İyi Gereksinim Yazma Teknikleri

Kasım ŞEN

KoçSistem Bilgi ve İletişim Hizmetleri A.Ş, Yazılım Teknoparkı, Teknokent, ODTU/Ankara

kasims@kocsistem.com.tr

Özet

Gereksinim mühendisliği kapsamında gereksinimleri ortaya çıkardıktan sonra bu gereksinimleri belgelemek ve paydaşlara sunmak önemlidir. Gereksinim mühendisliğinin en önemli amacı kullanıcı ihtiyaçlarını açık, doğrulanabilir, kısa ve öz gereksinimlere dönüştürebilmektir. Tüm paydaşların gereksinimlerden aynı anlamı çıkarması, dokümanda tutarsızlık olmaması ve gereksinimlerin kaliteli olması gereklidir. Bu nedenle yazılan gereksinimlerin iyi bir gereksinim olarak tanımlanabilmesi için bazı kriterlerin ortaya konması gerekir. Gereksinimleri elde etmenin yanı sıra kaliteli gereksinim yazabilmek de oldukça zordur. Tam, eksiksiz ve tüm paydaşların anlayabileceği gereksinimler, yazılımın kalitesini de doğrudan etkileyecektir. Bu makalede yazılım gereksinimlerini kaliteli ve iyi biçimde yazma üzerine odaklanılmıştır. Bu makalede iyi gereksinimlerin özellikleri ve kontrol listesi önerilmektedir.

Abstract

After eliciting requirements of software within the requirements engineering process, documenting the requirements and presenting to all stakeholders is very important. Important aspect of requirements engineering, transform the needs to the requirements as clear, simple and concise. Getting same meaning from requirements by all stakeholders, being consistent and quality requirements are very important. Some criteria are provided to write good requirements. It is also difficult as eliciting requirements. Writing good requirements influence the quality of software. This paper focuses on characteristics of good requirements and check lists.

1. Giriş

Gereksinim mühendisliği kapsamında gereksinimleri ortaya çıkardıktan sonra, bu gereksinimleri belgelemek ve paydaşlara sunmak önemlidir. Gereksinim mühendisliğinin en önemli amacı kullanıcı ihtiyaçlarını açık, doğrulanabilir, kısa ve öz gereksinimlere dönüştürebilmektir Tüm paydaşların gereksinimlerden aynı anlamı çıkarması, dokümanda tutarsızlık olmaması ve gereksinimlerin kaliteli olması gereklidir. Bu nedenle yazılan gereksinimlerin iyi bir gereksinim olarak tanımlanabilmesi için bazı kriterlerin ortaya konması gerekir. Gereksinimleri elde etmenin yanı sıra kaliteli gereksinim yazabilmek de oldukça zordur. Tam, eksiksiz ve tüm paydaşların anlayabileceği gereksinimler yazılımın kalitesini de doğrudan etkileyecektir.

Birçok yazılım gereksinim dokümanı kötü yazılmış gereksinimlerle doldurulmuştur. Bir ürünün kalitesi ürünü oluşturan maddelerin kalitesine bağlıyken, yazılımın kalitesi de gereksinimlerin

kalitesine bağlıdır. Bu makale ile iyi gereksinimlerin karakteristik özellikleri açıklanmış ve bu özellikler için ipuçları verilmiştir. Mükemmel gereksinim yazmak imkansızdır ve gerekli de değildir, önemli olan yazılan gereksinimlerin, iyi gereksinimin taşıması gereken özelliklerin büyük çoğunluğunu taşımasıdır. En iyiye ulaşmak için çaba gösterilmelidir. Birçok zorluklarla elde edilen gereksinimleri eksik, hatalı, anlaşılmaz veya yanlış anlaşılan gereksinimler olarak belgelemek, yazmak; hem harcanan emeğin kaybolması hem de gereksinim elde etme sürecinin başına dönerek aynı şeyleri tekrar yapmak ve bu nedenle yazılımın maliyetini arttırmak demektir.

2. İyi Gereksinimlerin Özellikleri

Gereksinimleri yazmak ilk bakışta kolay bir iş olarak görülür. Çoğunlukla gereksinim kaynaklarının söylediklerini doğrudan yazmak veya dokümanlardan elde edilen gereksinimleri aynen aktarmak, gereksinimleri yazmak olarak algılanabilir. Bunun bir tekniğinin olduğu ve gereksinimlerin de belli nitelikler taşıması gerektiğinin farkına varılamaz[1]. Kaliteli bir gereksinimin içermesi gereken özellikleri konusunda üzerinde tam bir uzlaşma sağlanmış kriterler kümesinden bahsedilemez. Yaşanmış tecrübelerden paylaşılmış olanlardan, düşülen hatalarda varılan sonuçlardan ve yapılan ortak sınıflandırmalardan bazı özellikler ortaya çıkarılmıştır. Gereksinimlerin bu özellikleri taşıyacak biçimde yazılmasının tasarım sürecindeki geri dönüşleri azaltacağının farkına varılmalıdır. Tüm dokümanı kontrol listeleri ile baştan sona gözden geçirmek yerine her bir gereksinime gereken önem ve özeni göstererek yazmak daha etkili ve verimli bir işlemdir.

Gereksinimlerin söz diziminde şu yapılar olmak zorundadır[2].

- Kullanıcı ve aktör: Sistemden fayda sağlayacak kişi veya nesne
- Eylem: –cak" gibi gelecek zaman kipinde olmalı ya da geniş zaman kipinde olmalıdır.
- Nesne: Ortaya çıkacak sonuç ürün, bilgi, fayda
- Niteleyici: Ölçümlenebilir, belirleyici değer aralığı

Örnek bir gereksinim:"Çağrı merkezi operatörü, arananın detaylı bilgilerini iki saniye içinde sorgulayıp sonucunu görebilecektir" (Gelecek zaman)
Diğer bir örnek gereksinim ise: "Pilot, uçağın kalkış açısını tek eliyle kontrol eder" (Geniş zaman)

2.1. Doğruluk

Gereksinimler, sunulacak olan fonksiyonu doğru olarak tanımlamalıdır[2]. İstenilen ile gereksinimde açıklananların farklı olması halinde gereksinim kaynağına bakılmalıdır. Gereksinimlerin türüne göre uygun kaynağın belirttiği ifadeler dikkate alınmalıdır. Çatışmalar olması halinde her iki kaynağa da gereksinimler doğrulatılmalıdır. Bu yöntemde yanlış gereksinime neden olan kaynak ortaya çıkacaktır. Çoğunlukla uç kullanıcılar ile yöneticilerin gereksinimleri birbiriyle çatışabilmektedir. Bunun nedeni de yöneticinin olayların detayına inmeden, yüzeysel bakış açısından gereksinimlere odaklanması, uç kullanıcıların da yönetimsel kararlara bağımlı gereksinimler hakkında hüküm vermesidir. Örneğin, yönetim tarafından alınmamasına karar verilen haftalık bir raporu, o raporu önceden takip eden kullanıcı zorunlu bir gereksinim olarak sunabilir.

2.2. Zorunluluk

Gereksinimler, geliştirilecek sistem için mutlaka zorunlu olmalıdır[1]. Gereksinim olarak yazılan ifadenin dokümandan çıkarılması veya hiç eklenmemesi halinde değişen bir şey olmuyorsa bu ifade gereksinim olarak değerlendirilemez. Gereksinimlerin zorunluluğu konusunda tam bir mutabakat olmalıdır. Paydaşlar arasında gereksinimin geliştirilmesine yönelik şüpheler varsa bu gereksinimin gerekliliğinin yeniden değerlendirilmesi ve ortak bir kararda mutabık olunması gerekir. Bazı kullanıcılar dilek ve temennilerini zorunlu gereksinim gibi sunabilmektedir. Örneğin, kullanıcılar

birçok işlemin otomatik olarak sistem tarafından yapılmasını isterler. Tasarımı zorlaştıracak, proje maliyetini arttıracak bu tip gereksinimler için kullanıcının iş yapış biçimlerinin yeniden araştırılması ve bu gereksinimin fayda-maliyet analizinin yapılarak zorunluluğuna karar verilmesi gerekir. Ayrıca geliştirilecek sisteme direnç gösteren kimi kullanıcılar da maliyetin arttırılması için kişisel kaprislerine kapılarak olmayacak gereksinimleri zorunlu gibi sunabilmektedir.

2.3. Doğrulanabilirlik, Ölçülebilirlik

Gereksinimler, ölçülebilir değerler tarafından doğrulanabilir olmalıdır[1]. Doğrulama yapabilmek için ilgili gereksinimin kabul değerleri veya değer aralıkları belirlenmelidir.Gereksinimler sübjektif ifadeler içermemelidir. Sübjektif ifadelerin kabul değerleri kişiden kişiye değişebileceği için ölçümleme yapılamaz. Örneğin "güzel", "kolay", "hızlı" gibi ifadelerin ölçümlemesi yapılamayacağından bu tip ifadeleri içeren gereksinimler iyi gereksinimler değildir. Gereksinimleri yazarken kabul kriterlerini yazma zorunluluğu olmamasına rağmen, testlerde kullanılmak üzere doğrulanabilirlik açısından ölçülebilir kabul kriterlerin gereksinimlerde belirtilmesi faydalı olmaktadır. Çünkü doğrulanamayan gereksinim aslında gereksinim değildir. İyi bir gereksinim test edilirken belli kıstasları destekleyebilmelidir. Kişilere göre değişen kabul kriterleri belirlenirse test yapanın o andaki ruh haline bağımlı olarak ya tüm testlerden başarıyla geçer ya da başarısız sayılır.

2.4. Ulaşılabilirlik, Erişilebilirlik

Gereksinimler, sistemin kısıtları kapsamında ulaşılabilir ifadeler olmalıdır[1]. Projenin bütçe, zaman ve insan kaynağı ile gerçekleştirilemeyecek gereksinimler erişilebilir değildir. Gereksinimler, projenin kısıtlarını zorlamayacak şekilde ya yeniden düzenlenmelidir ya da tamamen dokümanlardan çıkarılmalıdır. Gereksinimin erişilebilir olup olmadığını anlayabilmek için teknik fizibilite çalışması yapılmalıdır. Teknik olurluğun olması bile projenin zaman ve bütçe kısıtları nedeniyle ilgili gereksinimin erişilebilir olmasını sağlayamaz. Kullanıcılara izin verilirse yazılımın gelecekten haber vermesini bile isteyebilirler. Her gereksinimin ek kaynak ve maliyet olarak düşünülmesi ve proje planındaki sınırlara uyulması gereklidir. Teknik olurluğu sağlanan ama kaynak açısından kabul edilemeyen gereksinimlerin zorunluluğu ortaya çıkarsa, müşterinin bilgisi dahilinde proje planının revize edilmesi ve zaman planının yeniden yapılması gereklidir.

2.5. Açıklık, Sadelik

Gereksinimler tek bir konudaki gereksinimi açıklamalı ve açık, öz, anlaşılır ve basit olmalıdır[2]. Gereksinimler ve gereksinim dokümanlarının edebi içeriklere sahip olmasına gerek yoktur. Ağdalı ve anlaşılması güç ifadelerin kullanılması halinde gereksinimin asıl amacından çıkılabilir. Basit ve bağlantı ekleri içermeyen, muğlak olmayan, ihtimallere yer vermeyen, tartışılmayacak kadar açık ve net ifadeler ile iyi bir gereksinim yazılır. Gereksinimi okuyan kişiler gereksinim üzerinde yorum yapmasına gerek duymayacak biçimde muğlak olmayan, açık ve net olmalıdır. Konuşma dilinde yazılan gereksinimler belirsiz olmaya yatkındır. Belirsizliği azaltmak için şu kelimelerden kaçınılmalıdır: Kolay, basit, sade, hızlı, verimli, birkaç, maksimum, minimum.

2.6. Tutarlılık

Gereksinimler birbirleriyle tutarlı olmalıdırlar[2]. Birbiriyle çelişen, çatışan, bütünlüğü bozan gereksinimler olmamalıdır. Özellikle gereksinim kaynaklarının çeşitliği nedeniyle bir kaynaktan alınan gereksinim diğeriyle çelişebilir. Bu durumda gereksinimler arasında izlenebilirlik ve ilişki şemalarının hazırlanması ve tutarsızlıkların tespit edilmesi gerekir. Bu durum gereksinimlerin doğruluğunu da azaltacaktır. Doğruluk kısmında açıklamalar yapılmıştır.

2.7. Önceliklendirilebilirlik

Her gereksinim geliştirme sırasına göre önceliklendirilmelidir[2]. Herkes kendi gereksiniminin öncelikli olduğunda ısrar etse de mutlaka bir kısmı daha da önceliklidir. En azından üç kademeli bir önceliklendirme yapılmalıdır. Bunlardan yüksek önceliğe sahip gereksinimler, ilk sürümde hazırlanacak olan ve olmazsa olmaz olanlar. Orta önceliğe sahip olanlar, önemli ve zorunlu olmasına rağmen yüksek öncelikli gereksinimler tamamlandıktan ele alınacak olan gereksinimler. Düşük öncelikli olanlar ise kaynaklar el verdiği sürece yapılacaktır. Her gereksinimin bir öncelik sınıfı vardır. Kullanıcılardan gereksinimleri elde ederken düşünülen öncelik sınıflarının kullanıcılara açıklanması ve kullanıcılardan her bir gereksinime bir öncelik vermesi istenmelidir.

2.8. Değişebilirlik

Gereksinimler daha sonra ortaya çıkan nedenlerden dolayı değişikliğe uğrayabilir[2]. Bu durumda gereksinimlerin önceki haline kolayca erişim gerekir. Bunun için de gereksinimlerin tekil olacak şekilde numaralandırılması ve birbiriyle ilgili gereksinimlerin gruplanması sağlanmalıdır. Değişikliğe ait bir tarihçe yapısı kurulması, ileride ortaya çıkabilecek tartışmalarda belge olacaktır. Ayrıca değişikliğin sebebi ve nelere istinaden bu değişikliğin yapıldığının bilgisi de yazılmalıdır. Yapılan bir toplantı veya görüşme nedeniyle bir değişiklik yapılmış ise toplantı raporuna ait tarih, kişi ve yer eklenmelidir. Bu yöntem değişikliğin aksinin iddia edilmesinde geçerli bir savunma olacaktır.

3. Gereksinimler Nasıl Yazılmamalı

Gereksinimleri iyi yazmanın yöntemi, gereksinimleri nasıl yazmamaktan geçer. Yapılan hatalar genelde aynı türden olduğu için neler yapılmaması gerektiğini ortaya çıkarmak daha kolaydır. Böylece önceden düşülmüş tuzaklara yeniden düşülmeyerek hatasız, anlaşılır, eksiksiz ve geri dönüşü olmayan gereksinimler yazılmış olur. Aşağıda belirtilen ipuçları yazılım mühendislerinin yaşadıklarından ve tecrübelerinden ortaya çıkmış olanlardır. Bunların dışında da bazı kritik ipuçları, yol gösteren yaşanmış senaryolar da mevcut olabilir. Bunların da paylaşılması halinde aşağıdaki liste daha da zenginleşecektir.

3.1. Belirsizlik ve birden fazla anlam taşıyan ifadeler yer almamalıdır

Bu durum en çok düşülen bir hata olmasına rağmen gereksinimleri açık ve anlaşılır biçimde yazmak oldukça zordur[2]. Bir de anlaşılırlığı arttırmak için çok fazla kelime kullanmak da gereksinimlerin okunurluğunu azaltarak sıkıcı bir hal alacaktır. Özellikle “ya da”, “veya”, “bununla birlikte” gibi kelimeler içeren gereksinimler potansiyel muğlak gereksinimlerdir ve bu kelimeler gereksinimin belirsizliğini arttıracaktır. Mümkün olduğunca terimler kullanılmamalıdır. Kullanılması gereken hallerde bunların bir terim sözlüğünde detaylı açıklaması yapılmalıdır. Ayrıca “kullanıcı-dostu”, “çok yönlü”, “esnek”,”yaklaşık”,”tahminen”, “etkili”, “verimli”, “modern” gibi kişiden kişiye değişen kelimeler de kullanılmamalıdır Örneğin, “Aynı sistem hata halinde görsel veya sesli uyarılar verecektir” gibi bir gereksinim yazılırsa, hata mesajlarının sesli mi görsel mi olacağı belirsizdir. “Aynı sistem” neye karşılık gelmektedir. Uyarılar kullanıcılara nasıl verilecektir. Uyarıların hergün adresine mektup ile postalanması da bu gereksinimi sağlar görünmektedir.

3.2. Gereksinimler bağlaçlar ile birbirine bağlanmamalıdır

Gereksinimlerde cümleleri bağlamak amacıyla "ve", "veya" bağlaçları ile geçiyorsa bunları ayrı ayrı gereksinim olarak yazmak gerekir[2]. Böylece hem gereksinimlerin okunurluğu arttırılmış olur hem de test edilirken her birine ayrı ayrı önem gösterilmiş olur.

3.3. Gereksinimleri genişleten, farklı yönlere çeken, dağıtan kelimelerden sakınmak gerekir

Bu durum daha çok gereksinimleri test ederken sorun oluşturur. Şöyle ki; test yapılırken yorumlama yapmak gerekir, koşulun tersi durumlarını, alternatif durumlarını da test etmek gerekecektir. Gereksinimleri yazarken "ise", "-diği zaman", "-in dışında", "-madıkça", "-masına rağmen" gibi ekleri içeren sözcüklerden kaçınılmalıdır. Örneğin, "Kullanıcıların veri girişi dışında aldığı hata mesajları alındığı zaman program otomatik olarak sistemi kapatacaktır".

3.4. Gereksiz yere uzun cümleler kurarak gereksinimler yazılmamalı

Gereksiz yere uzun cümleler kurarak gereksinimleri yazmamalı, konuyu dağıtmamak gerekir. Uzun gereksinimler birkaç adet gereksinime parçalanabilir. Gereksinim yazmak edebi eser yazmak değildir. Uzun cümleler hatta paragraflar yazmak işin teknik tarafı ile ilgilenen geliştiriciler ve kullanıcılar için sıkıcılık yaratır.

3.5. Gereksinimleri yazarken tasarım yapılmamalıdır

Gereksinimleri yazarken tasarım yapmak çok kolayca düşülen bir tuzaktır[2]. Gereksinimleri detaylı yazmaya başladıkça bazen gereksinim diye yapacağımız tasarımı yazarız. Analiz ve tasarımı aynı grup yapacak ise çoğunlukla bu tehlikeyle karşılaşılır. Gereksinimlerde bileşenlerin, nesnelerin veya sınıfların ismi geçmemelidir. Fonksiyon veya yordamlar tanımlanmamalıdır. Veritabanına yönelik ifadeler, tablo isimleri, alan isimleri ve indekslerden bahsedilmemelidir. Tasarıma yönelik ifadeler, gereksiz kısıtlarla kendimizi bağlamamıza neden olabilir. Örneğin, "Muhasebe hesapları veritabanında "hesap" tablosunda tutulacaktır", "Müşteri tablosunda ikincil indeks yaratılacaktır" gibi gereksinimler tamamen tasarıma yöneliktir. Muhasebe hesaplarını veritabanında değil de XML dosyalarda tutma yönünde karar verilirse gereksinimler bağlayıcı olur.

3.6. Gereksinimler ile planlar karıştırılmamalıdır

Gereksinimler ile planlar farklı algılanmalı ve karıştırılmamalıdır[2]. Planlar, işin temelini oluştururlar, gereksinimlere bağımlı değildirler, şartlara göre revize edilebilirler ama gereksinimler kararlı ve stabilize olmalıdır. Planların gereksinim olarak alınması halinde sürekli revize etme tehlikesini de göze almak gerekir.

3.7. Spekülasyon yapılmamalıdır

Gereksinimler, kullanıcılar ile geliştiriciler arasında yapılmış bir kontrattır[2]. Burada net olmayan üzerinde spekülasyon yapılabilecek kelimelerden kaçınılmalıdır. Bu kelimeler "çoğunlukla", "bazen", "ara sıra", "genellikle", "sık sık" gibi kelimelerdir. Herşeyi net olarak ortaya koymak daha sonra spekülasyon yapılmasının önüne geçecektir.

3.8. Gereksiz kuruntulardan kaçınılmalıdır

Gereksiz kuruntulardan kaçınılması gerekir. Hiçbir sistem mükemmel değildir[2]. İmkansızı istemek ve bunu sorun haline getirmek gereksizdir. Bu kuruntuları yaratan veya yaratabilecek kelimeler şunlardır: "%100 güvenilir", "asla bozulmayan", "her duruma göre uyarlanabilecek kadar esnek", "tüm platformlarda çalışan", "her kullanıcı", "her bir" vb. Örneğin "Program tüm hataları yöneterek işleyişine devam etmelidir" şeklinde bir gereksinimi sağlayacak bir sistem imkansızdır.

Burada hata türleri belirtilmeli ve diğer hatalarda nelerin yapılması gerektiği, hangi durumlarda sistemin işleyişinin sonlandırılacağı net olarak belirtilmelidir.

3.9. İhtimal içeren ifadelerden kaçınılmalıdır

Gereksinimlerin içinde geçen "-ebilir, belki, olasılıkla" gibi kelimeler şüpheli durumlar oluşturur[2]. Yazılım geliştiriciler, kullanıcıların mutlaka yapabilmesi gereken işlemlere odaklanırlar. Bu durumda böyle ihtimal içeren işlevler göz ardı edilir veya ertelenir. Bir şey ya yapılmalıdır ya da reddedilmelidir.

4. İyi Gereksinim Yazma Klavuzu

İyi gereksinim yazmak için kesin bir formül olmamasına rağmen bazı ipuçları verilebilir[3].

- Gereksinimleri kısa tutunuz. Paragraf boyutunda gereksinim yazmayınız.
- Dilin gramer, noktalama ve heceleme kurallarına uygun ifadeler yazınız.
- Terimler kullanmamaya özen gösterilmeli. Kullanılacak ise bunlar veri sözlüğünde açıklanmalıdır.
- Gereksinimlerin yeterince açıklandığını anlamak için bunları hem kullanıcı hem de geliştirici gözüyle tekrar okuyunuz.
- Anlaşılmayan gereksinimler için geliştiricilerin size sorular sormak zorunda kalacağını ve buna ne kadar dayanabileceğinizi düşününüz. Bu durumu her gereksinim için hayal ediniz.
- Yazılan gereksinimler ile tasarım ve geliştirme yapılacak kadar açık olması gerekir. Tasarım veya geliştime aşamasında fazladan açıklama ve açığa kavuşturma çabası gösterilmeyecek şekilde yazılmalıdır.
- Kullanıcılardan hikaye biçiminde alınan gereksinimleri, parça parça anlamlı gereksinimler halinde yazmak için en iyi yol yazılan her bir gereksinimin tek başına test edilebilir olmasını sağlamaktır.
- Gereksinimleri "ve", "veya", "ki" bağlaçlarıyla birbirine bağlamayınız.
- Gereksinimler bütünlüğü bozmamalı ve tutarlı olmalıdır. Kapsamın dışına çıkılmamalıdır
- Mükerrer gereksinimler olmamalıdır. Mükerrer gereksinim olması halinde değişikliklerin tümünde yapılması gerekecektir.

5. İyi Gereksinimler için Kontrol Listesi

Gereksinimleri yazdıktan sonra kontrol edebilmek için hazır olan kontrol listelerinden yararlanılabilir. Örnek bir liste aşağıda açıklanmıştır[4]. Buradaki sorular arttırılabilir ama her bir gereksinim için tüm bu soruların cevaplarını almak gereklidir.

- ***Zorunluluk:*** Bu gereksinim olmadan yazılımdan beklentiler karşılanabiliyor mu?
- ***Doğrulanabilirlik:*** Bu gereksinimin doğrulama kriterleri nelerdir? Bu kriterler gereksinimde açık olarak belirtilmiş midir? Doğrulama kriterleri sistemin sınırları içinde mi?
- ***Ulaşılabilirlik, erişilebilirlik:*** Bu gereksinim sistem kaynakları ve kısıtları içinde gerçekleştirilebilir mi? Gereksinimden beklentileri sağlamak imkansız mı?
- ***Açıklık, belirlilik:*** Bu gereksinim herkes tarafından aynı şekilde yorumlanabilir mi? Farklı anlamlar çıkartabilecek kelimeler yer alıyor mu? Gereksinim içinde terimler geçiyor mu? Cevap evet ise bunlar veri sözlüğünde detaylıca tanımlanmış mı?
- ***Tamlık, eksiksizlik:*** Gereksinimde karşılaşılabilecek tüm koşullar belirtilmiş mi? Müşteriden alınan tüm gereksinimler dokümanda yer alıyor mu?
- ***Tutarlılık:*** Gereksinim önceki gereksinimlerle çelişiyor mu? Gereksinim dokümanın bütünlüğünü bozuyor mu?

- ***İzlenebilirlik:*** Gereksinimin kaynağı belli mi? Gereksinimler arasında ilişki kurulmuş mu? Kurulmuş ilişkilerde iç içe geçme durumu var mı? Döngüsel izlenirlik oluş mu?
- ***Yer almak*:** Gereksinim sistem tasarımında yer alıyor mu? Tasarımsal olarak gerçekleştirilmiştir? Tasarımsal karşılığı var mıdır?
- ***Basitlik:*** Gereksinim sade ve açık mıdır?
- ***Tasarımdan bağımsızlık:*** Gereksinim sadece ne yapılacağına cevap veriyor mu? Nasıl yapılacağına ait ifadeler var mı? Gereksinim içinde tasarım öğeleri yer alıyor mu?
- ***Standart Yapılar:*** Gereksinimler gelecek zaman kipinde mi yazılmış? Dil kurallarına uyulmuş mu? Gereksinimde etkilenen aktör belirtilmiş midir?
- ***Tekil tanımlayıcı:*** Her gereksinim numaralandırılmış mıdır? Bu numaralar tekil olarak mı verilmiş? Gereksinimin değişiklik tarihçesi izlenebiliyor mu? Gerekli versiyon numaraları takip edilmiş mi?

Kaynakça

[1] Ivy Hooks, "Writing Good Requirements", 3. International Symposium of the INCOSE – Sayı 2, 1993.

[2] Ian F Alexander, Richard Stevens, "Writing Better Requirements", Pearson, 2002, ISBN 0-321-13163-0, sayfa 67-75.

[3] Karl E. Wiegers, "Writing Quality Requirements", Process Impact, www.processimpact.com

[4] Dr. Ralph R. Young, "Criteria of a Good Requirement", The Requirements Engineering Handbook (Artech House, 2004), www.ralphyoung.net