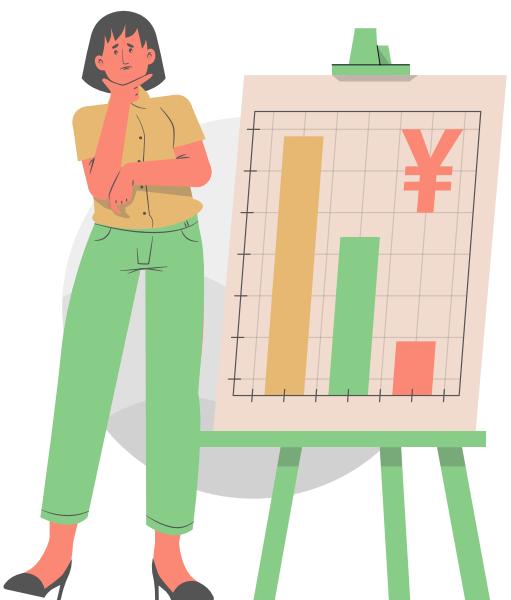


SİSTEM ANALİZİ VE TASARIMI - BIL206

Öğr. Gör. Buse Yaren TEKİN



İçerikler

Spiral Model

Artırımlı Model

Şelale Model

Çevik Model

Sistem Geliştirme Modellerinin Seçimi



SPIRAL MODEL

Bölüm 1

Spiral Model

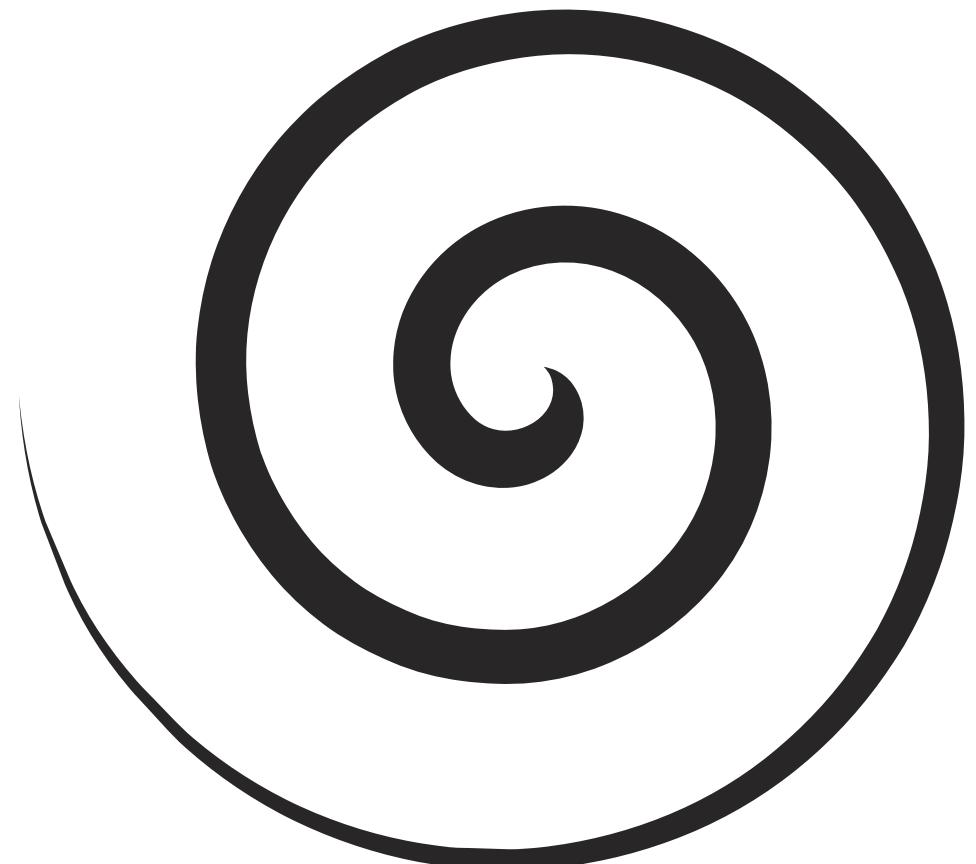
Spiral model, yinelemeli geliştirme süreci modelini Şelale modelinin öğeleriyle birleştiren, risk yönetimi için kullanılan bir sistem geliştirme yaşam döngüsü (SDLC) yöntemidir. Spiral model yazılım mühendisleri tarafından kullanılır ve büyük, pahalı ve karmaşık projeler için tercih edilir.

Bir diyagram olarak bakıldığında, spiral model çok sayıda döngüye sahip bir bobine benzer. Döngü sayısı her projeye göre değişir ve genellikle proje yöneticisi tarafından belirlenir. Spiralin her döngüsü, yazılım geliştirme sürecinde bir aşamadır.



Spiral Model

Spiral model ile geliştirme yapan takım öncelikle küçük bir gereksinim seti ile başlar ve bu gereksinimler için her bir geliştirme aşamasından geçer. Uygulama üretime hazır olana dek, giderek artan spirallerde ek gereksinimler için işlevsellik ekler.



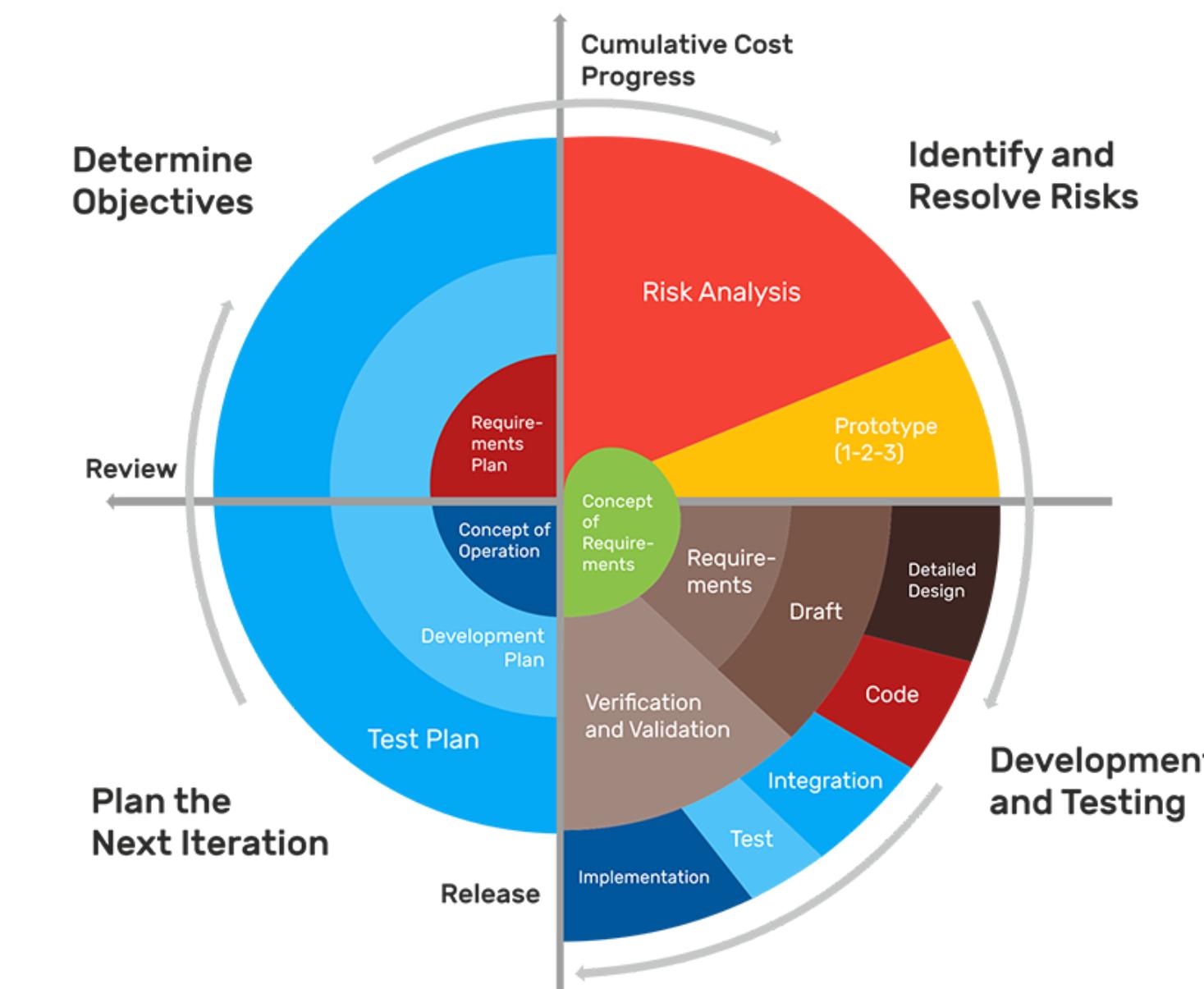
Spiral Model

Planning (Planlama): Yineleme için maliyet, program ve kaynakların tahmini yapılır. Sistem gereksinimlerinin anlaşılmasını da içerir.

Risk Analysis (Risk Analizi): Potansiyel risk tanımlaması yapılırken, risk azaltma stratejisi planlanır.

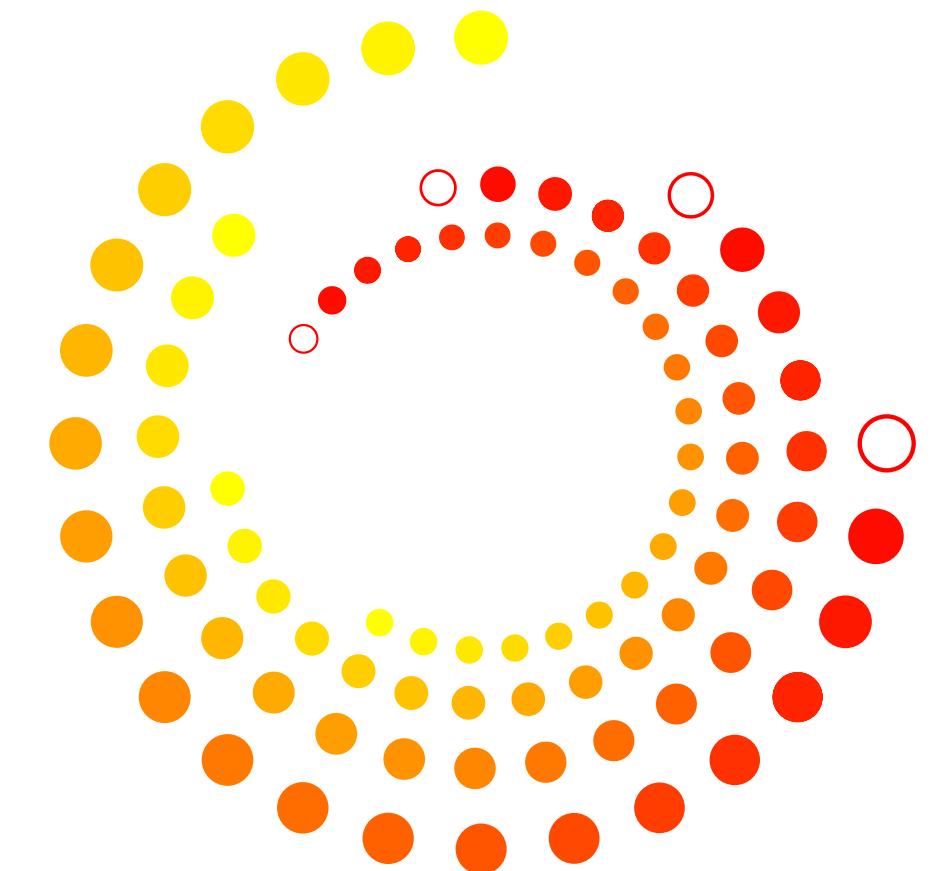
Engineering (Mühendislik): Test, kodlama ve yazılımın dağıtımını içerir.

Evaluation (Değerlendirme): Yazılımın müşteri tarafından değerlendirilmesi yapılır ayrıca maliyet aşımı ve program kayması gibi risklerin tanımlanması, izlenmesini içerir.



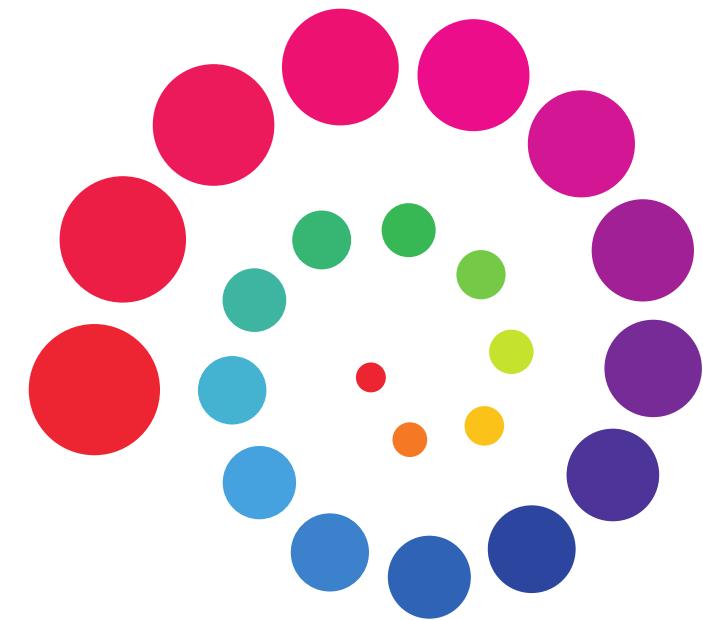
Ne Zaman Spiral Model Kullanmalıyız?

- Proje büyük ise.
- Sürümlerin sık olması gerekiğinde.
- Bir prototip oluşturulması uygulanabilir olduğunda.
- Risk ve maliyet değerlendirmesi önemli olduğunda.
- Proje orta ve yüksek riskli ise.
- Gereksinimler belirsiz ve karmaşık olduğunda.
- Değişiklikler sürekli olarak istenebilecek durumda ise.
- Ekonomik önceliklerdeki değişiklikler nedeniyle uzun vadeli proje taahhüdü mümkün olmadığında.



Avantaj ve Dezavantajları

- Avantaj: Ek işlevsellik veya değişiklikler daha sonraki bir aşamada yapılabilir.
- Dezavantaj: Programın kayması veya bütçeyi aşma riski vardır.
- Avantaj: Sürekli geliştirme sağlandığı için risk yönetimi kolaylaşır.
- Dezavantaj: Düzgün çalışması için spiral model protokolüne kesinlikle uyulmalıdır.
- Avantaj: Geliştirme hızlıdır ve özellikler sistematik bir şekilde eklenir.
- Dezavantaj: Ara aşamalara sahip olduğu için dokümantasyon daha fazladır.



ARTIRIMLI MODEL

Bölüm 2

Artırımlı (Incremental) Model

Bu modelde, gereksinimler yazılım geliştirme döngüsünün birden fazla bağımsız modülüne ayrılmıştır.

Her yineleme gereksinimler, tasarım, kodlama ve test aşamalarından geçer. Ve sistemin sonraki her sürümü, tüm tasarlanmış işlevler uygulanana kadar önceki sürüme işlev ekler.

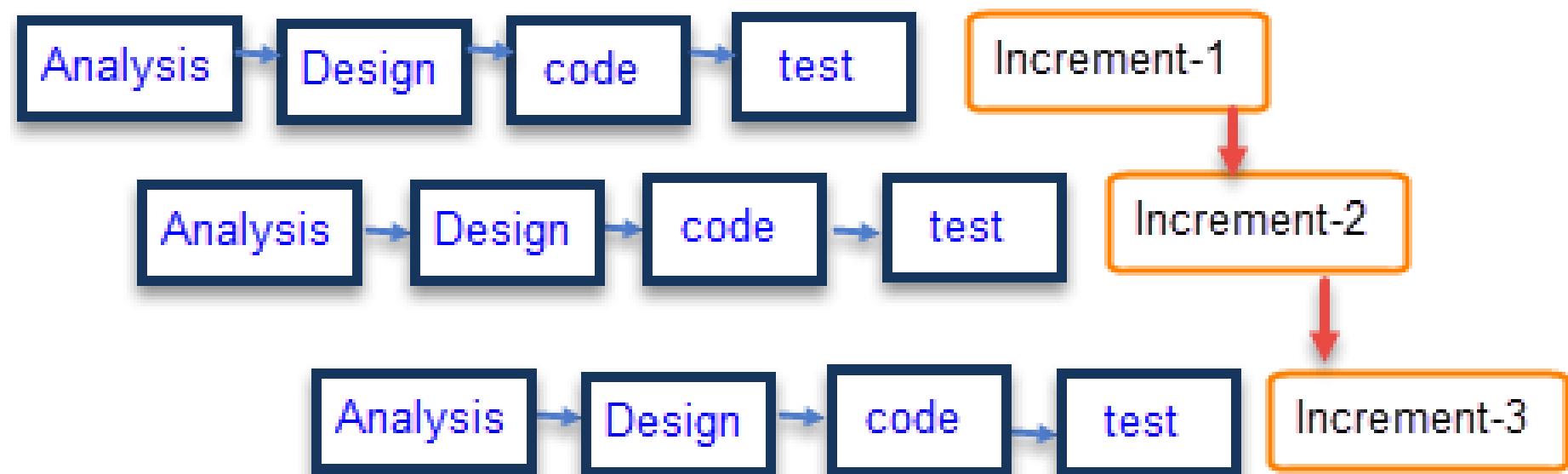
A vertical sequence of five black circles, each containing the number '1'. To the right of the first circle is a red plus sign ('+') and an equals sign ('='). To the right of the second circle is another red plus sign ('+') and an equals sign ('='). To the right of the third circle is another red plus sign ('+') and an equals sign ('='). To the right of the fourth circle is another red plus sign ('+') and an equals sign ('='). A red curved arrow originates from the bottom of the fourth circle and sweeps upwards and to the left, pointing towards the text 'Incremental Model'.

$$\begin{array}{r} 1 \\ + \\ 1 \\ + \\ 1 \\ = \\ 1 \\ + \\ 1 \\ = \\ 2 \end{array}$$

**Incremental
Model**

Artırımlı (Incremental) Model

İlk “increment” sağlandığında, sistem üretime alınır. Bu ilk “increment” dediğimiz aslında en basit gereksinimleri karşılayan, temel bir üründür ve ek özellikler bir sonraki “increment” ile eklenir. Temel ürün müşteri tarafından analiz edildikten sonra, gelecek increment için planlama yapılır.



Incremental Model

Artırımlı Model Özellikleri

- Sistem geliştirme birden çok mini geliştirme projelerine bölünür.
- Kısmi sistemler, toplam sistemi elde etmek için art arda inşa edilir.
- En yüksek önceliğe sahip gereksinim ilk olarak ele alınır.
- Gereksinimin geliştirilmesi tamamlandıktan sonra bu artırımın gereksinimi durdurulur.



Ne Zaman Artırımlı Model Kullanmalıyız?

- Sistem gereksinimleri tamamen anlaşılmış ise.
- Ürünün piyasaya erken sürülmeye talebi var ise.
- Yazılım geliştirme ekibi çok eğitimli değil ise.
- Yüksek riskli özellikler ve hedefler var ise.
- Web uygulamaları ve ürün tabanlı şirketler için daha fazla kullanımı bulunmaktadır.



Avantaj ve Dezavantajları

- Avantaj: Hızlı bir şekilde yazılım geliştirilir.
- Dezavantaj: İyi bir planlama ve tasarım gerektir.
- Avantaj: Gereksinimleri ve kapsamı değiştirmek ucuz ve esnektir.
- Dezavantaj: Tüm yazılım geliştirme döngüsünden önce bütün gereksinimler toplanmadığından, sistem mimarisi ile ilgili sorunlar çıkabilir.
- Avantaj: Diğer modellere göre daha ucuz.
- Dezavantaj: Bir birimde meydana gelen hatayı düzeltmek için bütün birimleri düzenlemek gereklidir ve çok zaman zaman alır.
- Avantaj: Hataları tespit etmek çok kolaydır.





ŞELALE MODEL

Bölüm 3

Şelale (Waterfall) Model

1970 yılında Winston Royce tarafından tanıtılmıştır.

Waterfall, yazılım geliştirmeyi ön-tanımlı aşamalara ayıran sıralı bir modeldir.

Her bir aşama başlamak için bir önceki aşamanın tamamlanmasını beklemelidir.

Her bir aşama spesifik bir aktivite gerçekleştirmek için tasarlanmıştır.

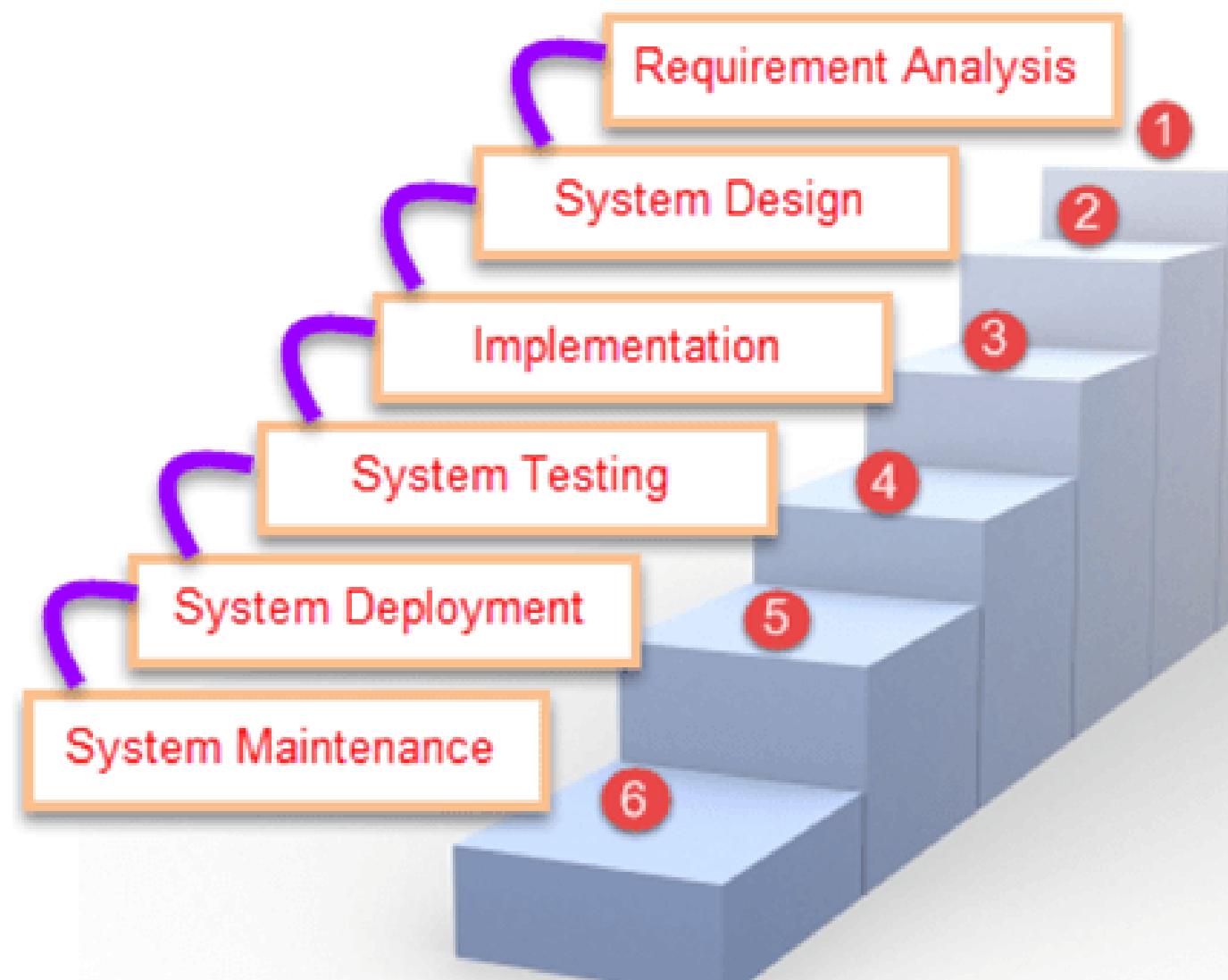


Şelale (Waterfall) Model

Requirement Analysis (Gereksinim Analizi): Geliştirilmesi istenen sistemin gereksinimleri detaylı olarak müşteriden toplanır.

Design (Sistem Tasarımı): Kullanılacak olan programlama dili, veritabanı ve diğer teknik detaylar tasarlanır.

Implementation (Uygulama): Kodlamanın yapıldığı aşamadır.

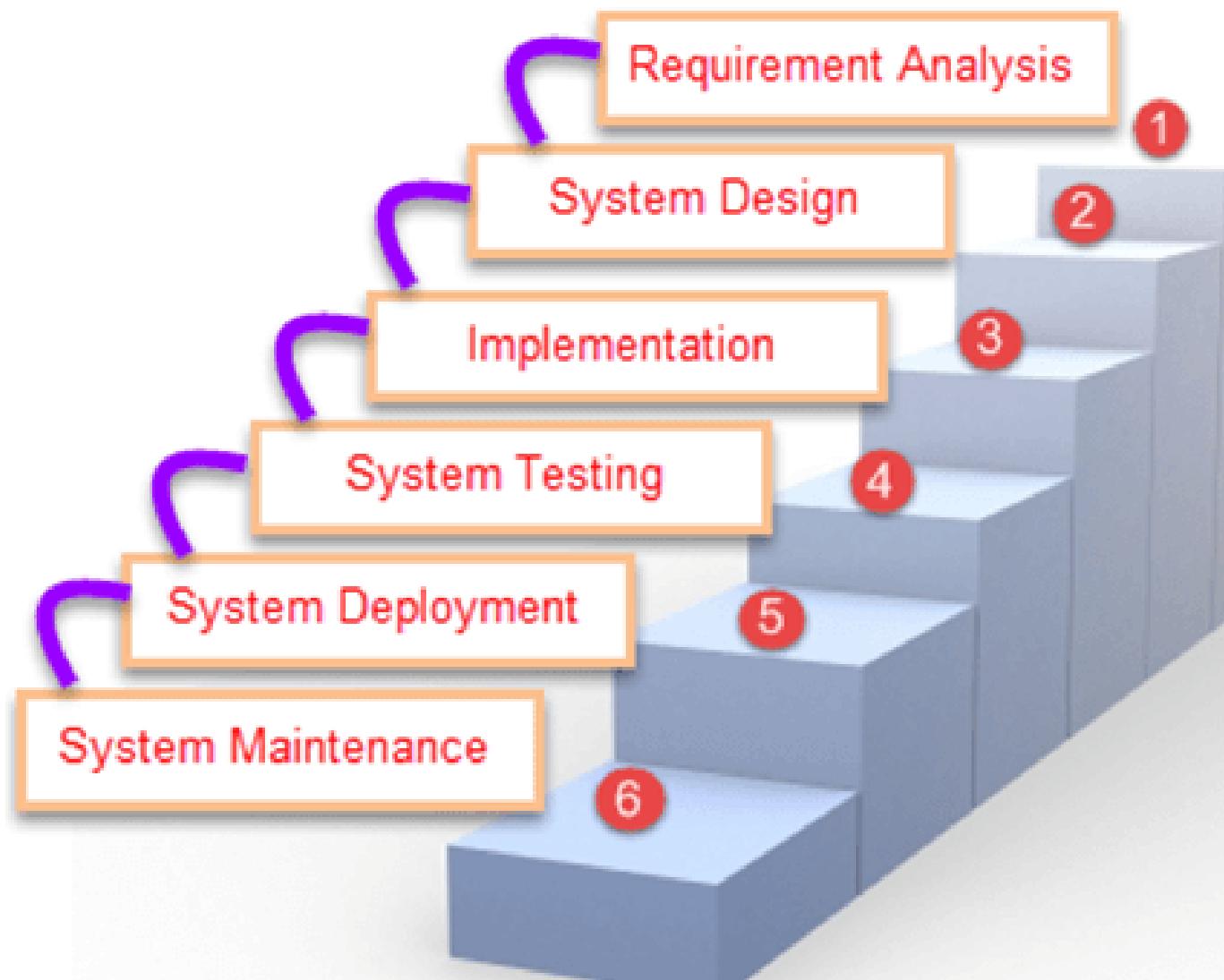


Şelale (Waterfall) Model

System Testing (Sistem Testi): Yazılımın istenilen özelliklere göre oluşturulduğunu doğrulamak için testler gerçekleştirilir.

System Deployment (Sistem Dağıtıımı): Uygulamanın dağıtıımı ilgili ortamda gerçekleştirilir.

System Maintenance (Sistem Bakımı): Dağıtımdan sonra, müşteri isteği doğrultusunda uygulama üzerinde değişiklikler gerçekleştirilebilir.



Ne Zaman Waterfall Modelini Kullanmalıyız?

- Gereksinimler çok sık değişmiyorsa.
- Uygulama çok karışık veya büyük değilse.
- Ortam sabitse.
- Kullanılan teknoloji ve araçlar sabitse.
- Kaynaklar müsait ise.



Avantaj ve Dezavantajları

- Avantaj: Bir aşama başlamadan, ondan önceki aşama tamamlanmalıdır.
- Dezavantaj: Hatalar sadece aşama sürerken düzeltilebilir.
- Avantaj: Proje tamamen minimum müşteri müdahalesi ile proje ekibine bağlıdır.
- Dezavantaj: Müşteri geri dönüşü oldukça değerli olmasına rağmen, devam eden geliştirme sürecine dahil edilemez.
- Avantaj: Ayrıntılı dokümantasyon her aşamada gerçekleştirilir.
- Dezavantaj: Dokümantasyon geliştirici ve test ekipleri için oldukça zaman alıcı olabilir.



ÇEVİK MODEL

Bölüm 4

Çevik (Agile) Model

Agile (Çevik) model geliştirilen ürünün hızlı teslimatı ile müşteri memnuniyeti ve süreç uyarlanabilirliğine odaklanan, iterative ve incremental süreç modellerinin birleşimidir.

Model, ürünü küçük artımlı yapılara böler, bu yapılar yinelemeler ile sağlanır. Her bir yineleme 1–3 hafta sürer.



Çevik (Agile) Model

Her yineleme, aşağıdaki gibi çeşitli alanlarda aynı anda çalışan çapraz fonksiyonel ekipleri içerir;

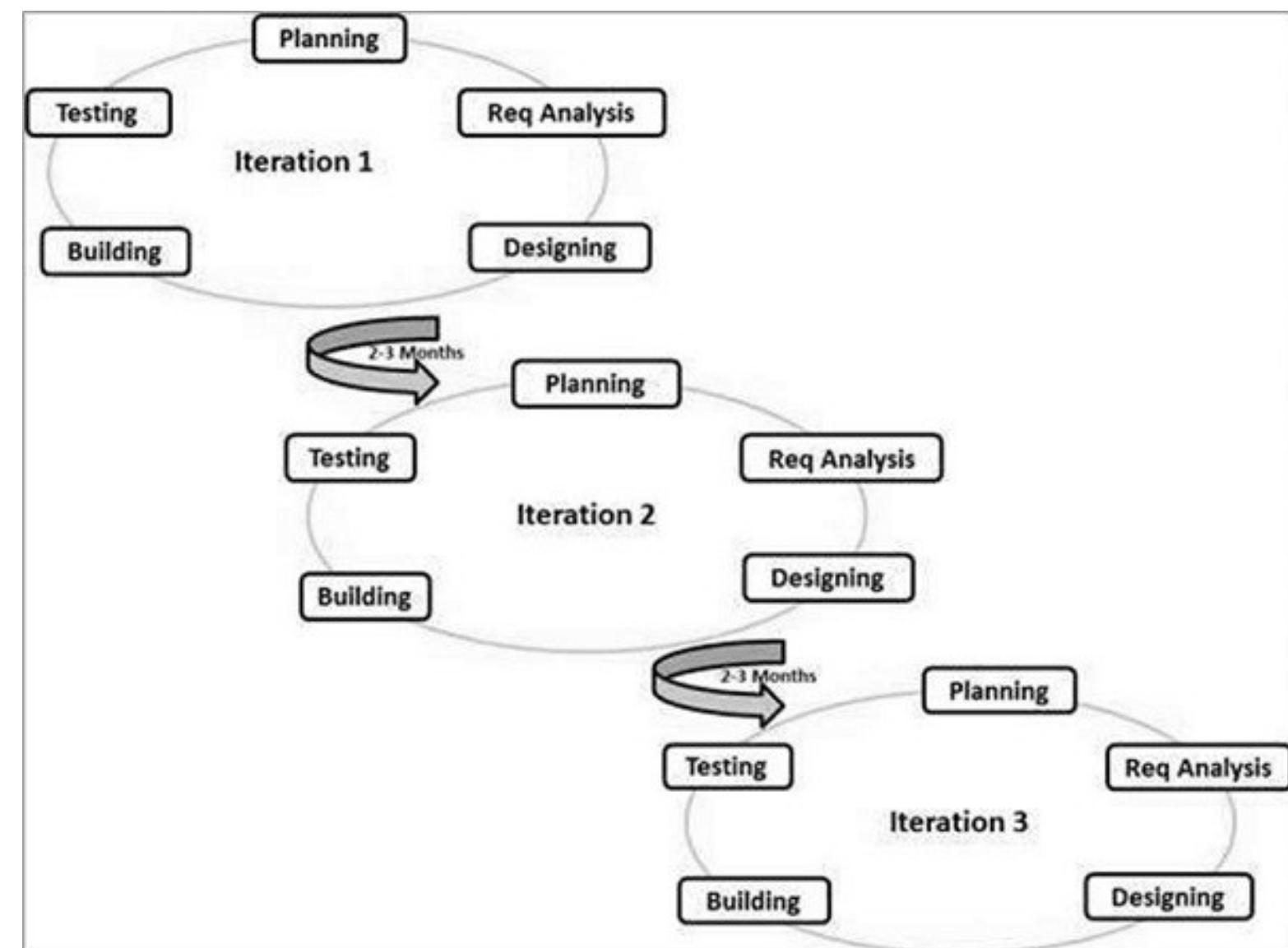
- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing

Yineleme sonucunda ürün müşteriye sunulur.



Çevik (Agile) Model

Çevik model, her projenin farklı şekilde ele alınması gereğine ve mevcut yöntemlerin proje gereksinimlerine en uygun şekilde uyarlanması gereğine inanmaktadır. Agile'da görevler, bir sürüme özgü özellikler sunmak için zaman kutularına (küçük zaman çerçeveleri) bölünür. Yinelemeli yaklaşım benimsenir ve her yinelemeden sonra çalışan yazılım yapısı sunulur.



Çevik (Agile) Model

Agile prensiplerini aşağıdaki şekilde sıralayabiliriz;

- **Bireyler ve etkileşimler:** Agile geliştirmede ortak konum ve eşli programlama gibi etkileşimler kadar öz-orgütleme ve motivasyon da önemlidir.
- **Çalışan yazılım:** Demo yazılım, müşterinin ihtiyaçlarını anlama konusunda sadece dokümantasyona bağlı kalmaya kıyasla en iyi iletişim aracıdır.
- **Müşteri işbirliği:** Projenin başında çeşitli sebepler nedeniyle gereksinimler tamamen toplanamadığından, sürekli müşteri etkileşimi ürün gereksinimlerini düzgün bir şekilde tamamlamak için oldukça önemlidir.
- **Değişikliğe yanıt vermek:** Agile gelişim değişiklik ve sürekli geliştirmeye hızlı yanıt verebilmeye odaklıdır.



SİSTEM GELİŞTİRME MODELLERİNİN SEÇİMİ

Bölüm 5

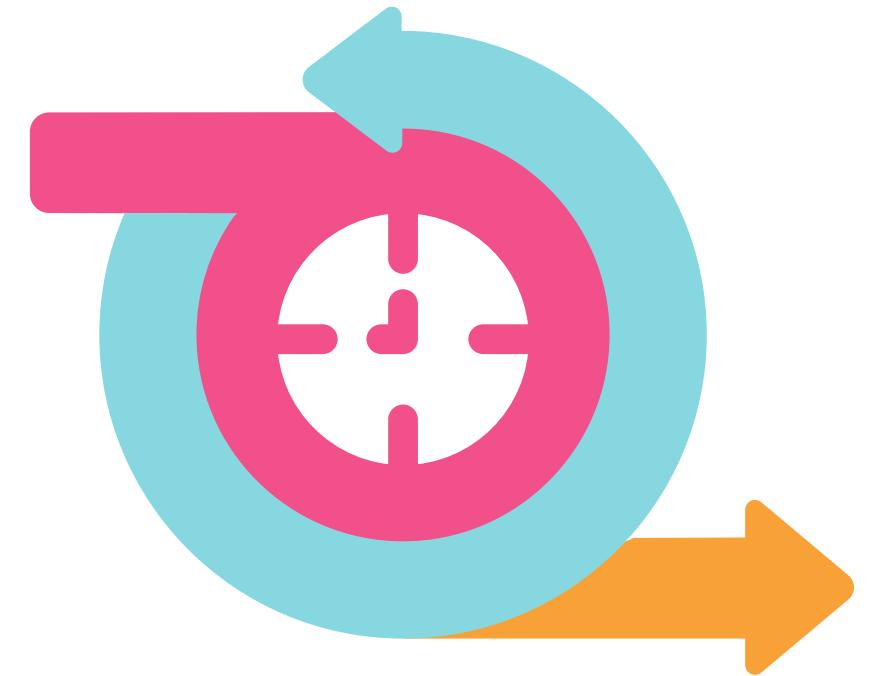


FEEDBACK



Sistem Geliştirme Modellerinin Seçimi

Anlatılan tüm geliştirme modeli yaklaşımlarında sistem analisti öncelikle organizasyonun yapısını, işleyiş biçimini, süreçleri vb. anlamak zorundadır. Sonra ayrıntılı sorular ile veri toplanır ve işletmedeki iş süreçleri belirlenir. Böylece gerçekleştirilecek uygulamanın tipi ortaya konulur. Farklı uygulama tiplerinde farklı sistem geliştirme teknikleri kullanılmalıdır.



Sistem Geliştirme Modellerinin Seçimi

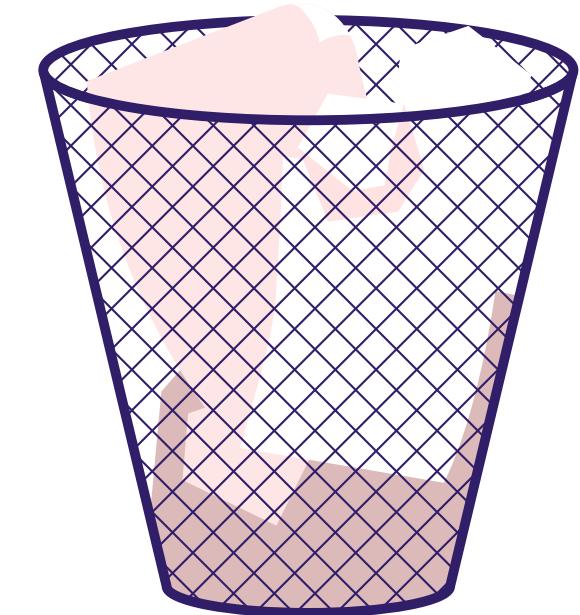
Bunun sebeplerinden bazıları aşağıda verilmiştir:

- **Maliyet ve değişiklik isteklerinin sıklığı:** Bazı sistemlerde değişiklik gereksinimlerini uygunlamak çok aşırı maliyetli olabilirken, diğer sistemlerde müşteri gereksinimleri doğrultusunda sık sık değişiklik yapmak gerekebilir. Pahalı ön analiz gerektiren ve gereksinimlerin tutarlı ve doğru olarak tüm müşteri isteklerini kapsadığının gösterildiği sistemlerde değişikliğin maliyetleri yüksek olacaktır.



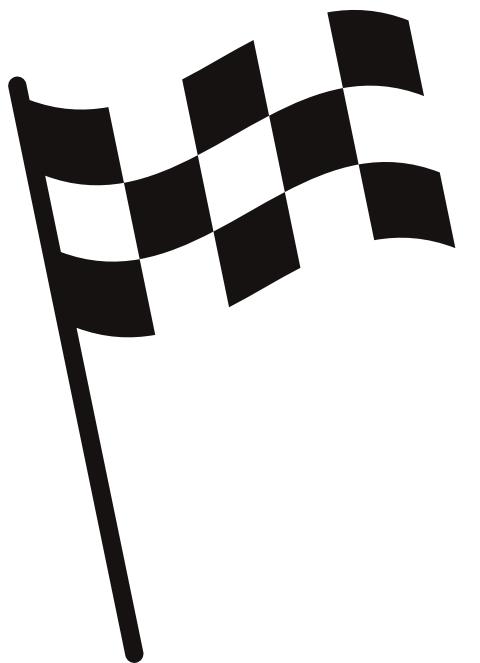
Sistem Geliştirme Modellerinin Seçimi

- **İşlevsel olmayan gereksinimlerin önemi:** Farklı sistemlerde işlevsel olmayan gereksinimlerin öncelikleri farklı olur.
 - Örneğin bankacılık bilgi sisteminde güvenlik gereksinimi birinci öncelikli olacaktır. Diğer yandan interaktif işlemlerin yoğun olduğu sanal gerçeklik sistemlerinde ise hızlı cevap verme ve kullanışlılık öncelikli gereksinimler olacaktır. Bu nedenle, **bankacılık sistemlerinde interaktif sanal gerçeklik tarzı arayüzler için çok fazla arayüz tasarıımı gerekmeyecek, sanal gerçeklik kullanılan oyun tarzı sistemlerde ise yüksek güvenliği sağlamak üzere teknikler kullanılacaklardır.**



Sistem Geliştirme Modellerinin Seçimi

- **Bilgi sistemi ömür ve teslim zamanlaması:** Bazı sistemlerin ömrü nispeten kısa olurken (mobil, web tabanlı sistemler) diğerlerinin onlarca yıllık ömrü olur (büyük kumanda ve kontrol sistemleri). Bazı sistemler, yararlı olma ömrü kısa olduğundan hızlı olarak teslim edilmelidir. Bu tip sistemlerde kısa ömürlü geliştirme ve hızlı teslim etme imkânı olan teknikler tercih edilir. Uzun yaşam ömrü olan sistemlerde ise uzun dönem bakım ve destek hizmetlerini sağlayabilecek tasarım modelleri ve sistem süreçleri kullanılır.



Sistem Geliştirme Modellerinin Seçimi

Sistem Geliştirme Yaşam Döngüsü (SGYD)	Doğrusal Ardışık Model	Prototip Model	Artımlı Model	Spiral Model	Bileşen Tabanlı Geliştirme Modeli	Çevik Model (Scrum)
Kapsam ve Planlama	Kapsam ve Planlama	Tur 1, 2, ... Müşteri İstekleri	Artım 1,2,... Kapsam Planlama	Tur 1, 2, ... Müşteri iletişim	Tur 1, 2, ... Müşteri iletişim	Planlama ve Mimari Tasarım
Gereksinim Analizi	Gereksinim Analizi	Tur 1, 2, ... Prototip Model Geliştir Güncelle Müşterinin Prototip Model Testleri	Artım 1,2,... Gereksinim Analizi	Tur 1, 2, ... Planlama Risk Analizi	Tur 1, 2, ... Planlama Risk Analizi	Devir 1,2,... Plan Değerlendirme Devir 1,2,... Seçim
Tasarım	Tasarım		Artım 1,2,... Tasarım	Tur 1, 2, ... Mühendislik	Tur 1, 2, ... Mühendislik Bileşen Tabanlı	Devir 1,2,... Geliştirme
Gerçekleştirme ve Test	Geliştirme ve Test		Artım 1,2,... Geliştirme ve Test	Tur 1, 2, ... Gerçekleştirme ve Kurulum Müşteri Değerlendirmesi	Tur 1, 2, ... Gerçekleştirme ve Kurulum Müşteri Değerlendirmesi	Devir 1,2,... Geliştirme Gözden Geçirme
Kurulum ve Dağıtım	Kurulum ve Dağıtım		Artım 1,2,... Sürüm 1,2,... Kurulumu	Tur 1, 2, ... Müşteri Değerlendirmesi	Tur 1, 2, ... Müşteri Değerlendirmesi	Kapanış
Operasyon ve Bakım	Operasyon ve Bakım					

Tabloda görüldüğü gibi Prototip model sadece gereksinim oluşturma aşamasında kullanıldığından diğer sistem geliştirme modellerine alternatif değildir. Diğer modellerde ise süreç içindeki alt fazlar arasında çok fazla bir fark yoktur. Modellerdeki süreçler arasındaki farklar: her bir fazı kaç kez uyguladığı (plan güdümlü veya iteratif yaklaşım), her bir fazda harcanılan kaynak miktarı (zaman, maliyet), fazlarda üretilen belgelerin miktarı ve geliştirme aşamasında değişiklik isteklerinin nasıl yönetildiğine göre oluşmaktadır.

Kaynaklar

1. Öğr. Gör. Aslı Birol, Sistem Analizi ve Tasarımı, BIL3403

Ders Notları

2. Anadolu Üniversitesi, BIL206, Sistem Analizi ve Tasarımı

Ders Notları

3. Medium, Kişisel Blog Yazısı.

