

Gereksinimlerin Doğrulanması Değişimi ve Yönetimi

- Müşterinin gerçekten istediği sistemi tanımlayan gereksinimlerin kontrol edilmesi sürecidir.
- Gereksinim problemlerinin bulunması ile ilgilendiği için açığa çıkarma ve analiz ile örtüşür.
- Gereksinimlerin hata maliyetleri yüksektir, bu nedenle onaylama çok önemlidir.
- Sistem değişikliği yaparak bir gereksinim probleminin düzeltme maliyeti tasarım ve kodlama maliyetinin düzeltilmesinden daha fazladır.
- Gereksinimlerde değişiklik, genellikle sistem tasarımı ve gerçekleştiriminin de değiştirilmesi gerektiği anlamına gelir.
 - Teslimattan sonra bir gereksinim hatasını düzeltmek, bir uygulama hatasını düzeltme maliyetinin 100 katına kadar mal olabilir.

Verification (Doğrulama) ve Validation (Geçerleme)

- Bu iki kavram gereksinim mühendisliğinin kalbidir.
- **Verification:** "Ürünü doğru inşa ediyor muyuz?"
(Gereksinim dökümanına, standartlara uyuyor mu?)
- **Validation:** "Doğru ürünü inşa ediyor muyuz?"
(Müşterinin gerçek ihtiyacını karşılıyor mu?)

Gereksinimleri Kontrol

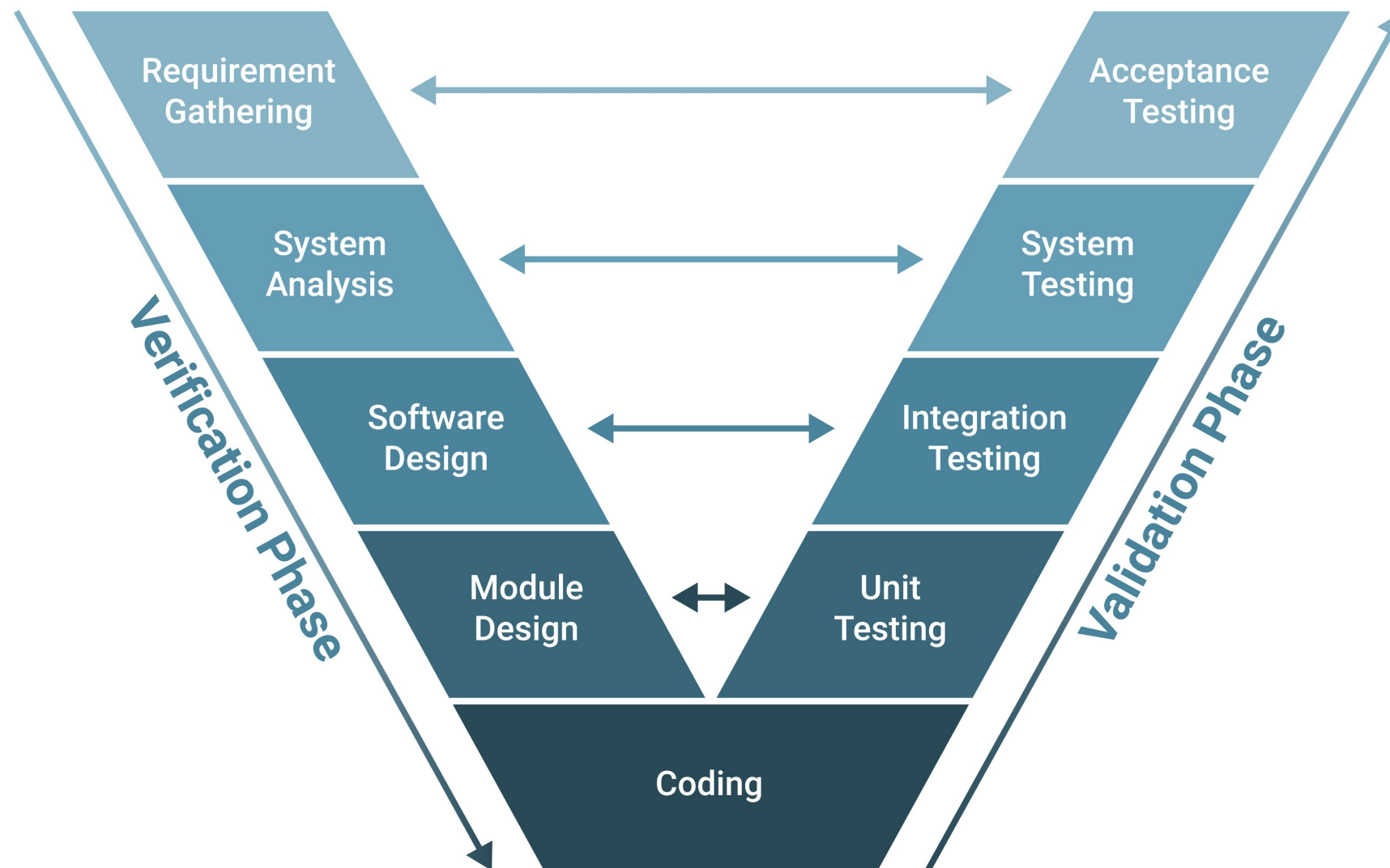
- Geçerlik Kontrolleri: Sistem, müşterinin ihtiyaçlarını en iyi şekilde destekleyen işlevleri sağlıyor mu?
- Tutarlılık Kontrolleri: Herhangi bir gereksinim çatışması var mı?
- Tamlık Kontrolleri: Müşterinin istediği tüm fonksiyonları ve kısıtlamaları içeriyor mu?
- Gerçekçilik Kontrolleri: Mevcut bütçe ve teknoloji göz önüne alındığında gereksinimler uygulanabilir mi? Zaman yeterli mi?
- Doğrulanabilirlik Kontrolleri: Gereksinimler kontrol edilebilir mi?

Gereksinimleri Doğrulama Teknikleri

- Gereksinim incelemeleri: Gereksinimlerin inceleme takımı tarafından sistematik manuel analizi.
- Prototipleme: Gereksinimleri kontrol etmek için sistemin yürütülebilir bir modelini kullanmak.
- Test takımı oluşturma: Gereksinimleri kontrol etmek için testler geliştirmek.

- Test incelemeleri;
 - Doğrulanabilirlik: Gereksinim gerçekçi bir şekilde test edilebilir mi?
 - Anlaşılabilirlik: Gereksinim doğru bir şekilde anlaşıldı mı?
 - İzlenebilirlik: Gereksinimin kaynağı açıkça belirtilmiş mi?
 - Uyarlanabilirlik: Gereksinim, diğer gereksinimler üzerinde büyük bir etki olmadan değiştirilebilir mi?

Doğrulama ve Geçerleme için V Modeli



- Kullanıcı Gereksinimleri <—> Kabul Testleri
- **Sol Taraf (Ne isteniyor?):** Müşteri ile görüşülür, "Sistem ne yapmalı?" sorusu cevaplanır. (Kullanıcı İhtiyaçları Dökümanı - URS).
- **Sağ Taraf (İstenen oldu mu?):** Yazılımın müşteri ihtiyaçlarını karşılayıp karşılamadığı test edilir.
- Kabul testleri teknik hataları (bug) bulmak için değil, iş akışının doğruluğunu kanıtlamak içindir.

- Sistem Gereksinimleri <—> Sistem Testleri
- **Sol Taraf (Sistem nasıl davranmalı?):** Kullanıcı gereksinimleri teknik dille ifade edilir. Fonksiyonel ve Fonksiyonel Olmayan gereksinimler belirlenir.
- **Sağ Taraf (Sistem bütünüyle çalışıyor mu?):** Yazılımın bir bütün olarak teknik spesifikasyonlara uyup uymadığı test edilir.
- Performans, güvenlik, stres ve yük testleri burada yapılır.

- Mimari Tasarım (High-Level Design) <—> Entegrasyon Testleri
- **Sol Taraf (Parçalar nasıl haberleşecek?):** Sistemin modülleri, veritabanı yapısı, API'ler ve arayüzlerin genel haritası çıkarılır.
- **Sağ Taraf (Parçalar uyumlu mu?):** Farklı modüllerin birbirleriyle doğru konuşup konuşmadığı test edilir.
- Hatalar genellikle modüllerin içinde değil, modüllerin arasındaki veri transferindedir ("Interface" hataları).

- Detay Tasarım (Low-Level Design) $\longleftrightarrow$ Birim Testleri (Unit Test)
- **Sol Taraf (İç mantık nasıl olacak?):** Her bir modülün içindeki algoritmalar, sınıflar (classes), metodlar ve veri tipleri tasarlanır.
- **Sağ Taraf (En küçük parça çalışıyor mu?):** Kodun en küçük birimlerinin (fonksiyon/metod) tek başına doğru çalışıp çalışmadığı test edilir.
- Bu testleri genellikle yazılımcılar yapar (White-box testing).

- V-Modeli'nin Waterfall (Şelale) yaklaşımı varken, Agile süreçlerde ise "Kabul Kriterleri" (Acceptance Criteria) üzerinden testler (TDD/BDD) koşulmaktadır.

Gereksinim İzlenebilirliği (Traceability)

- **İleriye Doğru İzlenebilirlik:** Her gereksinimin bir test senaryosu var mı? (Kapsama Analizi).
- **Geriye Doğru İzlenebilirlik:** Bu test senaryosu hangi gereksinimi test ediyor? (Gereksiz kodun önlenmesi).
- **İzlenebilirlik Matrisi (Traceability Matrix - RTM)** gösterimi. Satırlarda gereksinim ID'leri, sütunlarda Test Case ID'leri.

Gereksinimlerin Test Edilmesi

- Fonksiyonel Gereksinimlerin Test Edilmesi:
 - Gereksinimlerin "Test Edilebilir" (Testable) yazılmasının önemlidir. "Sistem, kullanıcı hatalarını engellemelidir." (Yanlış) vs. "Kullanıcı, 'Tarih' alanına bugünden daha eski bir tarih girdiğinde, sistem 'Geçmişe dönük işlem yapılamaz' hata mesajını göstermeli ve 'Kaydet' butonunu pasif hale getirmelidir." (Doğru).
 - **Black-Box Testing (Kara Kutu Testi):** Kodun içine bakmadan sadece gereksinim girdileri ve beklenen çıktılar üzerinden test yapılmasıdır. Sistemin nasıl çalıştığını bilinmez, sadece ne yapması gerektiğiyle (Gereksinimlerle) ilgilenir. Test Uzmanları ve Son Kullanıcılar tarafından yapılır.
 - **Senaryo Bazlı Test:** Use-Case'lerin (Kullanım Durumları) test senaryolarına dönüştürülmesi.

- Fonksiyonel Olmayan Gereksinimlerin Test Edilmesi:
 - "Sistem hızlı çalışmalıdır." (Yanlış) vs. "Sistem, 1000 eşzamanlı kullanıcı altındayken, yapılan ürün sorgusuna en geç 2 saniye içinde yanıt vermelidir." (Doğru).
- **Performans Gereksinimleri:** Yük ve Stress testleri.
- **Güvenlik Gereksinimleri:** Penetrasyon testleri (Örn: Yetki matrisinin kontrolü).
- **Kullanılabilirlik Gereksinimleri:** Arayüz testleri ve UX analizleri.

- Kullanıcı Kabul Testleri ve Gereksinimler:
 - Kullanıcı Kabul Testleri, geliştiricinin "bitti" demesi değil, gereksinim sahibinin (müşterinin) "istediğim bu" demesidir.
 - Kullanıcı Kabul Testleri, senaryoları doğrudan iş gereksinimlerinden (Business Requirements) türetilir.
 - **Beta Testi:** Gerçek ortamda, gerçek verilerle gereksinimlerin doğrulanması.

- Birim (Unit) testleri geçen bir yazılım, müşteri önüne çıkınca çalışmayabilir. Çünkü birim testler parçaları kontrol eder, senaryo testleri ise parçaların birlikte çalışıp çalışmadığını ve iş akışının mantığını kontrol eder.

- Gereksinim İncelemeleri;
 - Gereksinim tanımı formüle edilirken düzenli gözden geçirmeler yapılmalıdır.
 - İncelemelere hem müşteri hem de yüklenici personeli dahil edilmelidir.
 - İncelemeler resmi (tamamlanmış dökümanlarla) veya gayri resmi olabilir.
 - Geliştiriciler, müşteriler ve kullanıcılar arasındaki iyi iletişim sorunları erken bir aşamada çözebilir.

Gereksinimlerin Bakım, Değişiklik Yönetimi ve Etki Analizi

- Problemler tam olarak tanımlanamadıkları için gereksinimlerin eksik olması kaçınılmazdır.
- Ayrıca iş ortamındaki değişiklikler sebebiyle sistem gereksinimleri değişebilir.

- Sistemin iş ve teknik ortamı kurulumdan sonra her zaman değişir.
- Sistem için ödeme yapan ile sistem kullanıcıları nadiren aynı kişilerdir.
- Büyük sistemlerde farklı gereksinimleri olan paydaşlar vardır.

- Bakım Türleri ve Gereksinim İlişkisi
 - **Düzeltilici (Corrective):** Hataların düzeltilmesi. (Aslında karşılanmamış bir gereksinimin tamiri).
 - **Uyarlayıcı (Adaptive):** Çevre değişikliği (Örn: Yeni Android sürümü, yeni vergi kanunu). Bu durum mevcut gereksinimlerin güncellenmesini gerektirir.
 - **Mükemmelleştirici (Perfective):** Yeni özellikler eklenmesi. Yeni gereksinimlerin analizi.
 - **Önleyici (Preventive):** Gelecekteki sorunları önleme (Refactoring).

- Değişiklik Yönetimi (Change Management)
 - Yazılım yaşar ve değişir. Müşteri "Şu butonu şuraya alalım" dediğinde süreç nasıl işler?
 - **Scope Creep (Kapsam Kayması):** Kontrolsüz gereksinim değişikliklerinin projeyi batırması.
 - Her değişiklik talebi (Change Request - CR), yeni bir gereksinim gibi ele alınmalı, analiz edilmeli ve onaylanmalıdır.

- Etki Analizi (Impact Analysis)
 - Soru: "Gereksinim A'yı değiştirirsem, sistemin başka neresi bozulur?"
 - İzlenebilirlik matrisinin burada devreye girmesi:
Gereksinim A —> Tasarım Modülü X —> Test Senaryosu Y.
 - Etki analizi yapılmadan yapılan bakım, "Yaparım" deyip sistemi başka yerden patlatmaktır (Regression).

Gereksinimlerin Yönetimi

- Gereksinimlerin yönetimi, gereksinim mühendisliği süreci ve sistem geliştirme sırasında değişen gereksinimleri yönetme sürecidir.
- Gereksinimler evrimleşirken gereksinim değişimlerinin etkisini değerlendirebilmek için tek tek gereksinimleri izlemek ve bağımlı gereksinimler arasında bağlaçlar oluşturmak gerekir.
- Gereksinimleri değiştirmek çok paydaşlı durumlarda gereksinim öncelikleri paydaşlar arasında sıkıntı olabileceğinden mümkünse bağımsız bir otorite tarafından karar verilmeli.

Gereksinim Yönetimi

Planlaması

- Gereksinimlerin tanımlanması: Her gereksinim, diğer gereksinimlerle çakışmaması için benzersiz bir şekilde tanımlanmalıdır.
- Bir değişim yönetim süreci: Bu, değişikliklerin etkisini ve maliyetini değerlendiren bir dizi faaliyettir.
- İzlenebilirlik kuralları: Bu politikalar, her gereksinim arasındaki ve gereksinimler ile sistem tasarımı arasındaki ilişkileri tanımlar.
- Araç (tool) desteği: Kullanılabilecek araçlar, uzman gereksinimler yönetimi sistemlerinden hesap tablolarına ve basit veritabanı sistemlerine kadar çeşitlilik gösterir.

Gereksinim Değişim Yönetimi

- Problem analizi ve değişim spesifikasyonu: Süreç gereksinim problemi ya da belli bir değişiklik önerisi ile başlar. Bu aşamada, sorunun veya değişiklik önerisinin geçerli olup olmadığını kontrol etmek için analiz edilir. Bu analiz, daha özel bir gereksinim değişikliği teklifine yanıt verebilecek veya talebin geri çekilmesine karar verebilecek.
- Değişim analizi ve maliyetlendirme: Önerilen değişikliğin etkisi, izlenebilirlik bilgileri ve sistem gereksinimlerine ilişkin genel bilgiler kullanılarak değerlendirilir. Bu analiz tamamlandıktan sonra, gereksinim değişikliğine devam edip etmeme kararı verilir.
- Değişim gerçekleştirimi: Gereksinimler dokümanı ve gerektiğinde sistem tasarımı ve uygulaması değiştirilir. İdeal olarak, dökümanın, değişikliklerin kolayca uygulanabilmesi için organize edilmesi gerekir.

Kaynakça

- Ian Sommerville, Yazılım Mühendisliği
- Doç. Dr. Sedat AKLEYLEK'in ders notları
- <http://www.kriptarium.com/gereksinim.html>
- Bilkent Cyberpark Webinar