

Yazılım Gereksinim Analizi

Yazılım Hakkındaki Yanılgılar

- Müşteri açısından yanılgılar;
 - Programa başlarken amaçları genel olarak belirlemek yeter, ayrıntılar sonra kararlaştırılabilir.
 - Yazılım esnektir. Değişen gereksinimler kolayca sisteme uyarlanabilir.
 - Sonuç: Yazılım esnek bir oyun hamurundan çok kil veya cam gibidir.

- Programcı açısından yanlışlar;
 - Yazılımı tamamlayıp müşteriye teslim edince işimiz biter.
 - Yazılımı tamamlamadan kalitesini ölçemem.
 - Yazılım eşittir program. Gereksinim analizine dokümantasyona ihtiyaç yok.
 - Yazılım mühendisliğinin gereklerini uygulayarak boşuna çaba harcıyoruz.

- İdari açıdan yanılgılar;
- İşler yetişmiyorsa takıma yeni programcılar ekleriz.
- Geliştirmesini üstlendiğim yazılımı tamamen veya kısmen fason yaptırırım.
- Açık kaynak yazılım üretirsem kar edemem.

İyi Yazılımın Özellikleri

- Kabul edilebilirlik; Kullanıcı tarafından ve kullanıcının kullandığı diğer sistemlerle entegrasyonu
- Güvenilebilirlik ve güvenlik
- Verimlilik; Sistem kaynaklarını kullanım açısından
- Bakım kolaylığı

GEREKSİNİM ANALİZİ BAŞARILI BİR PROJENİN
TEMELİDİR

GEREKSİNİM ANALİZİNDE YAPILACAK HATA EN
MALİYETLİ HATALARDAN BİRİDİR

- Çok departmanlı kuruluş için tasarlanacak yazılımda taleplerin yönetilmesi için süreç, talep yöneticisi tarafından başlatılır.
- Taleplerin gereksinime dönüştürülmesi aşaması İş Analistleri tarafından yapılır.
- İş analistleri taleplerin yapılabilirliğinin olup olmadığına yani **olurluluk çalışması** ile başlar.

- Duruma sistematik yaklaşılması gerektiğinden Gereksinim Mühendisliği (Requirements Engineering) olarak da adlandırılır.
- Sistemden beklenti ne?
- Sistemin kısıtları ne?
- Sistemin başında yapıldığı için tüm sistem etkilenir.

- Sistem geniş açıdan değerlendirilmeli.
- İstekler ile yazılım dinamikleri, sınırları iyi irdelenmeli.
- Tüm gereksinimler yazılım aşamasında önce belirlenmese de yazılım genelde bir takım işidir ve yazılımcıların temel gereksinimleri bilmesi çok önemlidir.
- Yazılım gereksinimleri test mühendislerinin referans alabileceği ve testlerin tasarlanmasında kullanılan bir girdi olarak düşünülebilir. Gereksinim doğru tanımlanmış, net açık ve anlaşılır olduğunda, yazılım mühendisleri ve test mühendislerinin işleri kolaylaşacaktır.

Gereksinimler Eksik ya da Yanlış Analiz Edilirse?

- Verimlilik düşer
- Kaynak maliyeti artar
- Süre maliyeti artar
- Müşteri memnuniyetsizliği

Gereksinim Tipleri

- Kullanıcı gereksinimleri: Kullanıcı için tasarlanan akış şemaları, doğal dille, tablolarla oluşturulabilir. Kullanıcı senaryoları için (use-case) için temeldir. Teknik detayı çok barındırmaz. Gereksinimler tam anlatılamazsa, birkaç ihtiyaç tek bir ihtiyaç gibi anlatılmak istenirse gibi sebeplerden dolayı problem yaşanabilir.
- Sistem gereksinimleri: Sistem servislerinin detaylı tanımlanması ile donanımsal yapılara özgü gereksinimlerdir. Yüksek seviyeli bir dil yada gereksinimler için tasarlanmış özel bir dil ile tanımlanmalıdır.

Etki Alanlarına Göre Gereksinimler

- İşlevsel gereksinimler: Sistemin çalışmasını doğrudan etkileyen, sistemin içinde bir fonksiyona sahip olan gereksinimlerdir. Sistemin ne yapacağını yapısal ve işlevsel olarak ortaya koyarlar. Bunlar olmazsa sistem çalışmayacaktır. Sistemin neler yapacağını soyut olarak değil de detaylandırılmış biçimde belirler. Örneğin bir otel rezervasyon sisteminin ayrıntıları.
- İşlevsel olmayan gereksinimler: Sistemin çalışmasını etkilemez. Sistemin daha çok kısıtları ile fiziksel ortam, ara yüzler, kullanıcı odaklı olma, güvenlik, güvenilirlik, kalite güvence gibi soyut niteliklerini belirleyen gereksinimlerdir. Yazılımlara işlevsellik katmamasına rağmen bu tip gereksinimler özellikle yazılım kalitesi açısından kritik rol oynarlar. Bu gereksinimler yazılımda karşılanmadığı sürece yazılımın kullanılabilirliği yetersiz kalacaktır. Örneğin bellek ve performans beklentisi, uygulama dili, arayüz rengi, tarih bilgisinin nerede olacağı vb.

- İşlevsel Gereksinimler;
 - Sistemin girdileri nelerdir?
 - Sistemin çıktıları nelerdir?
 - Veriler ne yapılacak işlenecek mi, tutulacak mı?
 - Sistemin öncesinde ve sonrasında neler çalışacak?

- İşlevsel olmayan gereksinimler;
 - Kullanılabilirlik, verimlilik, performans, tepki süresi, sürdürülebilirlik, yeniden kullanılabilirlik vb.
 - Sistemde kullanılacak platformlar, teknolojiler vb.
 - Sistemin çalışması ile ilgili kullanılacak metodolojiler, masraflar ve teslimat tarihi vb.
 - Organizasyon eğitimleri, standartları vb.
 - Kanuni kısıtlar vb.

Gereksinim Zorlukları

- Uygulama alanına yabancı olmak
- Uygulama alanının kompleks olması
- Tüm sistemi bir kişinin bilmesi
- Müşterilerin bilgileri ortaya koyamaması
- Gereksinimlerin tutarsız ve eksik olması
- Gereksinimler farklı açılardan elde edilmesinde aynı durum için uyumsuzluk ve tutarsızlık oluşabilir
- Analist-müşteri ve analist-geliştirici arasında yanlış anlaşılmalarda

Gereksinim Amaçları

- Geliştiricilere, müşterilerin sistemi nasıl istedikleri anlatılır.
- Tasarımcılara, hangi fonksiyon ve özelliklerin kullanılacağını belirler.
- Test takımına, müşteri isteklerini gerçekleştirmeyi gösterir.

Gereksinim Özellikleri

- Doğru
- Çelişkisiz
- Bütün
- Gerçekçi,
- Mevcut donanım ile sürdürülebilir olması
- Müşterinin isteğini yansıtması
- Doğrulanabilir olması.

Gereksinim Süreçleri

- Olurluk (fizibilite) çalışması
- Gereksinim analizi
- Onaylama ve doğrulama
- Gereksinim yönetimi

Olurluk (Fizibilite) Çalışması

- Müşteri kim?
- Alan ne?
- Faydaları ne?
- Uygun bütçe ve teknoloji mevcut mu?
- Ek hizmetler için müşteri maliyeti karşılayabilecek mi?
- Tüm çalışma için olurluk (teknik, ekonomik, yasal) çalışması yapılmalı
- Fizibilite raporu hazırlanmalı
- Karlılık-maliyet analizi yapılmalı
- Risk ve öncelik değerleri değerlendirilmeli

Gereksinim Analizi

- Tüm paydaşların sistemden beklentileri bir metodoloji ile ortaya konmalı.
- Çatışma yönetimi yapılmalı ve tutarsızlıklar çözümlenmeli
- Gereksinimler önceliklendirilmeli
- Gereksinimler müşterilerin anlayabileceği şekilde tanımlanmalı
- Ayrıntılı gereksinim belirtimi ile ortak olan ve olmayan noktalar ortaya konulmalı. Yazılım tasarımına giriş yapılır. Yazılım tasarımındaki nesneler için soyut tanımlar yapılır ve üst düzey diyagramlar oluşturulur.
- Küçük bir prototip oluşturulur.

Onaylama ve Doğrulama

- Müşteri onayı olmayan gereksinimler projeye dahil edilmemeli.
- Tüm gereksinimler resmi olarak onaylatılmalı.
- Gereksinimleri doğrulamak için kontrol listeleri oluşturulmalı.

Gereksinim Yönetimi

- İş, organizasyon ve teknik değişikliklerin kontrolü ve yönetimi sürecidir.

	(E=Evet, H=Hayır, U=Uygulanamaz)	
	(E / H / U)	Açıklamalar
Standartlara Uygunluk		
Doküman için standartlar/kılavuzlar ve isimlendirme kuralları belirlenmiş mi?		
Doküman formatı tanımlı standart/kılavuz ile uyumlu mu?		
Doküman boyunca standartlar ve isimlendirme kuralları takip edilmiş mi?		
Doküman İçeriği		
Üst düzey sistem genel bakışı var mı?		
Üst düzey sistem diyagramları iç ve dış arayüzleri ve veri akışlarını gösteriyor mu?		
Sistemin fonksiyonel akışı açıkça ve tam olarak tanımlanmış mı?		
Yazılım çevresi (ör, donanım, yazılım kaynakları, kullanıcılar) tanımlanmış mı?		
Tüm referans dokümanları listelenmiş mi?		
Tüm tanımlamalar, kısaltmalar ve kısaltmalar konmuş mu?		
Kullanıcı karakteristikleri tanımlanmış mı?		
Genel tasarım ve uygulama kısıtları not edilmiş mi?		
Uygulamayı etkileyen genel varsayımlar belirtilmiş mi?		
Hafıza gereksinimleri sağlanmış mı?		
Zamanlama ve hafıza limitleri donanım kısıtlarıyla uyumlu mu?		
Yazılım performansı üzerindeki tüm limitler ve sınırlamalar tanımlanmış mı?		
Her bir fonksiyon ayrı ayrı tanımlanmış mı?		
Her bir fonksiyon için herhangi yazılım sınırlamaları tartışılmış mı?		
Emniyet-kritik yazılım gereksinimleri tanımlanmış mı?		
Güvenlik gereksinimleri tanımlanmış mı?		
Gizlilik gereksinimleri tanımlanmış mı?		
Yazılım kalite gereksinimleri (güvenilirlik, taşınabilirlik, tekrar kullanabilirlik, bakılabilirlik) tanımlanmış mı?		
Çevresel gereksinimler ve durumlar tanımlanmış mı?		
Tüm paketleme gereksinimleri tanımlanmış mı?		
Tüm teslim gereksinimleri tanımlanmış mı?		
İşlemsel bilgisayar donanımı için gereksinimler sağlanmış mı?		
Bilgisayar yazılım kaynakları (işletim sistemi, ağ yazılımı, veritabanları, test yazılımı) tanımlanmış mı?		
Tam bütünleştirme, test ve kabul kriterleri belirlenmiş mi?		
Gereksinimler için test metotları (test, gösterim, analiz veya denetleme) tanımlanmış mı?		
Fonksiyonel gereksinimler özgün numaralandırılmış mı?		
Diğer ana fonksiyonlara veya dış varlıklara olan arayüz gereksinimleri açıkça tanımlanmış mı?		
Her bir yazılım gereksinimi bir veya daha fazla üst düzey sistem gereksinimine ilelenebilir mi?		

Gereksinim Dokümanları

- Sistem tasarımcısı için gereksinimlerin ayrıntıları ile belirlendiği doküman
- İşlevsel ve işlevsel olmayan gereksinimler için mümkünse ayrı ayrı doküman
- Test engineer için dokümanlar izlenebilir olmalı

Yazılım Geliştirme Ortamları

- Programlama dili
- Veritabanı
- CASE (Computer Aided Software Engineering - Bilgisayar Destekli Yazılım Mühendisliği) araçları

Kaynakça

- Ian Sommerville - Yazılım Mühendisliğine Giriş
- <https://slideplayer.biz.tr/slide/2786995/MeralErtugrul>
- <https://www.mobilhanem.com/yazilim-gereksinim-analizi/>
- <bilgisayarkavramlari.com/SadiEvrenSeker>
- Gül, Z. (2006). Yazılım geliştirme sürecinin iyileştirilmesi ve Türkiye uygulamaları, İstanbul Teknik Üniversitesi, (Doctoral dissertation, Fen Bilimleri Enstitüsü)
- Dr. Öğr. Üyesi Yunus Emre SELÇUK'un ders notları