

Erciyes Üniversitesi
Bilgisayar Mühendisliği Bölümü

BZ 313 Yazılım Mühendisliği

6. Gereksinim Analizi

Gereksinimler

Gereksinimler, sistemin işlevini müşterinin bakış açısından tanımlar.

- Gereksinimler, sistemin işlevselliğini, kısıtlamalarını ve hedeflerini belirler.
- Gereksinimler hem müşteri hem de geliştirme personeli tarafından anlaşılabilir olmalıdır.

Geliştirme ekibi ve müşteri, gereksinimleri tanımlamak için birlikte çalışmalıdır.

Gereksinimler Neden Önemlidir?

Gereksinimlerle ilgili olan alanlar kırmızı işaretlenmiştir.

Başarısız yazılım projelerinin nedenleri

Eksik gereksinimler	13.1%
Kullanıcı katılımı eksikliği	12.4%
Kaynak eksikliği	10.6%
Gerçekçi olmayan beklentiler	9.9%
Yönetici desteği eksikliği	9.3%
Değişen gereksinimler ve özellikler	8.8%
Planlama eksikliği	8.1%
Sisteme artık gerek olmaması	7.5%

Gereksinimlerin anlaşılmaması, geliştiricilerin yanlış sistem oluşturmalarına neden oldu.

Kaynak: Standish Group

Gereksinimleri Tanımlama Adımları

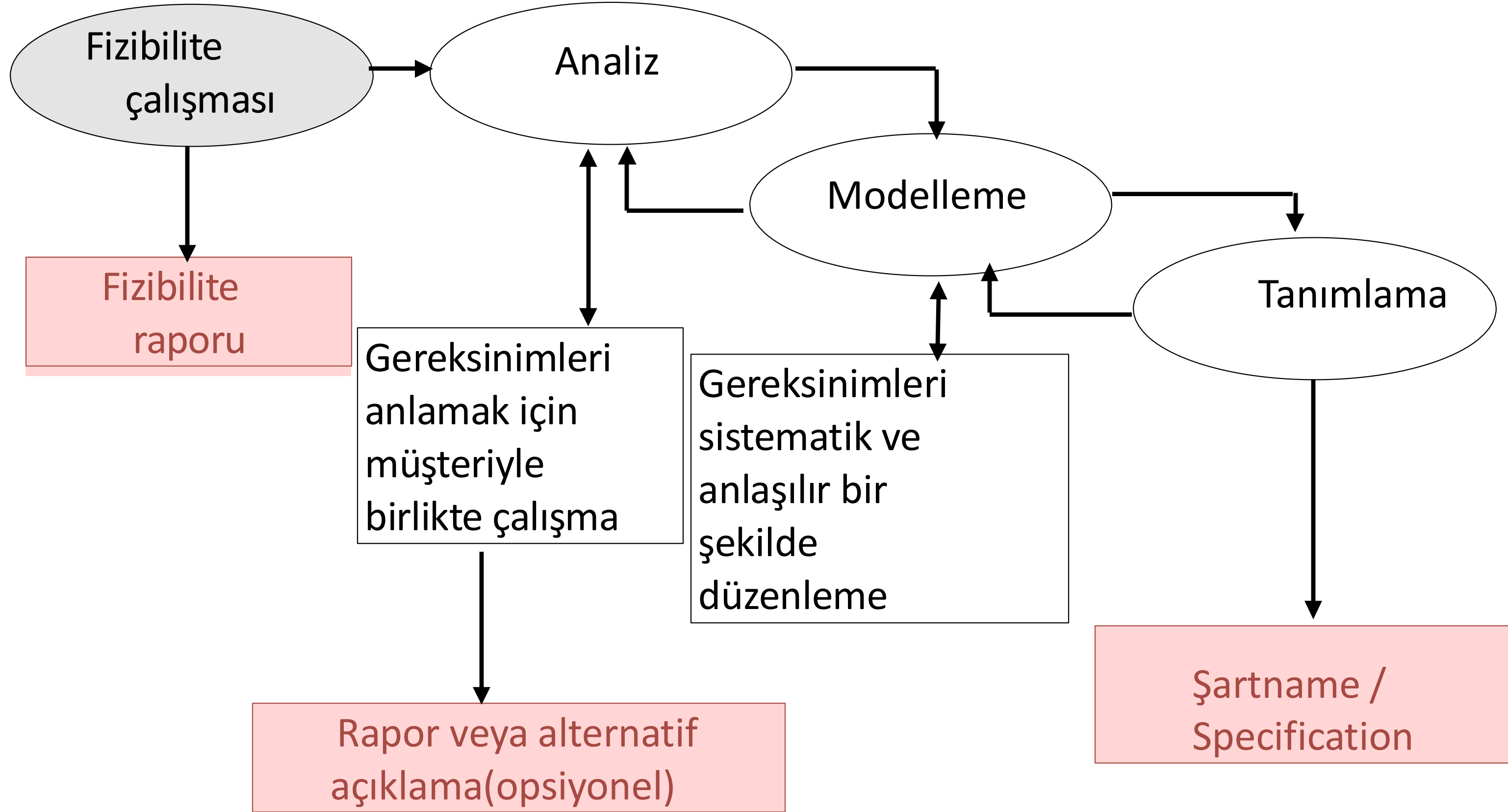
Gereksinimlerin tanımlanması birkaç adımda ele alınabilir:

- Client, customer ve kullanıcılarla istişare ederek işlevselliği oluşturmak için **analiz**.
- Gereksinimleri sistematik ve anlaşılır bir şekilde organize etmek için **modelleme**.
- Gereksinimlerin **tanımlanıp, kayıt altına alınıp ve iletilmesi**.

Heavyweight süreçlerde tasarıma başlamadan önce bütün bu adımlar gerçekleştirilir.

Lightweight süreçlerde, bu adımlar her sprint için ayrı ayrı yapılır.

Gereksinim Adımları



Heavyweight ve lightweight arasındaki en büyük fark ara raporların olup olmadığı ve spesifikasyonda ne kadar ayrıntı bulunduğu.

Gereksinimlerde Yaşanan İkilemler

Ne yapmanız gerektiğini bilmeden bir sistem oluşturamazsınız.

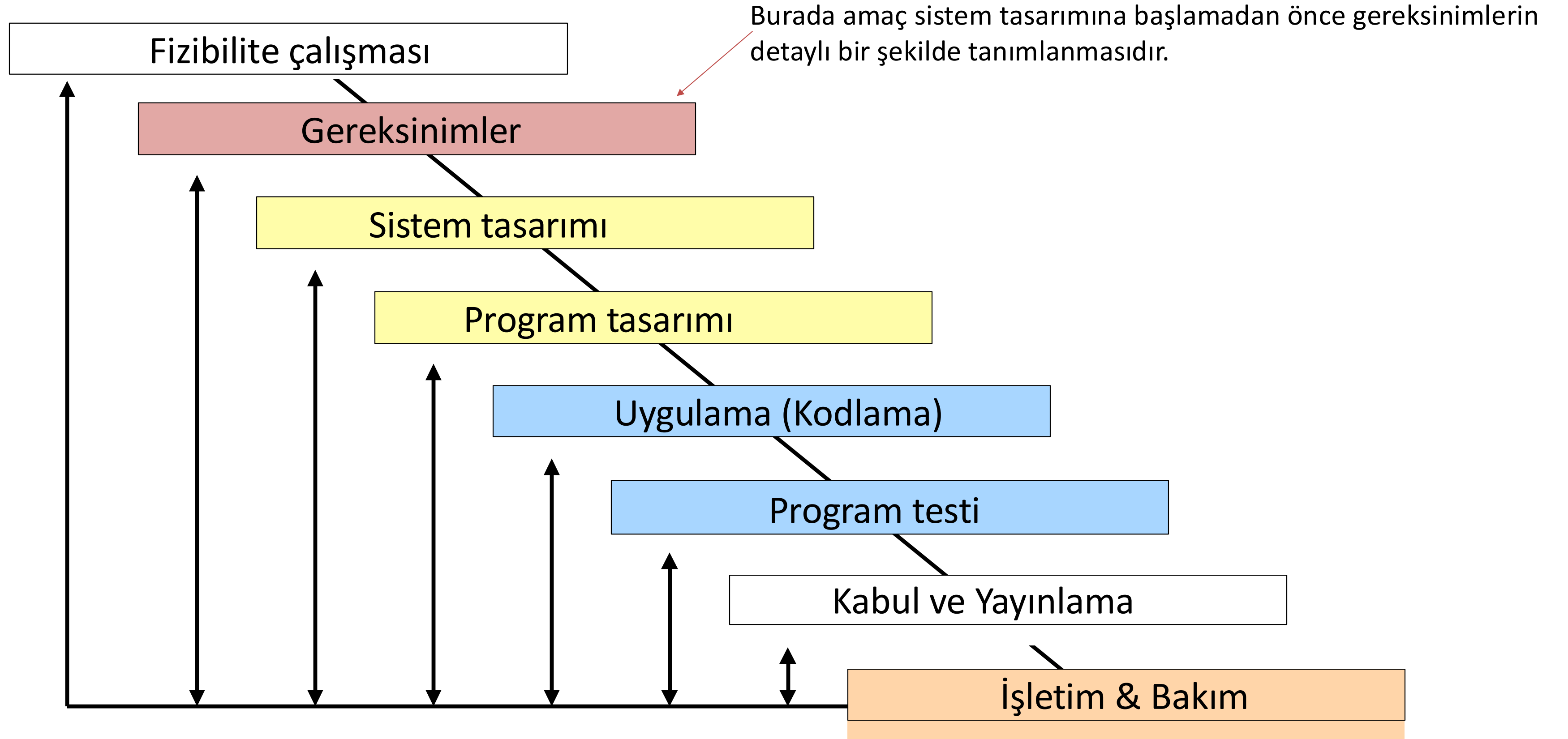
ANCAK...

Müşteriler, gereksinimleri yalnızca kısmi olarak anlayabilir durumdadır.

- Sistemi gördüklerinde yeni özellikler isteyebilirler.
- Çoğu zaman da bu değişiklikler oldukça büyük olur.
- Bu değişiklikleri yapabilmeniz için sistemin büyük bir bölümüne tekrar çalışmanız gerekebilir.

Bu hem **heavyweight** hem de **lightweight** süreçler için ortak problemlerdir.

Heavyweight Süreçler: Modifiye Şelale Modeli



Heavyweight Süreçlerde Gereksinimler

Heavyweight süreçlerde ayrıntılı spesifikasyonlar beklenir

- Her gereksinimi ayrıntılı olarak belirten yazılı belgedir.
- Belge Müşteri ve geliştiriciler tarafından dikkatlice kontrol edilir.
- Sözleşme belgesi olabilir.
- Kabul testi için kullanılacaktır.

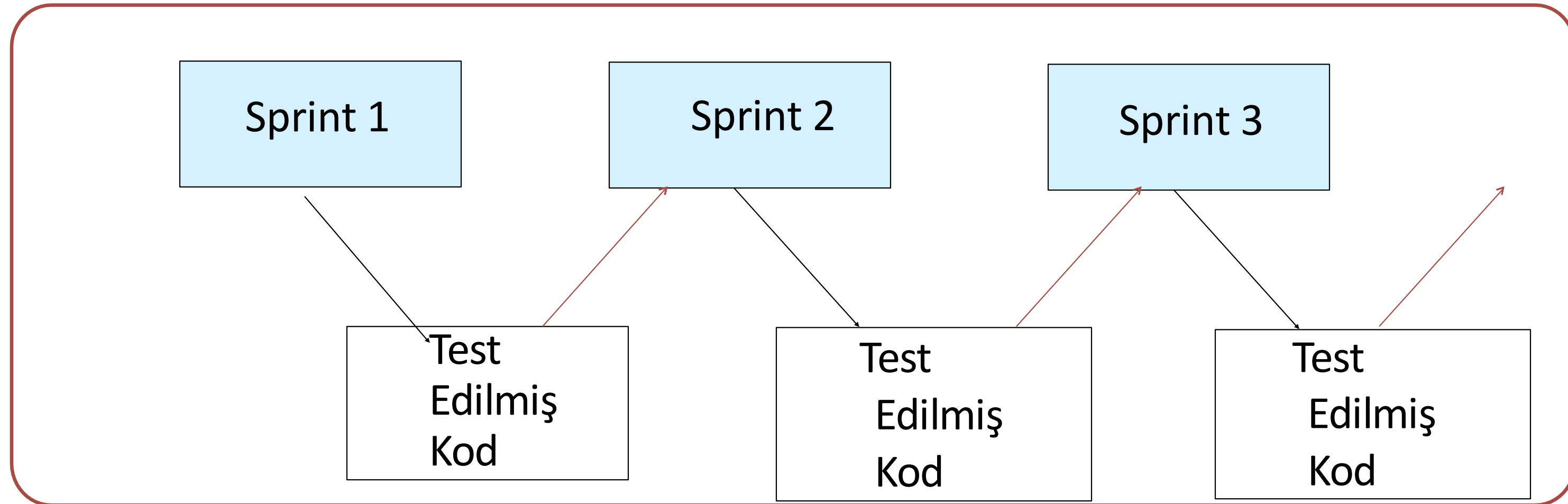
Zorluklar

- Spesifikasyon zaman alıcıdır ve oluşturulması zordur.
- Spesifikasyonun bakımı zordur
- Ayrıntılı bir spesifikasyonu kontrol etmek sıkıcıdır.
- **Müşteriler bu süreçleri çoğu zaman anlamlandıramaz.**

Ayrıntılı bir gereksinim spesifikasyonu (belirtim) oluşturmanın ve sürdürmenin zorluğu, birçok firmanın lightweight süreçleri tercih etmesinin nedenlerinden biridir.

Lightweight Süreçler: Çevik Geliştirme

Her sprint'in kendi
gereksinimleri vardır.



Lightweight Süreçlerde Gereksinimler

Lightweight süreçlerde gereksinimler her sprintte bir sefer geliştirilir.

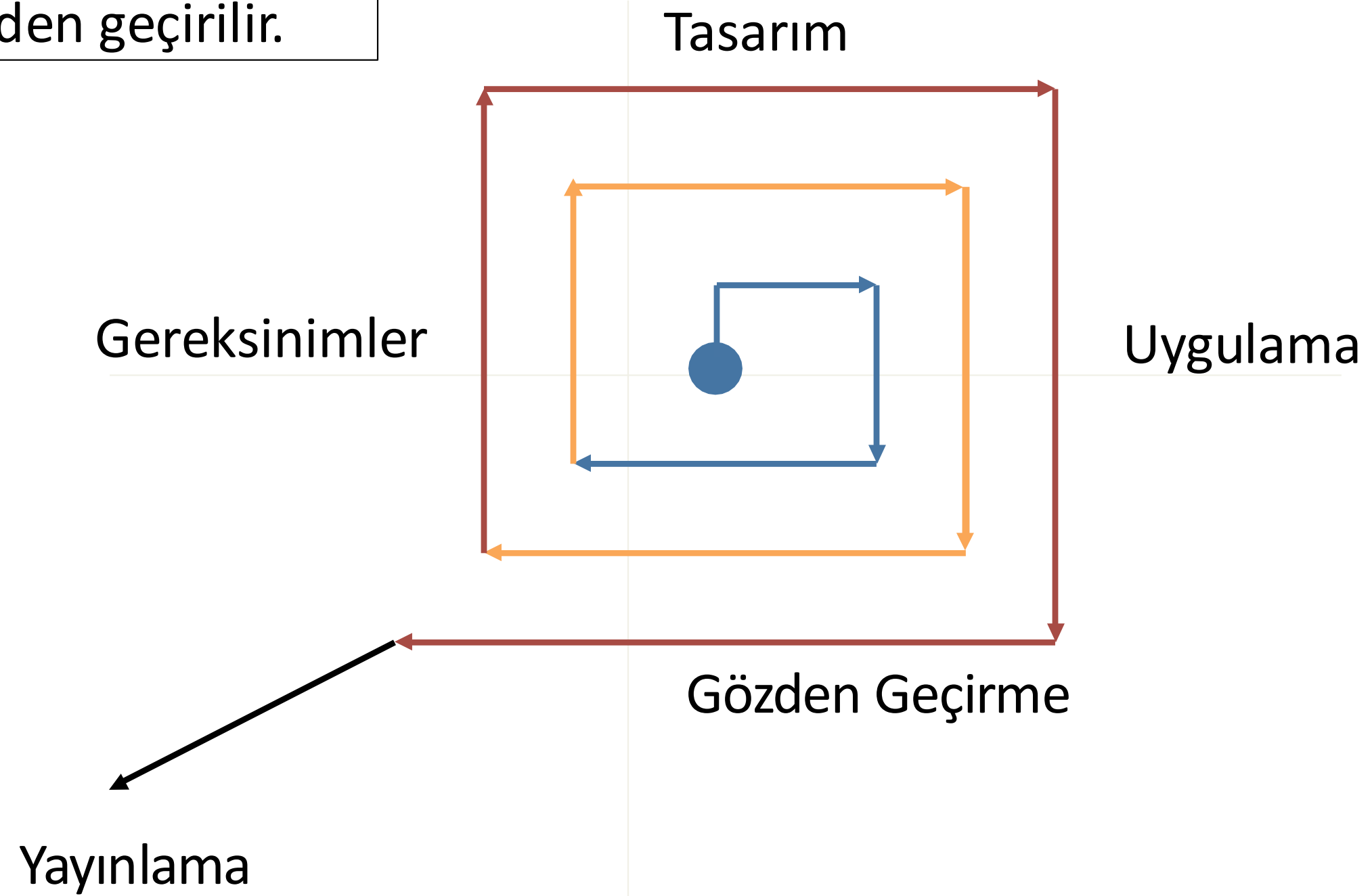
- Gereksinimleri kontrol etmek için çalışır kod kullanılır.
- Müşteri ve geliştiriciler gereksinimler üzerinde birlikte çalışır.
- Sprint sırasında minimum dokümantasyon oluşturulur.
- Gelecekte uygulamaya bakım uygulayacaklar için daha eksiksiz belgelere ihtiyaç vardır, ancak ayrıntılar kodda verilmiştir.

Zorluklar

- Bazı gereksinimler sistem genelinde tanımlanmalıdır ve tek bir sprint içinde tanımlanamaz, örneğin veri tabanları, güvenlik mimarileri, genel kullanıcı arayüzü tasarımı.
- Yeni bir sprinte başladığınızda önceki sprintlerdeki gereksinimlerin eksik veya yetersiz olduğunu görebilirsiniz. Bu yüzden gelecekteki sprintlerin gereksinimleri, önceki sprintlerin büyük ölçüde yeniden düzenlenmesine yol açabilir.

Middleweight Süreçler: Yinelemeli İyileştirme

Gereksinimler her yinelemede gözden geçirilir.



Middleweight Süreçlerde Gereksinimler

Middleweight süreçlerde gereksinimler yinelemeli olarak geliştirilir.

- İlk yinelemede ana gereksinimlerin bir taslağı oluşturulur.
- Her yinelemede taslak daha da iyileştirilir ve ayrıntılar eklenir.
- Gelecekte bakım yapacak kişiler için dokümantasyon gereklidir ancak ayrıntılar kodda verilmiştir.

Zorluklar

- Her yinelemede önceki çalışmaların büyük ölçüde tekrar incelenmesine neden olabilir. Bu yüzden yapılacak toplam iş miktarı önemlidir.
- Geliştiriciler genellikle sistemi bir bütün olarak düşünmek yerine önceki geliştirmelere yeni eklemelerde bulunmaya eğilimlidir.

Tartışma

Büyük bir sistem oluşturmak için olabildiği kadar esnek olmalısınız. Hem heavyweight, hem de lightweight süreçler çeşitli problemler barındırır.

ANCAK...

Her iki süreç de mantıklı kullanılırsa oldukça verim alınabilir.

- Modifiye şelale modeli gibi heavyweight bir süreç kullanılırken, gereksinimleri orta düzeyde ayrıntılı olarak belirleyin ancak çeşitli revizyonlara da açık olun. Bazı detaylar sürecin ilerleyen zamanlarına bırakılabilir.
- Çevik gibi lightweight bir süreç kullanılırken, sistem genelinde gereksinimleri ve genel sistem mimarisini sürecin başlarında, belki de normal sprint'lere başlamadan önce geliştirin.

Gereksinimlerde Ulaşılması Gereken Hedefler

- Gereksinimleri **anlarken** doğru ayrıntılara **odaklanın**.
- Müşterinin ve geliştiricilerin **gereksinimleri** ve **bu gereksinimlerin doğracağı** sonuçları anladığından emin olun.
- Gereksinimleri müşteri için **açık bir şekilde tanımlayın**. Bu, yazılı bir spesifikasyon, prototip sistem veya başka bir iletişim şekli olabilir.
- Gereksinimleri, **sistemi tasarlayacak, uygulayacak** ve **bakımını** yapacak kişiler için **anlaşılır bir şekilde** tanımlayın.

BZ 313 projelerinizde, müşteri ile gereksinimleri doğrulamak için ilk gereksinimleri çıkardıktan sonra müşteriye bir sunum yapmalısınız:

"Gereksinimlerinize ilişkin anlayışımız şudur: ..."

Böylelikle müşteriye neyin gerekli olduğunu söylersiniz.

Gereksinim Belirleme Teknikleri

Brainstorming (Beyin Fırtınası): Bir grup insanın hızlı bir şekilde fikir üretmesi ve paylaşması tekniğidir. Çeşitli fikirler ortaya atılarak gereksinimlerin belirlenmesi sağlanır.

Interview (Görüşme): Gereksinimleri anlamak için kullanıcılarla veya paydaşlarla yapılan birebir görüşmelerdir. Yüz yüze ya da çevrimiçi olabilir.

Focus Groups (Odak Grupları): Belirli bir kullanıcı grubuyla yapılan tartışmalardır. Katılımcılar belirli bir konu veya işlev hakkında görüşlerini ve beklentilerini ifade ederler.

Benchmarking (Kıyaslama): Mevcut çözümleri veya rakip ürünleri inceleyerek onların güçlü ve zayıf yönlerini analiz etmektir. Bu analiz, yeni gereksinimlerin belirlenmesine yardımcı olur.

Questionnaire and Surveys (Anket ve Araştırmalar): Kullanıcıların ve paydaşların ihtiyaçlarını anlamak için soruların yer aldığı anketlerdir. Büyük kullanıcı grupları için uygundur.

Gereksinim Belirleme Teknikleri

Role-playing and Viewpoints (Rol Yapma ve Bakış Açıları): Farklı kullanıcı rollerinin temsil edilerek gereksinimlerin belirlenmesidir. Her rolün bakış açısını dikkate alarak daha kapsamlı gereksinimler toplanabilir.

Document Analysis (Doküman Analizi): Mevcut belgeleri inceleyerek gereksinimleri belirleme yöntemidir. Örneğin, mevcut sistem dokümantasyonları veya kullanım kılavuzları incelenir.

Observation (Gözlem): Kullanıcıların yazılımı nasıl kullandığını doğrudan gözlemleyerek gereksinim belirleme tekniğidir. Özellikle kullanıcı davranışlarının anlaşılmasında etkilidir.

Facilitated Workshops (Kolaylaştırılmış Atölye Çalışmaları): Farklı paydaşların katıldığı, bir moderatör tarafından yönlendirilen grup çalışmalarıdır. Fikirlerin paylaşılması ve gereksinimlerin belirlenmesi sağlanır.

Joint Application Development (JAD) (Ortak Uygulama Geliştirme): Geliştiriciler, kullanıcılar ve paydaşların bir araya gelerek ihtiyaçları ve beklentileri belirledikleri atölye çalışmalarıdır.

Gereksinim Belirleme Teknikleri

Prototypes (Prototipler): Yazılımın erken bir sürümünün hazırlanmasıdır. Kullanıcılar, prototip üzerinden geri bildirim vererek gereksinimlerin netleştirilmesine katkıda bulunurlar.

Persona (Kişilik Profili): Tipik kullanıcı profillerini temsil eden, ayrıntılı karakter tanımlamalarıdır. Kullanıcı davranışlarını ve gereksinimlerini anlamaya yardımcı olur.

Use Cases (Kullanım Senaryoları): Kullanıcının sistemle nasıl etkileşime geçtiğini tanımlayan senaryolardır. Sistemin ne yapması gerektiğini somut örneklerle anlatır.

User Story (Kullanıcı Hikayesi): Kullanıcı gereksinimlerini kısa ve öz şekilde anlatan cümlelerdir. Genellikle, “Bir kullanıcı olarak ... yapmak istiyorum çünkü ...” formatında ifade edilir.

Ethnography (Etnografi): Kullanıcıların doğal ortamlarında gözlemlenmesidir. Kullanıcıların gerçek dünyadaki davranışlarını ve ihtiyaçlarını anlamak için yapılır.

İhtiyaç Analizi: Müşterilerle Görüşmeler

Müşteri görüşmeleri, gereksinim analizinin kalbidir.

Müşteriler bolca belirsizlik içeren gereksinim kavramına sahip olabilir.

- Bolca zaman tanıyın.
- Müşteriyle görüşmeden önce hazırlanın ve neyi başarmaya çalıştığınızı net olarak belirleyin.
- Her şeyi not (tam not) olarak tutun. Ekip üyelerinden birini yazıcı olarak atayın.
- Anlamadıysanız, tekrar tekrar sonuna kadar araştırın.
- Duyduklarınızı tekrarlayın.

BZ313 projelerinizde, gereksinimleri analiz etmek üzere müşterinizle birkaç toplantı planlamanız gerekecektir.

Küçük grup toplantıları genellikle en etkili olanıdır.

Gereksinim Analizi: Gereksinimleri Anlayın

Gereksinimleri derinlemesine anlayın

- Domain anlama (genellikle geliştiriciler olarak müşterinin aşına olduğu domain'e hakim değiliz)

Örnek: Suyun yer altından evlere getirilmesi süreci

- Terminolojiyi anlama

Müşteriler genellikle özel terminoloji kullanır.
Anlamadıysanız, bir açıklama isteyin.

- Tüm paydaşların gerçek gereksinimlerinin anlaşılması

Müşteriler neye ihtiyaç duydukları konusunda net fikirlere sahip olmayabilir veya gereksinimleri açıkça ifade edemeyebilirler.

Soru sormaya devam edin, *“Neden işleri bu şekilde yapıyorsun?”*
“Bu gerekli mi?” “Alternatifler nelerdir?”

İhtiyaç Analizi: Yeni ve Eski Sistemler

Müşteriler genellikle eski sisteme o kadar aşinadırlar ki yeni bir sistemde ihtiyaç duyulmayan işlevlere sahip olduğunun farkına varamazlar.

Replacement (Değiştirme) sistem, mevcut bir sistemin yerini alacak bir sistemin oluşturulmasıdır ve bu tür sistemlerde meydana gelebilir.

Legacy (eski) sistem, değiştirilmeyen ancak genişletilen veya yeni bir sisteme arabirim oluşturması gereken mevcut bir sistemdir ve bu sistemlerde de meydana gelebilir.

İhtiyaç analizinde ayırt etmesi önemli olanlar:

- Eski sistemin yeni sistemde ihtiyaç duyulan özellikleri
- Eski sistemin yeni sistemde ihtiyaç duyulmayan özellikleri
- Önerilen yeni özellikler

Gereksinim Analizi: Konuşulmayan Gereksinimler

Konuşulmayan gereksinimleri keşfetmek, genellikle gereksinimleri geliştirmenin en zor kısmıdır çünkü genellikle organizasyonel dinamikleri içerir.

Örnekler:

- Değişime karşı direnç
- Departmanlar arası sürtüşme (Örneğin, personel transferi)
- Yönetimin güçlü ve zayıf yönleri

Paydaş Analizi

Paydaşları belirleyin. Bu sistemden etkilenecek kişileri belirlemeyi sağlar.

Bu sistemden kimler etkilenir?

- Müşteri (Client)
- Müşteri (Customer)
- Kullanıcı (birçok kategori)
- Üst yönetim
- Yöneticiler
- Bilgi işlem personeli

Örnek:

Web alışveriş sitesi (alışveriş yapanlar, yönetim, finans, depo)

BZ313 projenizde bir web uygulaması geliştirecek olursanız, projelerin genellikle kullanıcılar tarafından görülmeyen yönetim sisteminin, sitenin kullanıcılar tarafından görülebilen kısmından daha büyük olduğunu fark edebilirsiniz.

Bakış Açısı Analizi

Bakış Açısı Analizi

Her bir paydaş grubu tarafından görülen gereksinimleri analiz eder.

Örnek: Üniversite Mal Kabul Sistemi

- Başvuru
- Üniversite yönetimi

Kabul ofisi

Mali yardım ofisi

Özel ofisler (Örneğin, güzel sanatlar, geliştirme vs.)

- Akademik bölümler
-
- *Bilgi işlem personeli*
 - *İşletim ve bakım*

Özel Çalışmalar

Bazen gereksinimlerin bir parçası olarak özel çalışmalar yapmanız gerekebilir:

Pazar araştırması

odak grupları, anketler, rekabet analizi vb.

Örnek: Apple'ın güçlü yönlerini vurgulayan Windows XP çalışması

Teknik değerlendirme

deneyler, prototipler vb.

Örnek: Windows XP'nin daha hızlı başlatılması

Gereksinimlerin Belirlenmesi: Gerçekçilik ve Doğrulanabilirlik

Gereksinimler **gerçekçi** olmalı, yani gereksinimleri karşılamak mümkün olmalıdır.

Yanlış: *Sistem x yeteneğine sahip olmalıdır (bilinen hiçbir bilgisayar sistemi x'i makul bir maliyetle yapamıyorsa).*

Gereksinimler **doğrulanabilir** olmalıdır, yani gereksinimler **kabul testinin** temeli olduğundan, bir gereksinimin karşılanıp karşılanmadığını test etmek mümkün olmalıdır.

Yanlış: *Sistemin kullanımı kolay olmalıdır.*

Doğru: *Bir günlük eğitimden sonra, bir operatör saatte 50 işlem gerçekleştirebilmelidir. Bu kesin ve doğrulanabilirdir.*

Gereksinimlerin Belirtilmesi: İletişim

Heavyweight süreçlerde, gereksinimler tam bir spesifikasyonla tanımlanır. Lightweight süreçlerde ise spesifikasyon, belirsizliğin olabileceği seçilmiş parçaları kapsar.

Her ikisinde de hedef:

Müşteriler ve geliştiriciler arasındaki anlaşmanın kayda geçmesi

- Kabul testi için bir temel sağlar.
- Görünürlük sağlar.
- Sistem ve program tasarımı için bir temel oluşturur.
- Bu sistem veya alt sistem üzerinde çalışabilecek veya bu sisteme güvenebilecek diğer ekiplerle iletişim kurmak için kullanılabilir.
- Gelecekteki bakım yapacak geliştiricileri bilgilendirmeyi sağlar.



Lightweight süreçlerde hiçbir spesifikasyona ihtiyaç olunmadığı düşüncesi yaygın bir hatadır.

Lightweight Süreçler

Lightweight süreçlerde, sürecin başında müşteriyle mutabakata varılması gereken temel gereksinimlerle geliştirme sürecinin daha sonrasına bırakılabilecek ayrıntıları belirleyebilmek için deneyim ve muhakeme gücüne ihtiyacınız vardır.

Ayrıntılı spesifikasyonların genellikle gerekli olduğu örnekler

- İş kuralları (örneğin, bir muhasebe standardını takip etmesi gerekiyorsa belirtilmelidir.) ,
- Yasal kısıtlama (Örneğin, verilerin saklanması, mahremiyet yasaları)
- Veri akışı (örneğin, veri kaynakları, veri doğrulama)

Yaygın bir hata, önemli ayrıntıları kaçırmaktır. Bu, müşteri ve geliştiriciler arasında yanlış anlaşılmalara neden olur. Lightweight süreçlerin tüm amacı, minimum ara belgelere sahip olmaktır, ancak bazılarına ihtiyacınız hala vardır.

Lightweight Süreçler (devam)

Lightweight süreçlerde bir taslak spesifikasyon + diğer araçlar bulunur. Böylece;

- Temel gereksinimleri uygun ayrıntılarla açıklayan belgeler olacaktır.
- Müşteri ve geliştiriciler tarafından gözden geçirilmiş olması gerekiyor.

Diğer ayrıntılar ek araçlar tarafından sağlanacaktır, ör:

- Kullanıcı arayüzü mock-upları veya gösterimi.
- Modeller, örneğin veri tabanı şeması, durum makinesi vb..

Müşteriler prototipleri ve modelleri spesifikasyondan daha iyi anlar

- Yinelemeli geliştirme veya çevik süreçler, müşterinin nihai sistemin resmi bir spesifikasyon olmadan anlayabilmesini sağlar.

Devlet Sistemlerinde Gereksinimler

ABD'deki ve bütün dünyadaki devlet sistemleri, gecikmeler ve maliyet aşırımları ile birlikte zayıf işlevsellik konusunda bir üne sahiptir.

- Vergi mükelleflerinin parasının sorumlu kullanımı, her sürecin kararlaştırılan bir fiyattan tanımlanmış bir çıktıya sahip olduğu sözleşmelere yol açar.
- Birçok sözleşme, tasarım ve uygulama sözleşmesi yapılmadan önce ayrıntılı bir gereksinim spesifikasyonu talep edilir.
- Bu, genellikle gereksinimlerin değiştirilmesi için cezalarla birlikte heavyweight gelişmeye yol açar.

Ancak birçok hükümet sisteminde gereksinimler iyi anlaşılmamıştır.

- Gereksinimlerin esnek olduğu ve tasarımın yinelemeli olarak geliştiği bir geliştirme süreci kullanılmalıdır.
- Bu tür geliştirme süreçleri için sözleşmelerin yazılması zordur, ancak daha iyi yazılımların geliştirilmesini sağlar.

İşlevsel Gereksinimler

İşlevsel gereksinimler, sistemin gerçekleştirmesi gereken işlevleri tanımlar. "Sistem ne yapıyor?" sorusuna cevap verir.

Aşağıdaki gibi konuları içerirler:

- İşlemler
- Veri
- Kullanıcı arayüzleri

İşlevsel Olmayan Gereksinimler

Sistemin gerçekleştirmesi gereken işlevlerle doğrudan ilgili olmayan gereksinimler

Ürün gereksinimleri

performans, güvenilirlik, taşınabilirlik vb...

Organizasyonel gereksinimler

teslimat, eğitim, standartlar vb.

Dış gereksinimler

yasal, birlikte çalışabilirlik vb...

Pazarlama ve halkla ilişkiler

Örnek:

Bir yazılım anlaşması sonunda altı ayın sonunda yöneticilere canlı bir uygulamanın gösterilmesi üzerine anlaşıldı.

İşlevsel olmayan gereksinimlere örnek

Örnek: Kütüphane Sistemi

Müşteri personelinin aşına olduğu teknoloji kullanılmalı:

- Donanım ve yazılım sistemleri (IBM/Unix)
- Veritabanı sistemleri (Oracle)
- Programlama dilleri (C ve C++)

Müşteri resmi bir kütüphanedir

- Engelli kişilerin erişilebilirliği gibi devlet sistemlerini kapsayan düzenlemeler
- Diğer büyük kütüphanelerin saygı duyacağı bir sistem geliştirmenin önemi

Gereksinimler: Müşteri ile Müzakere

Bazen müşteri, uygulanması çok pahalı veya imkansız olan işlevler talep edebilir. Bazen de iki gereksinim birbiriyle çelişebilir.

Bu durumda müzakerede bulunmak gerekir.

- Gereksinimi müşteriyle konuşun. Neden isteniyor? Eşdeğer bir alternatif var mı?
- Endişenizin arkasındaki mantığı açıklayın. Teknik, organizasyonel ve maliyet etkilerini açıklayın.
- Önerilere açık olun. Anlayışınızda bir boşluk var mı? Belki ikinci bir görüş başka yaklaşımlar önerebilir.

Müşteri ve geliştirme ekibi bu soruları çözmesi önemlidir.

Örnek

Bir önceki projede müşteri çok çeşitli formatların desteklenmesini istemiş olabilir. Müzakereden sonra, ilk aşamanın yalnızca PDF'i destekleyeceği konusunda anlaşma gerçekleşebilir.

Gereksinimler v. Sistem Tasarımı

Teknik kararlar

- Gereksinim analizi, sistem tasarımı hakkında minimum varsayımlarda bulunmalıdır.
- Ancak gereksinim tanımı, bilgisayar teknolojileri ve mevcut kaynaklarla tutarlı olmalıdır.

Uygulamada, analiz ve tasarım iç içe geçmiştir. Fakat:

- 1. Tasarımla ilgili varsayımların gereksinim analizini etkilemesine izin vermeyin.*
- 2. Gereksinim analizinin sistem tasarımına önyargılı davranmasına izin vermeyin.*

Erciyes Üniversitesi

Bilgisayar Mühendisliği Bölümü

BZ 313 Yazılım Mühendisliği

6. Gereksinim Analizi

Ders Sonu