beklenen sonuç hücrelerini sırasıyla kırmızı, yeşil veya sarı renklendirir (Gandhi ve ark. 2005).

Tablo 8.9 Akıllı Ev Sisteminde Cezve İşlevini Etkinleştirme için FitNesse Gereksinimleri Spesifikasyonu/Test Senaryosu

Activate_Coffee_Pot		
Gün	Zaman	Çıktı
Pazartesi	7:00	coffee_pot_on_signal=yüksek
Pazartesi	8:00	coffee_pot_on_signal=düşük
Salı	7:00	coffee_pot_on_signal=yüksek
Salı	8:00	coffee_pot_on_signal=düşük
Çarşamba	7:00	coffee_pot_on_signal=yüksek
Çarşamba	8:00	coffee_pot_on_signal=düşük
Perşembe	7:00	coffee_pot_on_signal=yüksek
Perşembe	8:00	coffee_pot_on_signal=düşük
Cuma	7:00	coffee_pot_on_signal=yüksek
Cuma	8:00	coffee_pot_on_signal=düşük
Cumartesi	9:00	coffee_pot_on_signal=yüksek
Cumartesi	10:00	coffee_pot_on_signal=düşük
Pazar	10:00	coffee_pot_on_signal=yüksek
Pazar	11:00	coffee_pot_on_signal=düşük

212 Yazılım ve Sistemler için Gereksinim Mü hendisliđ i

Fitness'e iş levsellik açısından benzeyen baş ka açık kaynaklı araçlar da var. Örneğ in Salatalık, Scrum ortamlarında (cucumber.io) yaygın olarak kullanılan bir araçtır.

Gereksinimler Mü hendisliđ i Aracı En İ yi Uygulamaları

Hangi mü hendislik araç(lar)ını kullanırsanız kullanın, aracı akıllıca kullanmak ve belirli en iyi uygulamaları takip etmek uygundur. Cleland-Huang ve diğ erleri tarafından bu tü r uygulamaların mü kemmel bir seti sunulmaktadır. (2007):

Bir amaç için takip edin. Yani, hangi bağ lantıların gerçekten önemli olduđ unu belirleyin; aksi takdirde, çok sayıda yabancı bağ lantı oluş turulacaktır.

Uygun bir iz ayrıntı dü zeyi tanımlayın. Örneğ in, bağ lantılar uygun paket, sınıf veya yöntem dü zeyinde yerleş tirilmelidir.

Yerinde izlenebilirliđ i destekleyin. Öğ eler kendi yerel ortamlarında bulundukları için izlenebilirlik sağ layın.

İ yi tanımlanmış bir proje sözlü ğ ü kullanın. Paydaş larla yapılan ilk keş if toplantıları sırasında sözlü ğ ü oluş turun ve gereksinim mü hendisliđ i sü reci boyunca tutarlı bir ş ekilde kullanın.

Kalite gereksinimlerini yazın. İ zlenebilirlik için özellikle önemli olan IEEE 29148 gibi genel kabul görmü ş en iyi uygulamaları takip ettiğ inizden emin olun.

Anlamalı bir hiyerarş i oluş turun. Deneysel sonuçlar, hiyerarş ik olarak organize edilmiş gereksinimlerin akıllı bağ lantı yazılımına daha duyarlı olduđ unu göstermektedir.

Etki alanı içi anlam boş luğ unu kapatın. Örneğ in, aş ırı yü klenmiş terminolojiden, yani iki farklı bağ lamda tamamen farklı ş eyler ifade eden sözcü klerden kaçının.

Zengin içerik oluş turun. Her birine gerekçeleri ve alan bilgisini dahil edin gereklilik.

Son olarak, gereksinimleri iyileş tirmek için bir sü reç iyileş tirme planı kullandığ ınızdan emin olun. mü hendislik sü reci.

Disiplinli uygulamaları takip etmek, araç kullanımından daha iyi sonuçlar ve sü reçlerin iyileş tirilebileceğ i bir çerçeve ile sonuçlanabilir. Her proje planı, kullanılacak araçların ve nasıl kullanılacağı nın bir tanımını içermelidir.

Teş vik Destek Teknolojileri

Bu bölü mü , daha önce tartış ılan çeş itli gereksinim belirleme sü reçleri ve tekniklerini desteklemek için kullanılabilecek bazı teknolojilere değ inerek kapatıyoruz.

Bu teknolojiler ş unları içerir:

Wiki'ler

Mobil teknolojiler

Sanal ortamlar

İ çerik analizi

FreeMind, ortaya çıkarmak için bir araç olarak zaten tartış ılmış tı. gereksinimler belirtim belgesi.

Gereksinimlerin Ortaya Çıkarılması için Wiki'leri Kullanma

Wiki'ler, kullanıcıların metin ve görü ntü leri bir web sitesine biçimlendirebildiđ i ve gönderebildiđ i ortak bir teknolojidir. Eriş im kontrolü , parola koruması ve semafor benzeri koruma mekanizmalarıyla sağ lanır (yani, herhangi bir zamanda sitenin belirli bir sayfasına yalnızca bir kullanıcı yazabilir).

Wiki'ler iş birliđ i için, örneđ in grup çalış masını kolaylaş tırmak, kart giriş i (kart sıralama için), ş ablon tamamlama, anketler ve gereksinimler belgesini dü zenlemek için kullanılabilir. Ayrıca, wiki tabanlı gereksinimler doğ rudan yayınlama araçlarına ve doğ rulama araçlarına aktarılabilir. Birçok araş tırmacı, paydaş iş birliđ i (Ferreira ve da Silva 2008), gereksinim görü ş mesi (Yang ve diğ erleri 2008) ve dađ ıtılmış ortaya çıkarma (Liang ve diğ erleri 2009) dahil olmak ü zere çeş itli gereksinim faaliyetleri için wiki tabanlı gereksinim tekniklerini baş arıyla kullandı.

Ek olarak wiki'ler, test senaryolarını otomatikleş tirmeye yardımcı olabilecek etkileşimli belgeler oluş turmak için kullanılabilir (gereksinim belirtimlerinin tü m gereksinimler için kabul kriterleri içermesi gerektiđ ini unutmayın). Örneđ in, FitNesse, bir Fit çerçevesi ü zerine inş a edilmiş ü cretsiz, wiki tabanlı bir yazılım iş birliđ i GUI'sidir. FitNesse, yerleş ik, etkileşimli test senaryolarıyla Web tabanlı gereksinim belgeleri oluş turmak için bir çerçeve sağ lar (FitNesse 2007).

Mobil Teknolojiler

Gereksinim bilgilerini yerinde yakalamak için cep telefonları ve kiş isel dijital yardımcılar gibi çeş itli mobil teknolojiler kullanılabilir. Örneđ in, doktorlar hastalarla çalışırken, yü rü ttü kleri faaliyetlerle ilgili bilgileri doğ rudan gereksinim mü hendisine iletebilirler, bu mü hendis sahada olmak zorunda deđ ildir. Mobil cihazları kullanmak, fikirlerin ve keş iflerin anında kaydedilmesine olanak sağ ladıklarından özellikle yararlıdır. Böyle bir yaklaş ım, beyin fırtınası, senaryo oluş turma, anketler ve diğ er birçok standart gereksinim belirleme tekniklerini, mü ş terinin kolayca eriş ilemediđ i durumlarda (örneđ in, açık deniz yazılım geliş tirme durumlarında olduđ u gibi) destekleyebilir.

Örneđ in, Lutz ve ark. (2012) kullanıcıların akıllı telefonlar ve tabletler gibi Android özellikli cihazlar aracılıđ ıyla rol oynamasına ve bunu yaparken paylaş ılan bir ekranda görü ntü lenen bir temsil modeliyle etkileş ime girmesine izin veren CREWSpace adlı bir uygulama geliştirdi. Yazılım, rol yapma durumlarını takip ederek, nesne yönelimli yazılım sistemlerinde kullanılan bir beyin fırtınası aracı olan CRC kartları oluş turmayı baş ardı. CRC kartları daha sonra gereksinimler belgesine dahil edilebilir.

214 Yazılım ve Sistemler için Gereksinim Mühendisliği

gerektiren durumlarda mobil teknolojilerin kullanımının ortaya çıkışına dair iyi bir tartışma keşifler Maiden ve ark. (2007).

Sanal Ortamlar

Mimari tasarımcılar, uzun yıllardır ticari bina ve ev inşaatı için gereksinimleri ortaya çıkarmak için sanal izlenecek yollar kullandılar. Sanal dünya ortamları, gelişmiş grafikler ve dokunsal basınç sensörleri, kuvvet geri besleme çevre birimleri ve üç boyutlu görüntüleme cihazları gibi çeşitli özel cihazlar kullanan daha karmaşık simülasyonlardır. Sanal dünya ortamları, özellikle yeni veya karmaşık uygulama ortamları için gereksinimlerin test edilmesi, doğrulanması ve kabul edilmesinde gerçekçilik sağlamak için kullanılabilir. Russell ve Creighton (2011), sanal ortamları kullanmanın şunları yapabileceğini öne sürmektedir:

Mevcut eksiklikleri netleştirin

Potansiyel faydalara dikkat çekin

Öncesi ve sonrası koşullarını göster

Çeşitli aktörlerin ve rollerin izlenimlerini kişiselleştirin

Değerlemenin duygulanımsal bileşenlerine ilişkin daha paylaşılan bir takdir yaratmak, ve alternatif zaman çizelgelerinin hazır keşfi

Örneğin, havayolu bagaj taşıma sisteminde, havalimanının ilgili bölümlerinden üç boyutlu bir yerleşim oluşturulabilir ve paydaşlar, dijital sahneyi geçmek ve daha önce iletilen gereksinimleri doğrulamak için karakterleri (avatarları) kullanabilir.

Karmaşık sistemler için sanal ortamlar oluşturmak pahalı olabilir. Ancak yeni ve kritik görev sistemlerinde, sanal simülasyon kullanarak ince gereksinimleri keşfetmenin faydaları maliyete değebilir.

İçerik analizi

İçerik analizi, sosyal bilimlerde yapılandırılmamış bilgiyi yapılandırmak ve anlam bulmak için kullanılan bir tekniktir. Yani, yazılı veya yazılı konuşma eserlerini analiz ederek, paydaş için önemli şeyler elde edilebilir. Bu yazıları analiz ederken amaç, tekrar eden temaları belirlemektir (Krippendorff 2004).

İçerik analizi, gereksinim mühendisleri tarafından grup toplantılarından, yapılandırılmamış görüşmelerden, anket verilerinden veya e-postalardan (metne dönüşümlenilen herhangi bir metin veya yapı) elde edilen dokümanlar gibi yazılı eserler içindeki anlamı keşfetmek için kullanılabilir. Bu gizli anlamlar, paydaşların doğru ifade edemeyecekleri ince gereksinimleri ortaya çıkarabilir. Aslında, bu gizli gereksinimler genellikle en sorunlu olanlardır ve doğru teslim edildiğinde kullanıcı memnuniyetine yol açması muhtemeldir.

Gereksinim Mü hendisliđ i için Araç Desteđ i 215

Bir mü ş teri için akıllı bir ev için gereksinimleri topluyorum .

Mü ş teriyle ne hakkında röportaj yapmak için uzun zaman harcıyorum?
o ister.

Mü ş teriyle onun n ü yaş arken onunla etkileş im kurmak için zaman harcıyorum
ve gibi sorular sorun ("bulaş ık makinesini neden
gece, neden sabah olmasın?")

Mü ş teriyi pasif bir ş ekilde gözlemleyerek uzun zaman harcıyorum.
istediđ i hakkında sözel olmayan ipuçları almak için ş u anki evinde
ve arzular.

Evden bilgi topluyorum – kitaptaki kitaplardan
raf, duvardaki resimler, mobilya stilleri, hobilerin kanıtı,
çeş itli cihazlarda aş ınma ve yıpranma belirtileri vb.

Ş ekil 8.6 Bir akıllı ev belirleme görü ş me oturumundan alınan notların örnek
içerik analizi. Renkler yerine farklı yazı tipleri kullanılmış tır.

İ çerik analizi, çeş itli yazılarda tekrar eden aynı kelimeleri ve benzer ifadeleri etiketleyerek
(renkli vurgulayıcılarla) manuel olarak yapılabilir. Örneđ in, Ş ekil 8.6, bazı metinlerin örnek içerik
analizini içerir. Metin, bir akıllı ev konusunda bir mü ş teriyle röportaj yapan bir gereksinim
mü hendisinin not defterinden bir alıntıdır.

Metni okurken yinelenen temaları fark etmeye baş larız ve bunu yaparken onları farklı renkli
bir iş aretleyici ile vurgularız. Örneđ in, "ben" kelimesi defalarca geçiyor ve her ne sebeple olursa
olsun, bu ismin önemli bir temayı ima ettiđ ine karar veriyoruz ve onu tek renkle vurguluyoruz.
Ardından, akıllı evden veya "ev"den birkaç kez bahsedildiđ ini fark ederiz; bunları önceki temadan
farklı bir renkle vurgularız. Zaman kavramından da sıkça bahsedilir ("uzun zaman dilimleri",
"zaman" ve "zaman dilimleri") ve bu nedenle bunları tutarlı bir renkle vurgularız. Ve benzeri.

Bu sü reci otomatikleş tirmek için içerik analizi için ü cretsiz ve ü cretli araçlar kullanılabilir.

Gereksinim Metrikleri

Metrikler, gereksinim mü hendisliđ i sü reçlerini ve uygulamalarını izlemek ve iyileş tirmek için
gereksinim mü hendisliđ i yaş am döngü sü nü n çeş itli aş amalarında hesaplanabilir veya
toplanabilir. En kullanış ılı gereksinim yönetimi araçları, metriklerin toplanmasını tamamen
otomatikleş tirir veya kısmen yardımcı olur. Costello ve Liu (1995), araçlar tarafından izlenecek
aş ađ ıdaki gereksinim metrikleri tü rlerini tanımlar:

Volatilite
İ zlenebilirlik

216 Yazılım ve Sistemler için Gereksinim Mühendisliği

Bütünleşik
Kusur yoğunluğu
Arıza yoğunluğu
Arayüz tutarlılığı
Sorun raporu ve eylem öğesi sorunları

Gereksinim değişik iş kenliği metriği, zaman içinde gereksinimlerde yapılan silme, ekleme ve değişik işliklerin sayısını izler. Açıkçası, bu sayıların araçlar tarafından takip edilmesi gerekiyor. Genellikle, proje yaşam döngüsü içinde daha sonra ortaya çıkan sorunların, nispeten yüksek oynaklığa sahip gereksinimlere kadar izlenebilir olması söz konusudur.

İzlenebilirlik metrikleri, ARM tarafından toplananlar gibi gereksinim seviyeleri arasındaki bağlantı istatistiklerini ve test kapsamı ve yayılma istatistiklerini içerir. Test senaryosu kapsam metriği, birçok gereksinimin test senaryolarına sahip olduğunu gösterir ve test süresi metriği, bir test senaryosunun kaç gereksinimi kapsadığını gösterir. İz bağlantı metrikleri, gereksinimler, mimari ve tasarım öğeleri arasında da oluşturulabilir. IEEE 29148 standart niteliklerinin belirttiği gibi, yüksek düzeyde dahili ve harici izlenebilirlik arzu edilir.

Tamlik ölçütleri, TBD gibi yer tutucu terimlerin kullanım sayısı ile ilgilidir ve Şekil 4.3'te gösterildiği gibi gereksinim ayrıştırma seviyeleri ile ilgili olabilir.

Hata yoğunluğu metrikleri, bir gereksinim denetimi sırasında ortaya çıkarılan gereksinim hatalarının sayısını gösterir. Bu metrikler, ürün ve süreç oynaklığını tahmin etmede ve diğer metrikleri yorumlamada faydalıdır.

Gereksinim hata yoğunluğu, test yürütme veya son test analizi sırasında başlangıçta algılanan gereksinim hatalarının sayısını gösterir. Gereksinim hataları, bir dizi farklı metrik elde etmek için kritikliğe göre sınıflandırılabilir. Bu metrikler, testin etkinliğini belirlemede kullanılır. Hata yoğunluğu metrikleri, ürün/süreç oynaklığı ve kalitesinin tahmin edilmesinde kullanılır ve diğer metriklerin anlamlarının yorumlanmasında esastır (Costello ve Liu 1995).

Son olarak, problem raporu ve eylem öğeleri, projeyi izlemek ve kontrol etmek için kullanılacak tüm retilmiş metriklerdir. Örneğin, gereksinim başına kusurlar, gereksinimler dalgalanması (gereksinimlerdeki değişiklikler) ve gereksinimler kusur biriktirme listesi. Gereksinim yönetimi araçları bu ölçütlerin çoğunu oluşturabilir.

VIGNETTE 8.1 Çevik Gereksinimler Mühendisliği için Araçlar Çevik

süreçlerdeki gereksinim mühendisliği uygulamalarının eleştirilerinden biri, dokümantasyonun her zaman titiz bir şekilde yakalanmamasıdır. Çalışmalar, çevik pratisyenlerin her zaman proje yönetimine yardımcı olabilecek araç türlerini kullanmadığı ve süreç boyunca uygun yapıları yakalamadığı fikrini desteklemektedir.

sistem geliş tirme. Örneğ in, Kassab (2014), çevik metodolojileri kullanan uygulayıcıların sadece %50'sinin, araçların gereksinim mü hendisliği için kullanıldığını belirttiğini bulmuş tur.

Ancak bazı araçlar, dokü mantasyon eserlerinin kaydedilmesini sağ lamada önemli bir rol oynayabilir ve böylece bu çevik sü reci zenginleş tirebilir. Örneğ in, UML modelleme araçları, gereksinim belgelerinin öğ elerini yakalamaya yardımcı olabilir. Çevik ekipler, saha dış ındayken mü ş teriler ve ekip ü yeleriyle iletişim

halinde kalmak için Web tabanlı paylaş ılan ekip projeleri ve anlık mesajlaş ma araçları gibi geliş miş iletişimlileri de kullanabilir.

Bu araçlardan elde edilen eserler daha sonra yakalanabilir ve gereksinim belirtiminin öğ elerine dönü ş tü rülebilir.

Araş tırmacılar ayrıca, çevik projelerin, doğ rulama araçlarıyla birlikte gereksinim izlenebilirlik araçlarını kullanmanın da fayda sağ layacağını bulmuş lardır. Salatalık gibi entegre kabul testi araçları, gereksinimlerin ve test belgelerinin kalitesini de iyileş tirebilir (De Lucia ve Qusef 2010).

Egzersizler

- 8.1 Uygun bir gereksinim mü hendisliği i aracının seçilmesinde hangi kriterler kullanılmalıdır?
- 8.2 Gereksinim mü hendisliği inde belirli araçları kullanmanın herhangi bir sakıncası var mı? faaliyetler?
- 8.3 Açık kaynaklı bir araç seçerken hangi özellikleri aramalısınız?
- 8.4 Araçlar, dağ ıtılmış , kü resel gereksinim mü hendisliği i etkinliklerini nasıl etkinleş tirebilir? bağ lar? Bu konudaki eksiklikler nelerdir?
- 8.5 Bir ortam ş u anda katı gereksinimler mü hendisiyle ilgilenmiyorsa uygulamalar, araçlar tanıtılmalı mı?
- 8.6 Oluş turduğ unuz bir izlenebilirlik matrisi aracılığı ıyla ne tü r sorunlarla karş ılaş abilirsiniz? biri olmadan göremez mi?
- 8.7 FreeMind'i indirin ve akıllı telefonunuz için bir zihin haritası ü zerinde beyin fırtınası yapmak için kullanın. ev sistemi.
- 8.8 Ekteki Bölü m 7.2'de açıklanan gereksinim için bir FitNesse (veya Salatalık) test tablosu oluş turun.
- 8.9 DOORS'un en son sü rü mü nü indirin ve kurun (bu yazı yazıldığı ı sırada NG). Ş u anda tamamlamakta olduğ unuz kurs projесinin gereksinimlerini yönetmek için yeni bir proje oluş turun.
- *8.10 Gereksinim yönetimi araçlarını değ erlendirmek için Sud ve Arthur boyutlarını kullanarak, kamuya açık bilgileri kullanarak beş farklı ticari veya açık kaynaklı aracın değ erlendirmesini yapın.

218 Yazılım ve Sistemler için Gereksinim Mühendisliği

Referanslar

- Carrillo de Gea, JM, Nicolás, J., Alemán, JLF, Toval, A., Ebert, C., & Vizcaino, A. (2011). Gereksinim mühendisliği araçları. *IEEE Yazılımı*, 28(4): 86–91.
- Carrillo de Gea, JM, et al. (2015). Gereksinim mühendisliği araçları arasındaki benzerlikler ve farklılıklar: Nicel bir yaklaşım. *Bilgisayar Bilimi ve Bilgi Sistemleri*, 12(1): 257–288.
- Cleland-Huang, J., Settini, R., Romanova, E., Berenbach, B., & Clark, S. (2007). Otomatik izlenebilirlik için en iyi uygulamalar. *IEEE Bilgisayar*, Cilt 40, 27–35.
- Costello, RJ ve Liu, DB (1995). Gereksinim mühendisliği için metrikler. *Sistem Dergisi ve Yazılım*, 29(1): 39–63.
- De Lucia, A., & Qusef, A. Çevik yazılım geliştirmede gereksinim mühendisliği. *Web Zekasında Gelişen Teknolojiler Dergisi*, 2(3): 212–220.
- Daud, N., Kamalrudin, M., Sidek, S., & Ahmad, SSS (2014). Gereksinim doğrulama için gereksinim mühendisliği araçlarının gözden geçirilmesi. *Uluslararası Yazılım Mühendisliği ve Teknolojisi Dergisi*, 1(1).
- Eriksson, M., Morasi, H., Borstler, J. ve Borg, K. (2005). PLUSS araç takımı—Telolojik KAPILARI ve IBM rasyonelini genişletmek, kullanım senaryosu modellemede ürün hattını desteklemek için yürüldü. 20. IEEE/ACM Uluslararası Otomatik Yazılım Mühendisliği Konferansı Bildirilerinde, Long Beach, CA, s. 300–304.
- Ferreira, D. ve da Silva, AR (2008, Eylül). Wiki, iş birliği için dayalı gereksinim mühendisliği için destekledi. 4. Uluslararası Wiki Sempozyumu Tutanakları içinde, ACM.
- FitNesse projesi. (2007). www.fitnesse.org (Ocak 2017'de erişildi).
- FreeMind projesi. (2017). http://freemind.sourceforge.net/wiki/index.php/Main_Page (Şubat 2017'de erişildi).
- Gandhi, P., Haugen, NC, Hill, M., & Watt, R. (2005). Uygun belgeleri kullanarak yayın belirti oluşumu. *Çevik 2005 Konf. (Agile 05)*, IEEE CS Press, s. 253–258.
- Hammer, TF, Huffman, LL, Rosenberg, LH, Wilson, W., & Hyatt, L. (1998). Gereksinimleri ilk seferde doğru yapmak. *CROSSTALK Savunma Yazılım Mühendisliği Dergisi*, 20–25.
- Heindl, M., Reinisch, F., Biffel, S., & Egyed, A. (2006). Değer tabanlı gereksinim mühendisliği araç desteği seçimi. 32. EUROMICRO Yazılım Mühendisliği ve İleri Uygulamalar Konferansında. SEAA' 06, Ağustos 2006, s. 266–273.
- Hull, E., Jackson, K. ve Dick, J. (2011). Kapılar: Gereksinimleri yönetmek için bir araç. *Gereksinimler Mühendislik*, 181–198. Springer-Verlag, Londra.
- ISO/IEC TR 24766. (2009). Bilgi teknolojisi—Sistemler ve yazılım mühendisliği—Gereksinim mühendisliği için yetenekleri için kılavuz. Uluslararası Standardizasyon Örgütü (ISO), Cenevre, İsviçre.
- Kasab, M. (2014). Çevik yazılım geliştirme için gereksinim mühendisliği uygulamaları üzerine ampirik bir çalışma. 2014 yılında 40. EUROMICRO Yazılım Mühendisliği ve Gelişmiş Uygulamalar Konferansı (SEAA), IEEE.
- Kassab, M. & Ormandjieva, O. (2014). Süreç odaklı yaklaşımlarda işlevsel olmayan gereksinimler. *Yazılım Mühendisliği Ansiklopedisi*, Ed., Phillip A. Laplante, s. 1–11. Taylor ve Francis, New York.
- Krippendorff, K. (2004). *İçerik Analizi: Metodolojisine Giriş*. Adaçayı. Bin Oaks, CA.
- Lee, S.-W., Gandhi, RA ve Park, S. (2011). *İhtiyaç takibi*. P. Laplante'de (Ed.), *Yazılım Mühendisliği Ansiklopedisi*, s. 999–1011. Taylor ve Francis. Boca Raton, Florida.

- Liang, P., Avgeriou, P. ve Clerc, V. (2009, Temmuz). Anlamsal wiki kullanarak dağıtılmış gereksinim analizi için gereksinimler muhakemesi. Dördüncü IEEE Uluslararası Küresel Yazılım Mühendisliği Konferansı'nda, 2009. ICGSE 2009, s. 388–393, IEEE.
- Lutz, R., Schäfer, S., & Diehl, S. (2012, Eylül). İşbirlikçi gereksinim mühendisliği için mobil cihazları kullanma. 27. IEEE/ACM Uluslararası Otomatik Yazılım Mühendisliği Konferansı Tutanakları bölümünde, s. 298–301, ACM.
- Maiden, N., Omo, O., Seyff, N., Grunbacher, P., & Mitteregger, K. (2007). İş yerinde paydaş ihtiyaçlarının belirlenmesi: Mobil teknolojiler nasıl yardımcı olabilir. IEEE Yazılımı, Cilt 24, 46–52.
- RTCA. (2011). Havadaki Sistemler ve Ekipman Belgelendirmesinde Yazılım Konuları. DO-178C. Havacılık için Radyo Teknik Komisyonu (RTCA), Washington, DC.
- Russell, S. ve Creighton, O. (2011, Ağustos). Gereksinim mühendisliği için sanal dünya araçları. 2011'de Multimedya ve Keyifli Gereksinimler Mühendisliği-Yalnızca Açıklamaların Ötesinde ve Daha Fazla Eğlence ve Oyunlarla (MERE) Dördüncü Uluslararası Çalıştay, s. 17–20.
- Sud, RR ve Arthur, JD (2003). Gereksinim Yönetimi Araçları: Niceliksel Bir Değerlendirme. Teknik Rapor TR-03-10. Bilgisayar Bilimi, Virginia Tech. Blacksburg, VA.
- Volere gereksinimleri kaynakları. (2013). <http://www.volere.co.uk/tools.htm> (Şubat 2017'de erişildi).
- Yang, D., Wu, D., Koolmanojwong, S., Brown, AW ve Boehm, BW (2008, Ocak). Wikiwinwin: İşbirlikçi gereksinimleri müzakeresi için wiki tabanlı bir sistem. 41. Yıllık Hawaii Uluslararası Sistem Bilimleri Konferansı Bildirilerinde, s. 24–24, IEEE.
- Yue, T., Briand, LC ve Labiche, Y. (2011). Kullanıcı gereksinimleri ve analiz modelleri arasındaki dönüşüm yaklaşımlarının sistematik bir incelemesi. Gereksinim Mühendisliği, 16(2): 75–99.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>