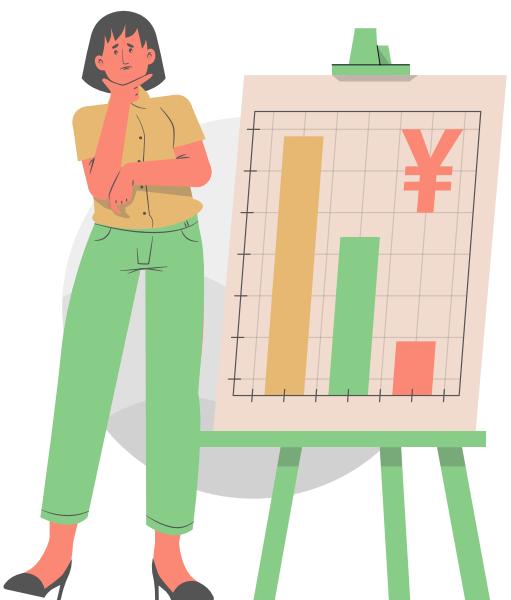


SİSTEM ANALİZİ VE TASARIMI - BIL206

Öğr. Gör. Buse Yaren TEKİN



İçerikler

Bilgi Sistemi Geliştirme Modelleri
Sistem Geliştirme Yaşam Döngüsü
Doğrusal Ardisık Model
Artımlı Model
Spiral Model
Çevik Model



BİLGİ SİSTEMİ GELİŞTİRME MODELLERİ

Bölüm 1

Bilgi Sistemi Geliştirme Modelleri

Bilgi sistemleri yazılım, donanım, veri ve birden fazla kişinin taraf olduğu iletişim ve ağ teknolojileri içeren karmaşık yapılardır. Bu nedenle bilgi sistemlerinin geliştirilmesi için standart hâle getirilmiş yöntemleri içeren modeller (sistem geliştirme modelleri olarak adlandırılır) kullanılmaktadır. Sistem geliştirme modelleri çerçevesinde; bilgi sistemi projelerinin yönetiminin kontrolü sağlanmakta ve uygulanılan yöntemler tanımlanmaktadır.



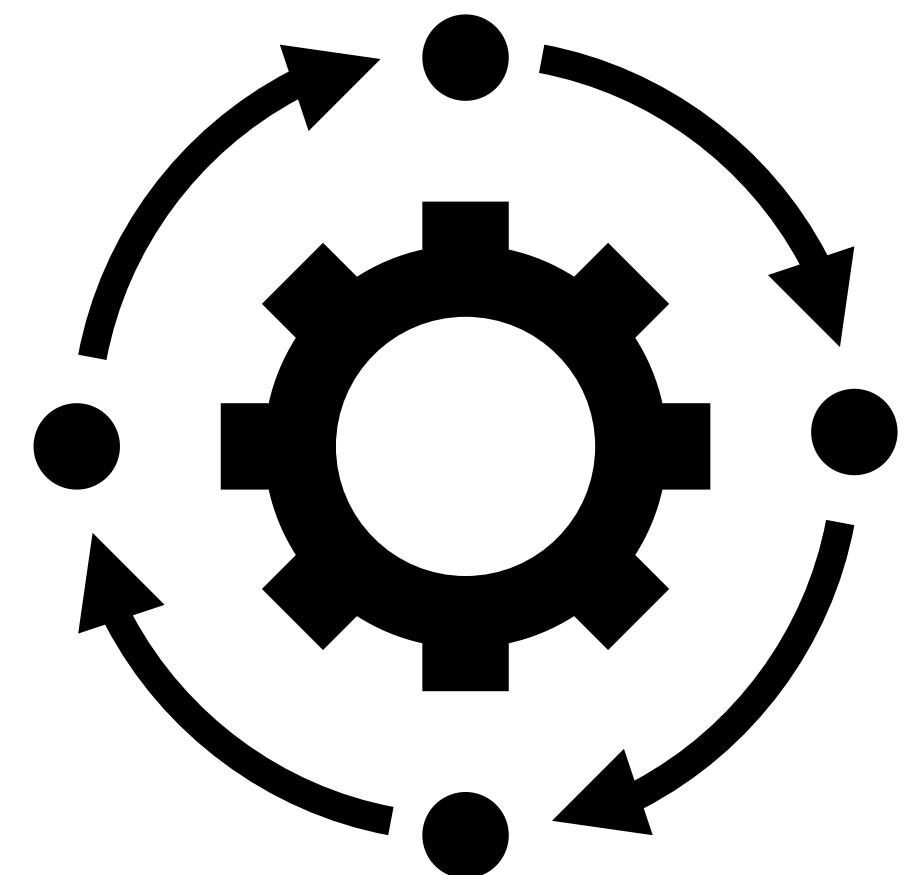
Bilgi Sistemi Geliştirme Modelleri

Modeldeki veri ve süreç tanımları ile iş ürünleri (diyagramlar, belgeler, formlar, veri ve raporlar, vb.) üretilmekte, kilometre taşları belirlenmekte ve kalite yönetimi sağlanmaktadır. Günümüzde işletmeler standartlaştırılmış sistem geliştirme modellerini, artan biçimde projelerine uyumlandırarak kullanmaktadır. Sistem geliştirme modellerinin kullanımı ile farklı projeler arasında aynı kaynakların paylaşılması sağlanmıştır.



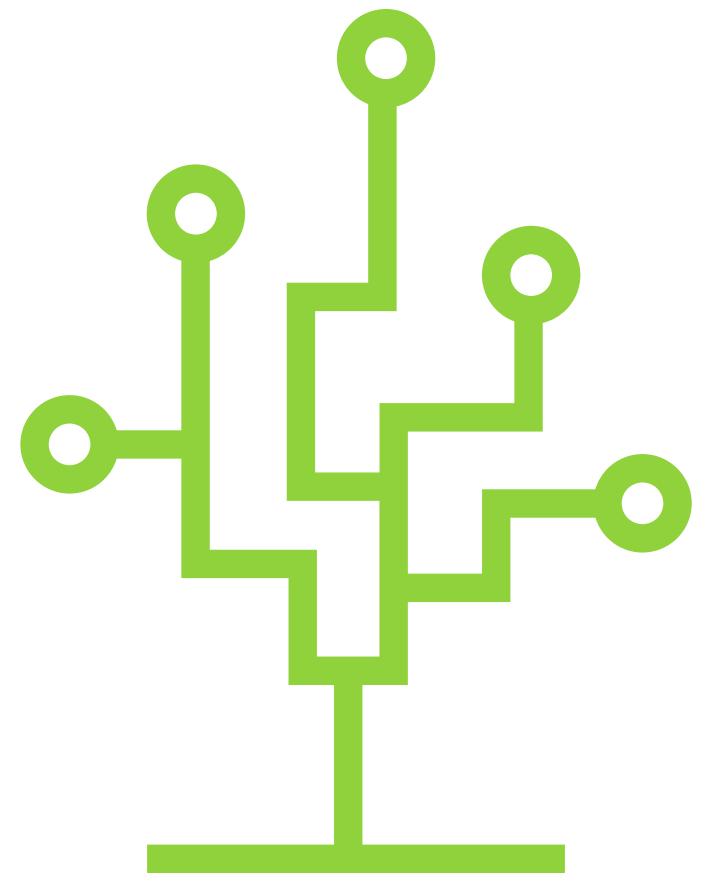
Bilgi Sistemi Geliştirme Modelleri

Sistem geliştirme modeli; faaliyetler, teknikler, uygulamalar, çıktılar ve sistem paydaşlarının kullandığı otomatik araçlar ile bilgi sistemlerinin geliştirilmesini ve sürekli iyileştirilmesini içerir.



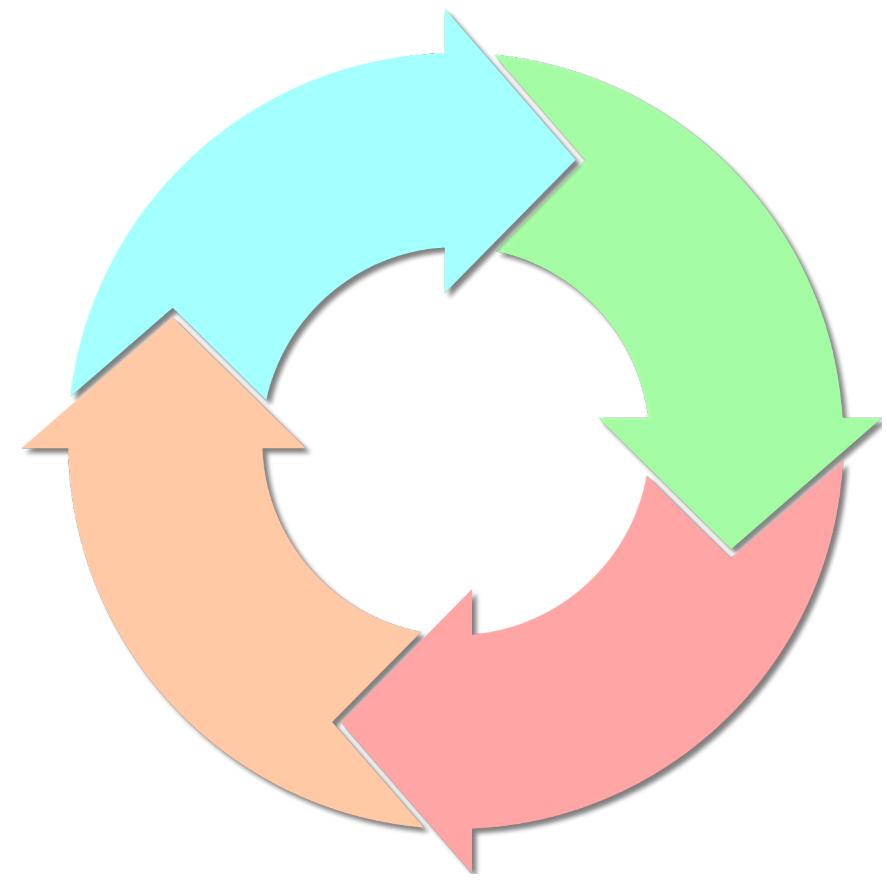
Bilgi Sistemi Geliştirme Modelleri

Sistem Geliştirme Yaşam Döngüsü, bilgi sistemi geliştirilirken uygulanan fazların kavramsal bir modelidir. Bu fazlar Kapsam ve Planlama, Gereksinim Analizi, Tasarım, Gerçekleştirme ve Test, Kurulum ve Dağıtım, Operasyon ve Bakım biçiminde sıralanır. Farklı sistem geliştirme modellerinde farklı sayıda faz olabilir. Faz sayısı konusunda analistler farklı görüşte olsalar da (tanımlar 3 fazdan 20 faza kadar değişebilmektedir), bu fazlarda ifade edilen genel yaklaşımın kullanılmasının gerekliliği konusunda herkes ortak anlayıştır.



Bilgi Sistemi Geliştirme Modelleri

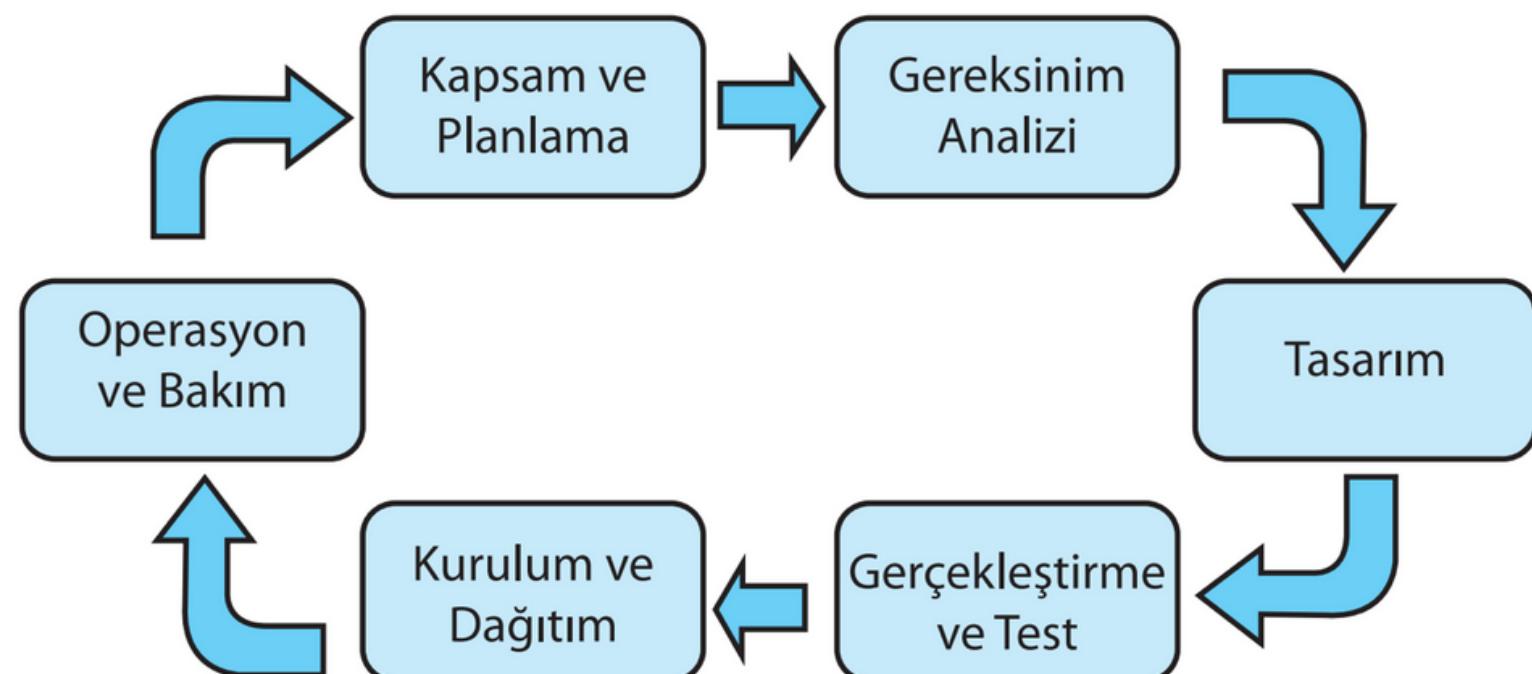
SGYD fazları sıralı olarak verilmiş olmakla beraber sıralı uygulanması gereklidir. Bu fazlar projeye bağlı olarak gerektiğinde kullanılabileceğini göstermektedir. Örneğin SGYD' nin herhangi bir fazında projedeki gerekliliklere göre daha önceki fazlara geri dönülebilir. Benzer biçimde iyi çalışmayan bir ürün kullanımından geri çekilerek tekrar geliştirilmek üzere başlangıç fazına geri döndürülebilir.



Bilgi Sistemi Geliştirme Modelleri

Büyük işletmeler genel olarak kendileri için özel olarak geliştirdikleri SGYD modellerini kullanırlar. Ancak bu özelleştirilmiş modeller de Şekilde verilmiş olan genel sistem geliştirme yaşam döngüsü fazlarını, tekniklerini ve araçlarını kullanır. Burada verilmiş olan SGYD fazları, bilgi sistemlerinin neredeyse tamamının yaşam döngüsünde kullanılabilir.

*Sistem Geliştirme
Yaşam Döngüsünün 6
Fazı (SGYD)*

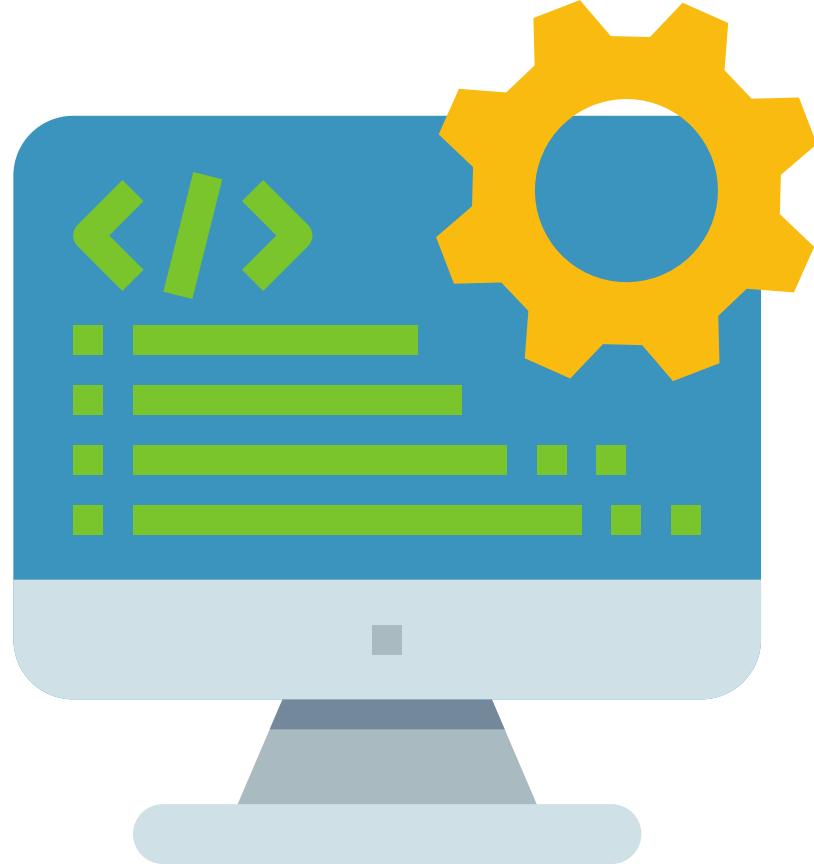


SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ

Bölüm 2

Kapsam ve Planlama

Sistem Geliştirme Yaşam Döngüsünün ilk fazında problemin kapsamı, fırsatları ve hedefleri tanımlanır. Projenin geri kalanının başarısı problemin doğru tanımlanmasına bağlıdır. Problemin kapsamında; projenin büyüklüğü, sınırları, projenin vizyonu, kısıt ve limitleri, projenin paydaşları, bütçesi ve zamanlaması oluşturulur. Bu aşamada ortaya konulan problemler bilgisayar destekli araçların kullanımı ile çözülperek işletmenin rekabet etme kabiliyeti arttırılır ve endüstri standartlarının yakalanması sağlanır.



Kapsam ve Planlama

Bu aşamada gerçekleştirilen önemli faaliyetlerden biri geliştirilecek bilgi sisteminin uygulanması ile işletmelerin kazanımlarının belirlenerek, hedefin tanımlanmasıdır. İlk fazın paydaşları sistem sahipleri, dâhilî sistem yöneticileri, proje yöneticileri ve sistem analistleridir. Genelde sistem kullanıcıları bu fazda yer almaz. Sistem sahipleri ve dâhilî sistem yöneticileri ile görüşmeler yapılır, bulgular özetlenir, projenin kapsamı tahmin edilir ve sonuçlar raporlanır.



Kapsam ve Planlama

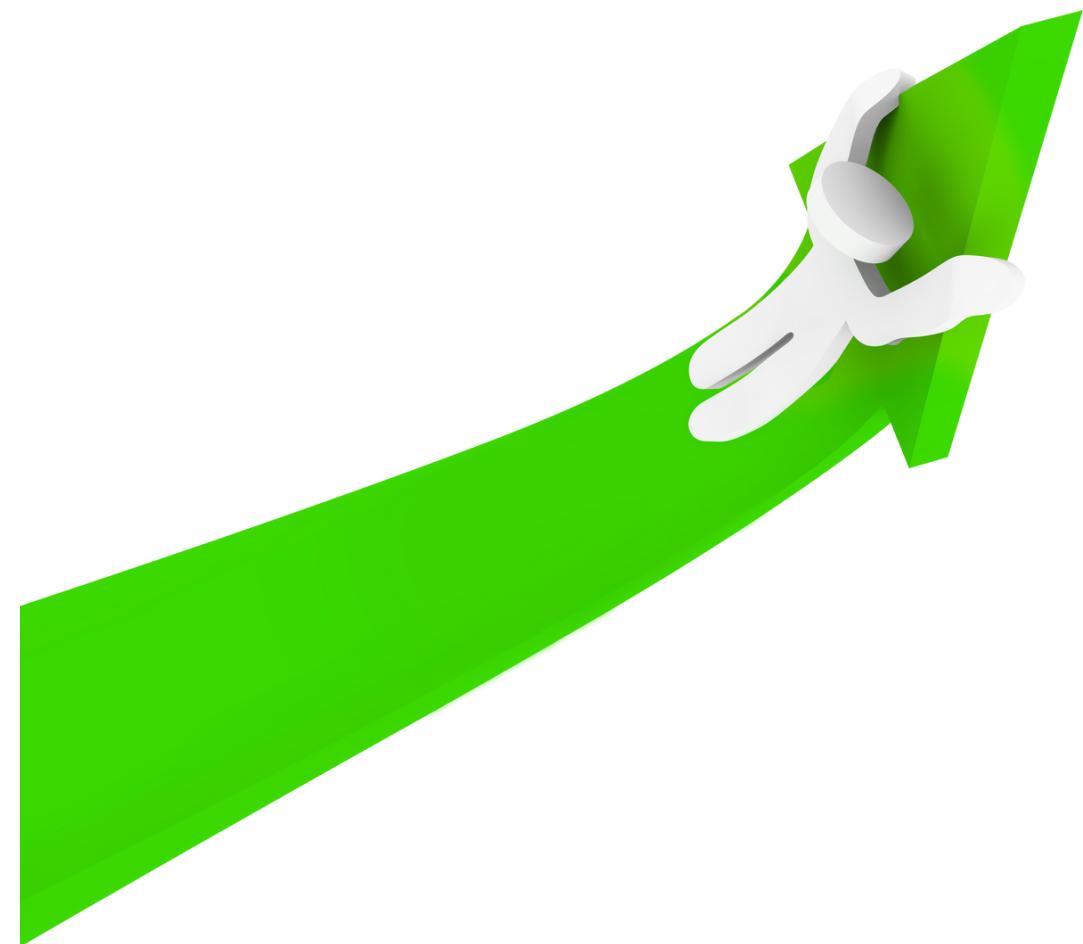
Sistem kapsam ve planlamanın diğer bir çıktısı da proje takımının izleyeceği belirli bir plandır. Bu proje planı standart SGYD'ni özelleştirir ve gerçekleştirilmesi için gereken zaman, para ve kaynakları belirler. İlk fazda oluşturulan en önemli çıktılardan biri iş beyanıdır. İş beyanı, bilgi sisteminin geliştirilmesini ifade eden bir sözleşme ya da anlaşmadır. Bu beyanda problemin tanımı, kapsamı ve projedeki tüm kaynakların genel zamanlama ve bütçesi yer alır.



Gereksinim Analizi

İkinci fazda sistem analisti sistem ihtiyaçlarını analiz eder. Analist, sosyal araç ve tekniklerin yardımıyla gereksinimleri tanımlar. Bu gereksinimler işlevsel, işlevsel olmayan ve ortam gereksinimleri olarak üçe ayrılabilir. İşlevsel gereksinimleri belirlemek için;

- Yeni sistem, kullanıcılarına hangi kabiliyetleri sunar?
- Hangi veri toplanmalı ve depolanmalıdır?
- Hangi performans düzeyi beklenmektedir?



Gereksinim Analizi

gibi sorulardan yola çıkarak, gereksinimler tanımlanır ve iş gereksinimleri önceliklendirilir. Ayrıca analist bu aşamada insan-bilgisayar etkileşimi ve sistemin çalışacağı ortamı ilgilendiren sorular ve cevapları da kullanır.

- Sistemi kanunlara uygun ve güvenilir yapmak için ne gereklidir?
- Yeni sistem, kullanıcı dostu olması için nasıl tasarlanmalıdır?
- Sistem, kullanıcıların iş görevlerini daha verimli yapacak özellikleri destekliyor mu?
- Sistemin çalışma ortamına göre desteklemesi gereken özellikler var mıdır?



Gereksinim Analizi

Bu soruların cevabı ise işlevsel olmayan ve ortam gereksinimlerini oluşturur. Sistem analisti, bu fazda bilgi kullanıcılarının işini yapması için nelerin gerekli olduğunu anlamaya çalışmaktadır. Mevcut sistem fonksiyonlarının ayrıntılarını öğrenmek için kim (sistemi kimler kullanıyor), ne (işletmenin faaliyetleri), nerede (ürünün çalışacağı yer), ne zaman (zamanlama) ve nasıl (mevcut iş süreçleri) sorularına cevap arar.



Gereksinim Analizi

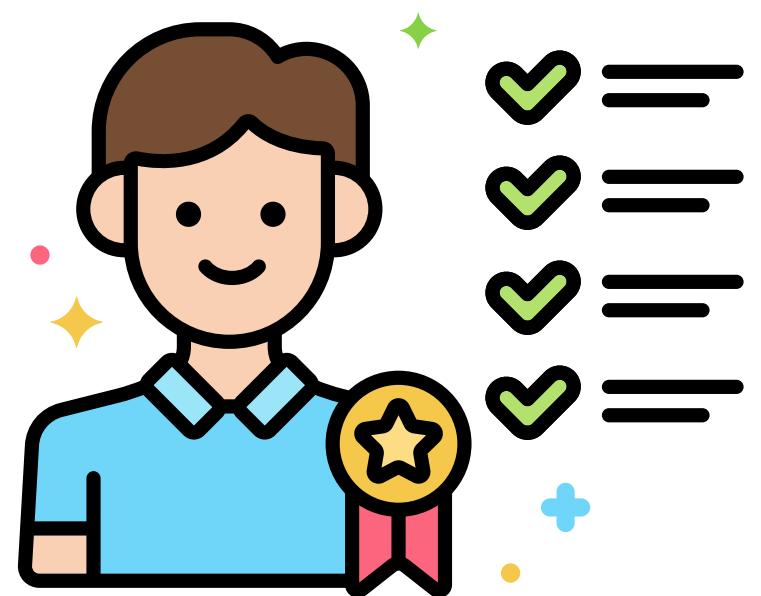
Bu aşamanın paydaşları proje yöneticileri, sistem analistleri ve sistem sahipleridir. Paydaşlara ayrıca sistem kullanıcıları da dahil edilebilir. Sistem tasarımcıları ve geliştirmeciler bu aşamada yer almaz.



Gereksinim Analizi

Bu fazda belirlenen gereksinimler önceliklendirilmelidir. Böylece proje zamanlamasında sıkışıklık oluşursa bu önceliklendirme kullanılabilir. Ayrıca ürünün **iteratif geliştirilmesi** hâlinde her bir iterasyonda tasarım ve geliştirmede kullanılacak gereksinimler bu önceliklere göre belirlenebilecektir.

İteratif geliştirmede, bilgi sisteminin birden fazla tekrar (iterasyon) geliştirme ile daha iyi hâle getirilmesi hedefenir.



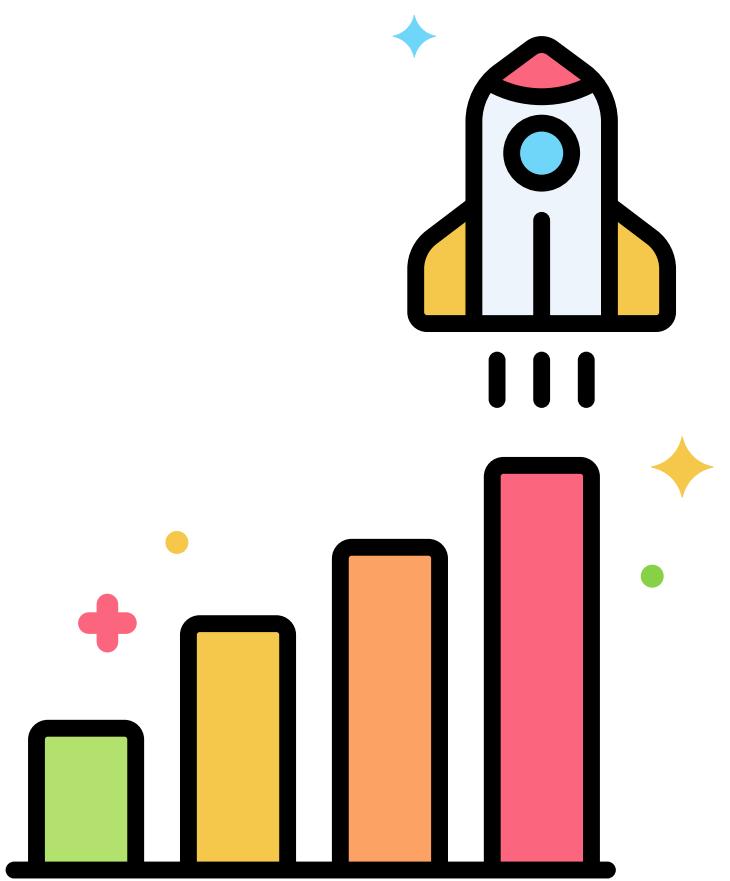
Gereksinim Analizi

Bu faz hiçbir şekilde atlanılmamalı veya kısaltılmamalıdır. Kullanıcı gereksinimleri- ni tam olarak anlamadan teknik çözümle uğraşan sistem tasarımcısı ve geliştiricilerinin oluşturduğu yeni sistem, kullanıcıların isteklerini karşılamayacak ve memnuniyetsizlik yaratacaktır.



Tasarım

SGYD'nin üçüncü fazı sistem tasarımıdır. Sistem tasarımında analist, alternatif çözüm önerilerini mantıksal ve fiziksel sistem özelliklerine dönüştürür. Sistemin girdileri, çıktıları, arayüzleri, raporları, veritabanları ve bilgisayar süreçlerinin tamamı bu fazda tasarlanır. Bir önceki fazda belirlenen iş gereksinimleri genellikle sözel ifadelerden oluşur. Sistem analistleri, kelimeleri sistem modelleri olarak adlandırılan diyagramlara dönüştürür ve gereksinimlerin tutarlığını ve bütünlüğünü test eder. Tasarım fazı Mantıksal tasarım ve Fiziksel tasarım olmak üzere iki aşamadan oluşur.



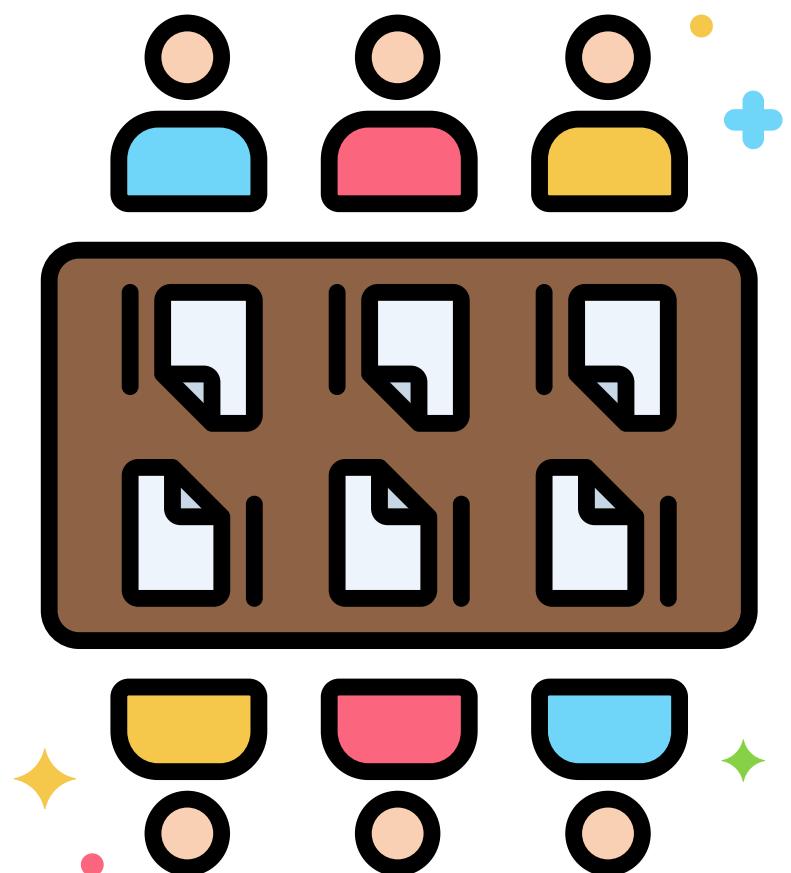
Tasarım

Mantıksal tasarım aşaması iş gereksinimlerini sistem modellerine dönüştürür. Daha önce belirlenen veri ve bilgi gereksinimlerinden mantıksal veri modelleri oluşturulur. İş süreç gereksinimlerinden mantıksal süreç modelleri, iş ve sistem arayüz gereksinimlerinden ise mantıksal arayüz modelleri oluşturulur. Bu modellerde genel sistemin kavramsal, etkinlik ve kullanım durum diyagramları kullanılır. Mantıksal tasarımda çözüm önerisi kullanılacak teknolojilerden bağımsız oluşturulur.



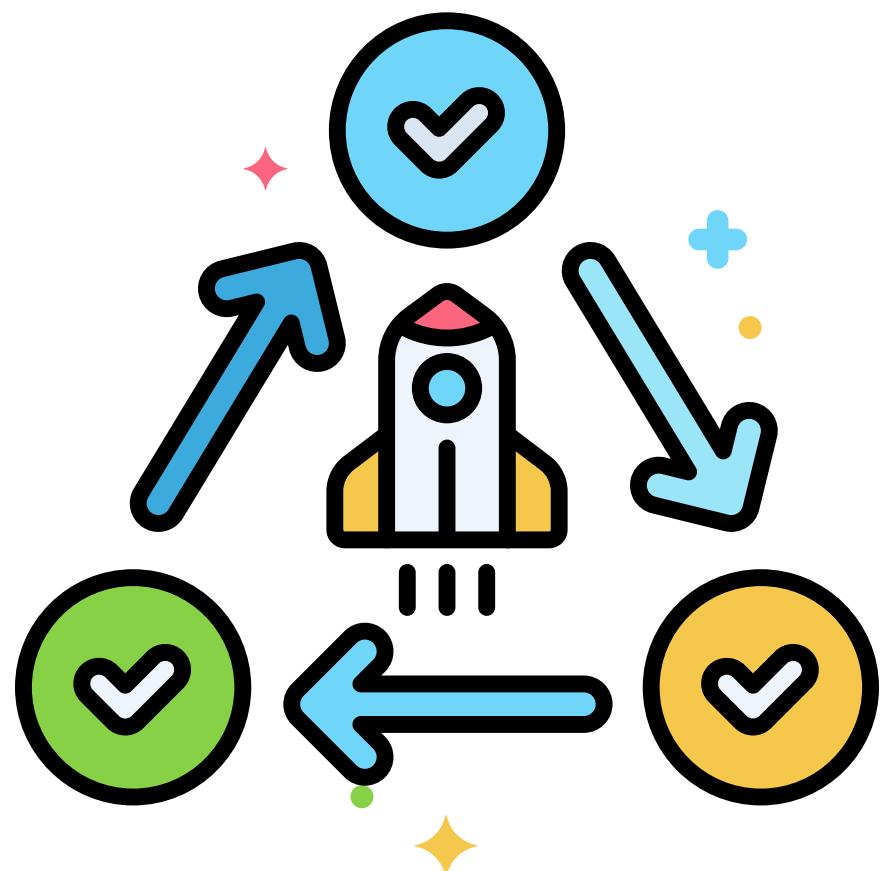
Tasarım

Fiziksel tasarıma geçmeden önce, verilen iş gereksinimleri ve mantıksal tasarım modelleri aşağıda anlatılan fizibilite sonuçlarına göre değerlendirilir ve bir mantıksal tasarıma karar verilir. İzleyen kesimde, çözüm önerilerinin hangi açılardan değerlendirildiğinin açıklandığı fizibilite türleri yer almaktadır.



Tasarım

Mantıksal tasarıma karar verildikten sonra yazılı ve/veya sözlü sistem önerisi oluşturulur. Bu öneri gerekirse genel sistem mimarisini de içerebilir. Sistemin mantıksal tasarım önerisinin onayından sonra Fiziksel tasarıma geçilir. Fiziksel tasarımda, Mantıksal tasarımlar fiziksel veya teknik özelliklere dönüştürülür.



Tasarım

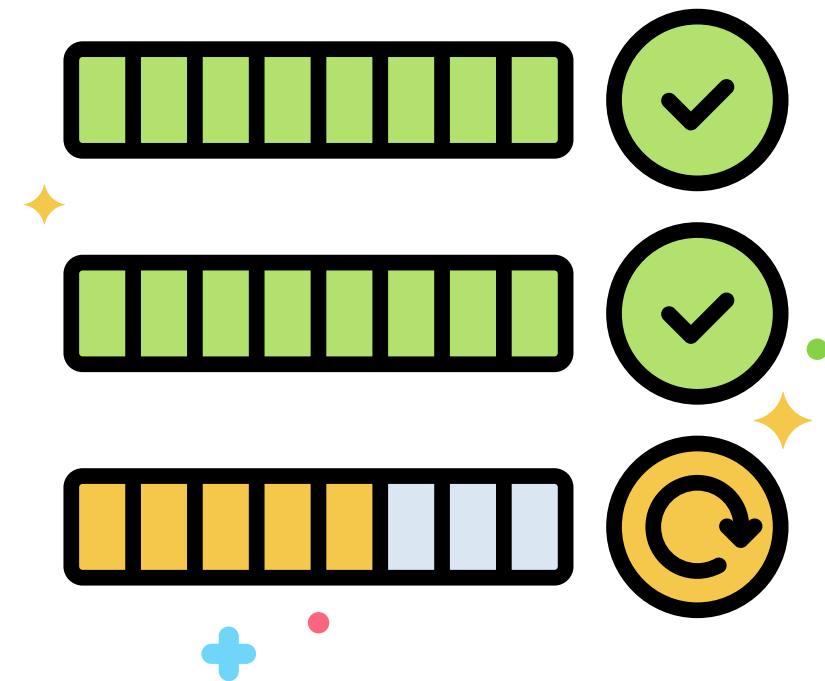
Tasarım fazındaki faaliyetler, farklı uygulamalar için geliştirilen bilgi sistemlerine göre değişiklik gösterir. Bilgi sistemlerinin tasarıımı aşamasında aşağıda verilen dört temel faaliyet gerçekleştirilir:



Tasarım

Yapısal tasarım: Sistem, sistemin temel (alt parçalar ya da modüller) parçaları, parçalararası ilişkiler ve dağılımını ifade eden bütün yapısı tanımlanır.

Arayüz tasarımı: Sistemin temel parçaları arasındaki arayüzler tanımlanır. Arayüz özellikleri kesin ve net olarak tanımlanmalıdır. Bu sayede bir alt sistem, nasıl geliştirildiklerini bilmenden diğer alt sistemlerle birlikte kullanılabilir. Arayüz özelliklerine karar verildikten sonra tüm bileşenler paralel olarak tasarlabilir ve geliştirilebilir.



Tasarım

Bileşen tasarımı: Her bir sistem bileşeninin tanımlanmış arayüz özelliklerine göre tasarlanması kapsar. Kullanılan geliştirme sürecine göre bu aşamada uygulamadan beklenilen fonksiyonları ifade etmek yeterli olabilir. Bileşen tabanlı sistem geliştirme modeli yaklaşımında ise tekrar kullanılabilir bileşenlerde yapılacak değişiklikler ve yeni tasarlanacak bileşenler ayrıntılı olarak tanımlanır.

Veritabanı tasarımı: Sistem veri yapısı tasarlanır ve bunların veritabanında karşılık gelen yapıları gösterilir.



Tasarım

Bu fazın paydaşları sistem tasarımcıları, sistem analistleri ve iş akışını paylaşmak üzere sistem kullanıcılarıdır. Tasarım sürecinin çıktıları, sistem geliştiricilere ve programcılara verilebilecek, sistem özelliklerini anlatan diyagram ve/veya yazılı raporlardan oluşur.



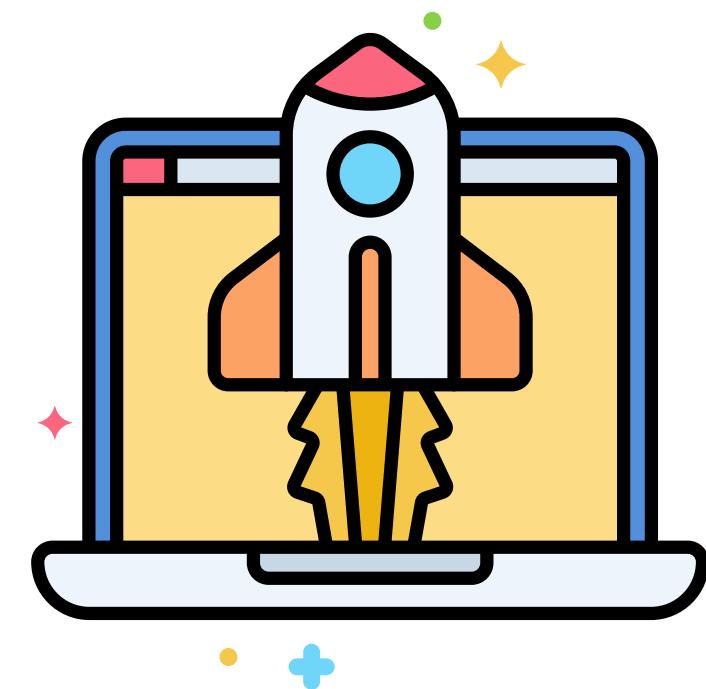
Gerçekleştirme ve Test

Sistem gerçekleştirme ve test fazında, özellikleri ortaya konulmuş olan sistem geliştirilir ve testleri tamamlanarak çalışır hale getirilir. Programlama sırasında, programcılar sistemi oluşturan yazılımı ve eski sistem ile yeni sistem arasındaki arayüzleri geliştirir. Gerçekleştirmenin en önemli yönlerinden biri olan testler sırasında da programcılar ve analistler program parçalarında ve tüm sistemde oluşabilecek hataları bulmak ve düzeltmek üzere testler uygular. Sistem, kullanıcılarla teslim edilmeden önce hataların yakalanması halinde, hataların çözümü çok daha az maliyetli olacaktır. Testler başlangıçta örnek veriler ile yapılırken teslimden önce testler gerçek veriler ile tamamlanır.



Gerçekleştirme ve Test

Bu fazın sonunda proje ekibi, kurulum ve dağıtım için çalışan fonksiyonel bir sistem ortaya koyar. Veri tabanları, ticari yazılım paketleri ve/veya özel geliştirilmiş yazılımlar ve kullanıcı-sistem arayüzleri bu sistemin parçalarını oluşturur. Testler tamamlandıktan sonra sistem kurulum ve dağıtıma hazırdır.



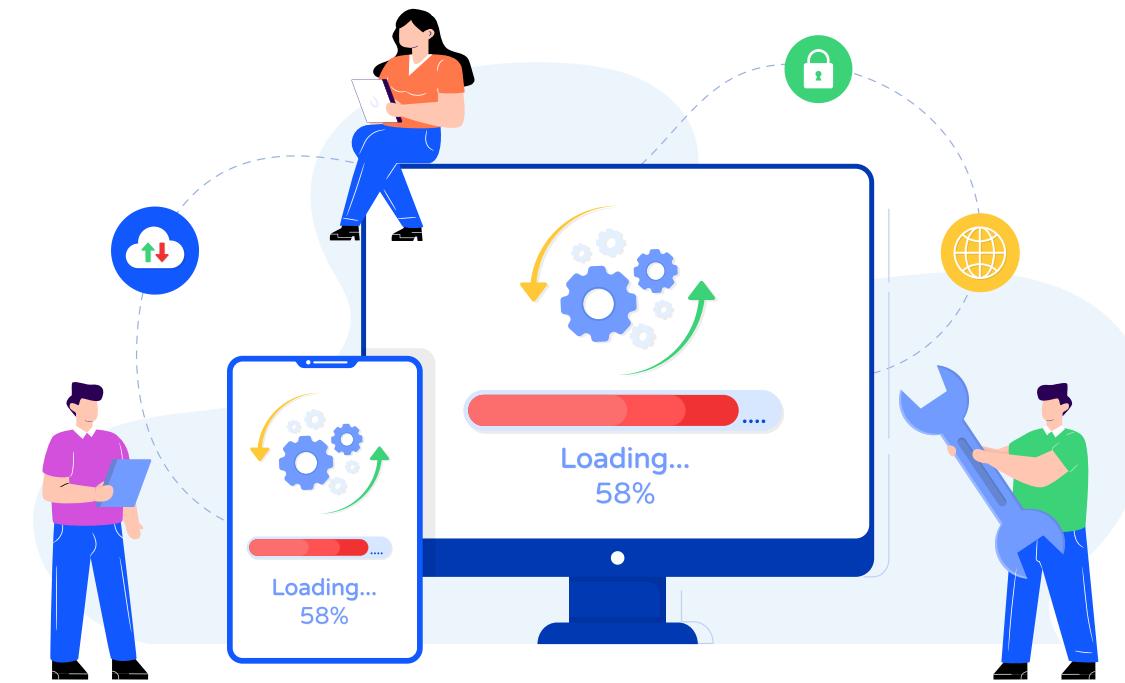
Kurulum ve Dağıtım

Gerçekleştirme ve test fazında oluşturulan fonksiyonel sistem, kurulum ve dağıtım fazının girdisini oluşturur. Sistem geliştiricileri, uygulama yazılımını, geliştirme ortamından üretim ortamına geçirmek üzere mevcut ya da yeni donanım üzerine kurarlar. Sistem analistleri sistem kullanıcılarını eğitir, birçok kullanıcı el kitaplarını ve üretim kontrol el kitaplarını oluşturur, eski veritabanının yeni veri tabanına aktarımını ve en son sistem testlerini yapar. Herhangi bir problem oluşması hâlinde hatayı düzeltmek üzere önceki fazlardan tekrar çalışma başlatılması gerekebilir. İşveren yeni sistemler için bazı eğitimler düzenleyebilir, ancak eğitimin gözetimi sistem analistinin sorumluluğundadır.



Operasyon ve Bakım

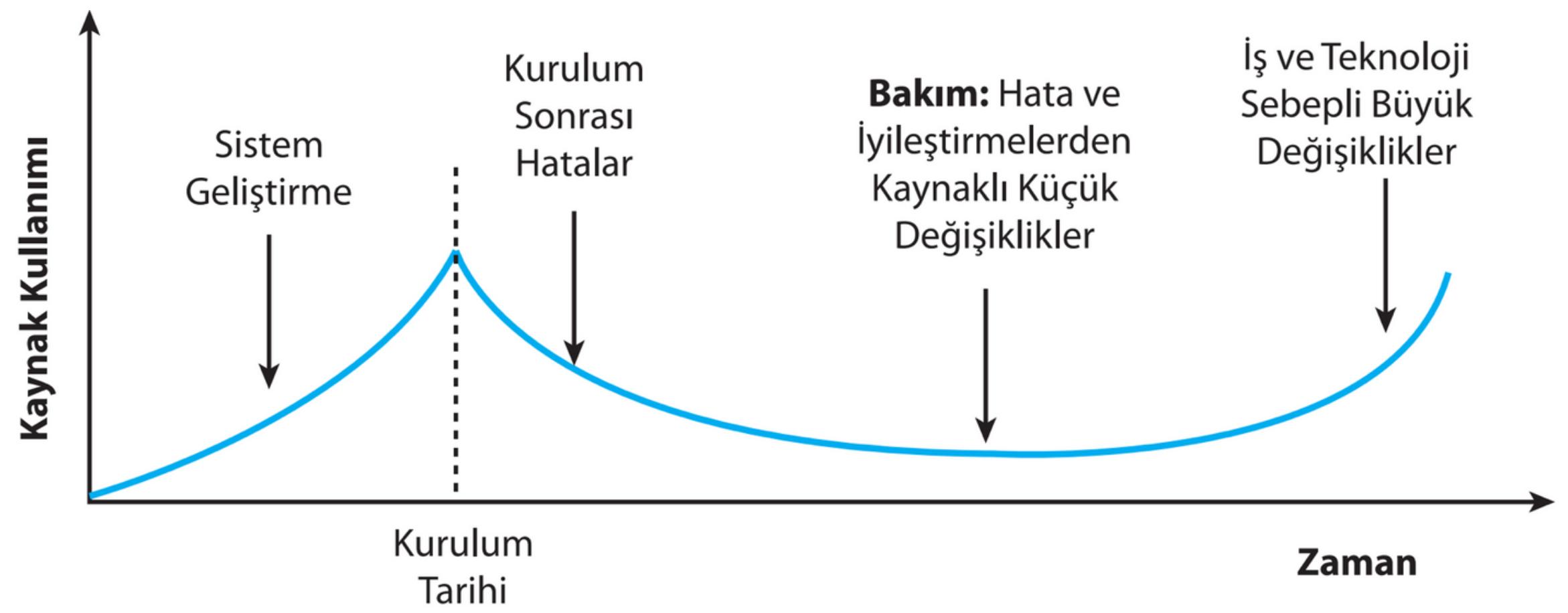
SGYD'nin son fazı Operasyon ve Bakımdır. Sistemin kurulup çalışmaya başladığı bu evre en uzun yaşam döngüsü fazıdır. Sistem işletmelerde çalışmaya başladıkтан sonra kullanıcılar bazen işleyiş ile ilgili problemler olduğunu fark eder, yaşam döngüsünün erken fazlarında görülmeyen problemler ortaya çıkar ya da iyileştirme yapmayı talep eder. Bu nedenle sistemin kullanıldığı süre içerisinde, programcılar sistem desteği sağlamak üzere kullanıcıların talep ettiği, iş süreçlerindeki değişimler doğrultusunda sistemi günceller.



Operasyon ve Bakım

Sistem Yaşam
Döngüsü Süresince
Kaynak Kullanımı

Kaynak:Kendall K.E.,
Kendall J. E. (2006).
s. 13.



Bu tip problemler artık bilgi sisteminin değiştirilme zamanı geldiğini gösterir. Operasyondaki bilgi sisteminin yaşam döngüsü sonlanmış ve yeni bir sistem için yeni bir yaşam döngüsüne başlamanın zamanı gelmiştir.

DOĞRUSAL ARDIŞIK MODEL

Bölüm 3



Bilgi Sistemi Geliştirme Modelleri

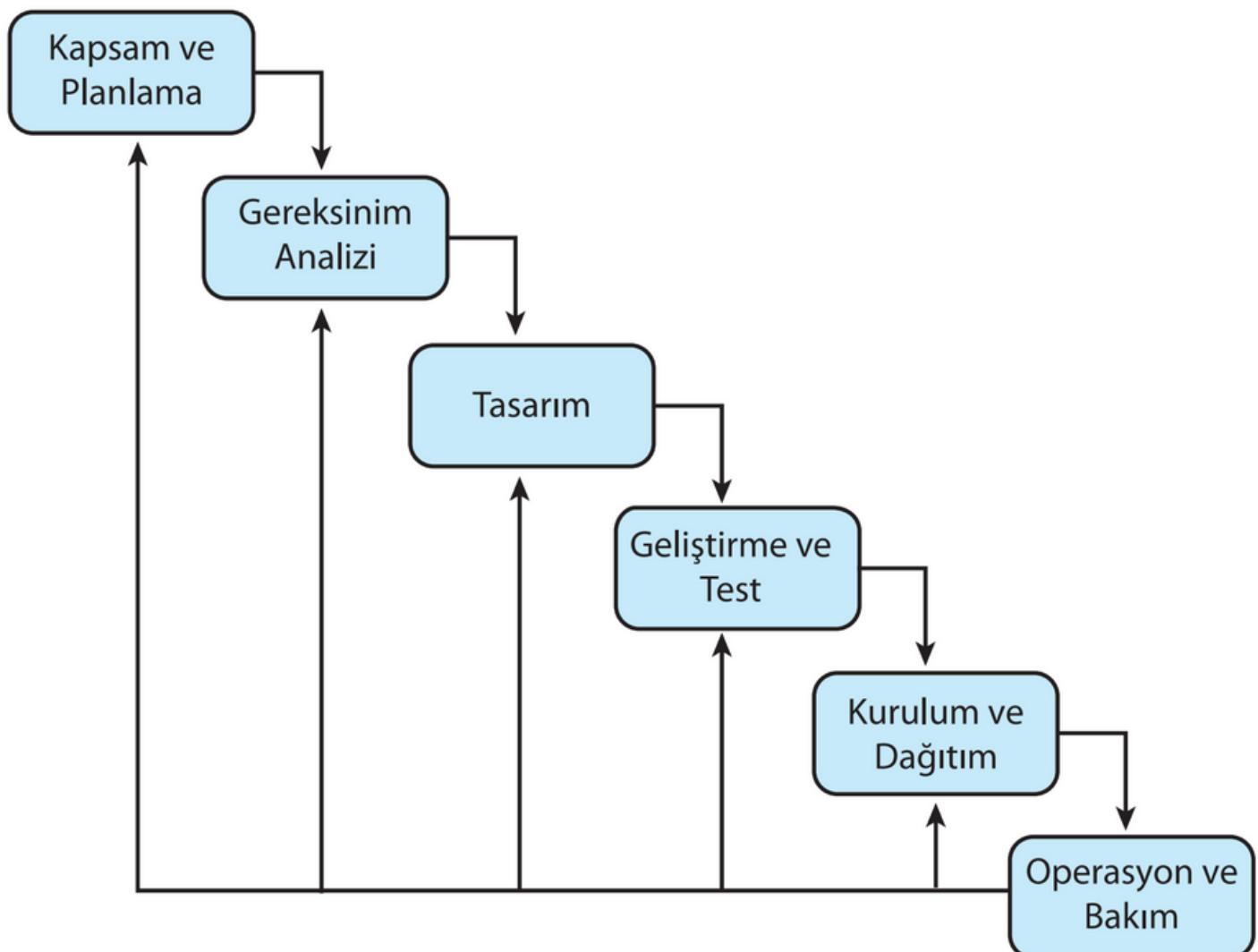
Sistem analizi ve tasarım süreçlerini farklı yaklaşımlar ile gerçekleştiren birçok model vardır. Bu modeller ile farklı uygulama alanlarındaki bilgi sistemlerinin uygun sistem analizi ve gerçekleştirme araçları ile geliştirilmesi desteklenmiş olur. Sistem geliştirme yazılım modellerinden Doğrusal Ardışık Model, Prototip Model, Artımlı Model, Spiral Model, Bileşen Tabanlı Geliştirme Modeli ve Çevik Model izleyen kesimde açıklanacaktır.



Doğrusal Ardışık Model

Sistem analizi ve tasarımında kullanılan Doğrusal ardışık model yaklaşımında fazlar sadece bir defa olmak üzere sıra ile işletilir. Doğrusal ardışık model aynı zamanda **Şelale Modeli (WaterFall Model)** olarak da isimlendirilir. Şekilde gösterildiği gibi Doğrusal ardışık model bu ünitenin ilk kesiminde açıklanmış olan genel SGYD sürecinden oluşmuştur.

Doğrusal Ardışık Model



Doğrusal Ardışık Model

Modelin ana aşamaları bilgi sisteminin doğrudan temel geliştirme faaliyetlerini göstermektedir.

i. Kapsam ve Planlama: Sistemin sağlayacağı hizmetler, kısıtları ve amaçları sistem kullanıcıları ile belirlenir.

ii. Gereksinim Analizi: Sistem geliştirme hedefi ile iş gereksinimleri kontrol edilir.

Her bir gereksinim, bir veya daha fazla sistem geliştirme hedefine katkıda bulunacak şekilde ayrıntılı olarak tanımlanır ve tüm sistemin özellikleri sistem kullanıcıları ile gözden geçirilir.

Doğrusal Ardışık Model

- iii. Tasarım: Sistem tasarıımı, gereksinimlerin tamamını karşılayacak donanım ve yazılım sistemlerinden oluşan sistem mimarisini oluşturur. Tasarım, geliştirilen bilgi sistemi yazılımının dört ayrı özelliğine odaklanır: veri yapısı, yazılım mimarisi, arayüz gösterimleri ve algoritma yapılandırması.
- iv. Geliştirme ve Test: Bu aşamada yazılım tasarıımı, programlar veya program parçaları olarak gerçekleştirilmektedir. Program testleri yazılım gerçekleştirmeye ile başlar. Testler, yazılımın tüm gereksinimleri karşıladığını göstermek ve olası hataları bulmak için uygulanır.

Doğrusal Ardışık Model

v. Kurulum ve Dağıtım: Sistem mimarisi çerçevesinde oluşturulmuş olan program parçaları bir araya getirilerek tüm sistem gereksinimlerinin karşılandığı test edilir. Testler sonrasında geliştirilen sistem, müşterinin kullanacağı şekilde kurulur ve uygulamaya alınır.

vi. Operasyon ve Bakım: Normalde bu SGYD içinde en uzun yaşam döngüsüne sahip fazdır. Sistem kurularak kullanımına başlanır. Bakım aşaması daha önceki keşfedilmemiş olan hataların düzeltilmesi, sistemin iyileştirilmesi ve sistemde oluşturulmak istenen yeni özelliklerin gerçekleştirilmesini içerir. Yazılım bakım aşamasındaki değişiklik gereksinimleri için mevcut programda uygulanmış olan önceki fazlar tekrar uygulanmaktadır.

Doğrusal Ardışık Model

Doğrusal ardışık model, her bir fazında gerçekleştirilen faaliyetler ve belgeleme ile yöneticilerin proje planlarına göre ilerlemeyi izlemelerine imkân sağlar. Bununla birlikte projenin esnek olmayan şekilde alt fazlara bölünmüş olması en önemli problemi oluşturur. Bu nedenle Doğrusal ardışık model en eski ve en yaygın kullanılan model olmakla birlikte, verimliliği en aktif destekleyenler tarafından bile sorgulanmaktadır.

Doğrusal Ardışık Model

Doğrusal ardışık modelin uygulanmasında ortaya çıkan bazı problemler aşağıda sıralanmıştır:

- i. Gerçek projelerde modelde kullanıldığı gibi ardışık bir sıralama nadiren uygulanır. Değişiklikler her aşamada ortaya çıkabilir. Proje takımının değişiklikleri yönetmesi zordur ve yüksek oranda kaynak (zaman, maliyet) tüketimine sebep olur.
- ii. Bazı sistemlerde müşteri tüm gereksinimleri en baştan ayrıntılı olarak belirtemeye yebilir. Oysa Doğrusal ardışık model tüm gereksinimlerin en başta tüm ayrıntılarıyla tanımlanmasını gerektirir ve başlangıçta olabilecek belirsizliklerin yönetimi zordur.

Doğrusal Ardışık Model

- iii. Müşteri sistemin gerçekleştirilerek uygulamaya alınmasında sabırlı olmak zorundadır. Sistemin çalışan sürümü projenin gerçekleştirme sürecinin sonlarına doğru hazır olur. Çalışan programın gözden geçirilmesinde ortaya çıkabilecek ana sorun projenin başarısızlığına dahi sebep olabilir.
- iv. Yaşam döngüsünde kullanılan ardışık yaklaşım, bazı proje elemanlarının diğer fazlardan gelecek olan çıktıları beklemesinden dolayı bloklanmasına ve üretim için harcayacağı zamanları israf etmesine sebep olabilmektedir.



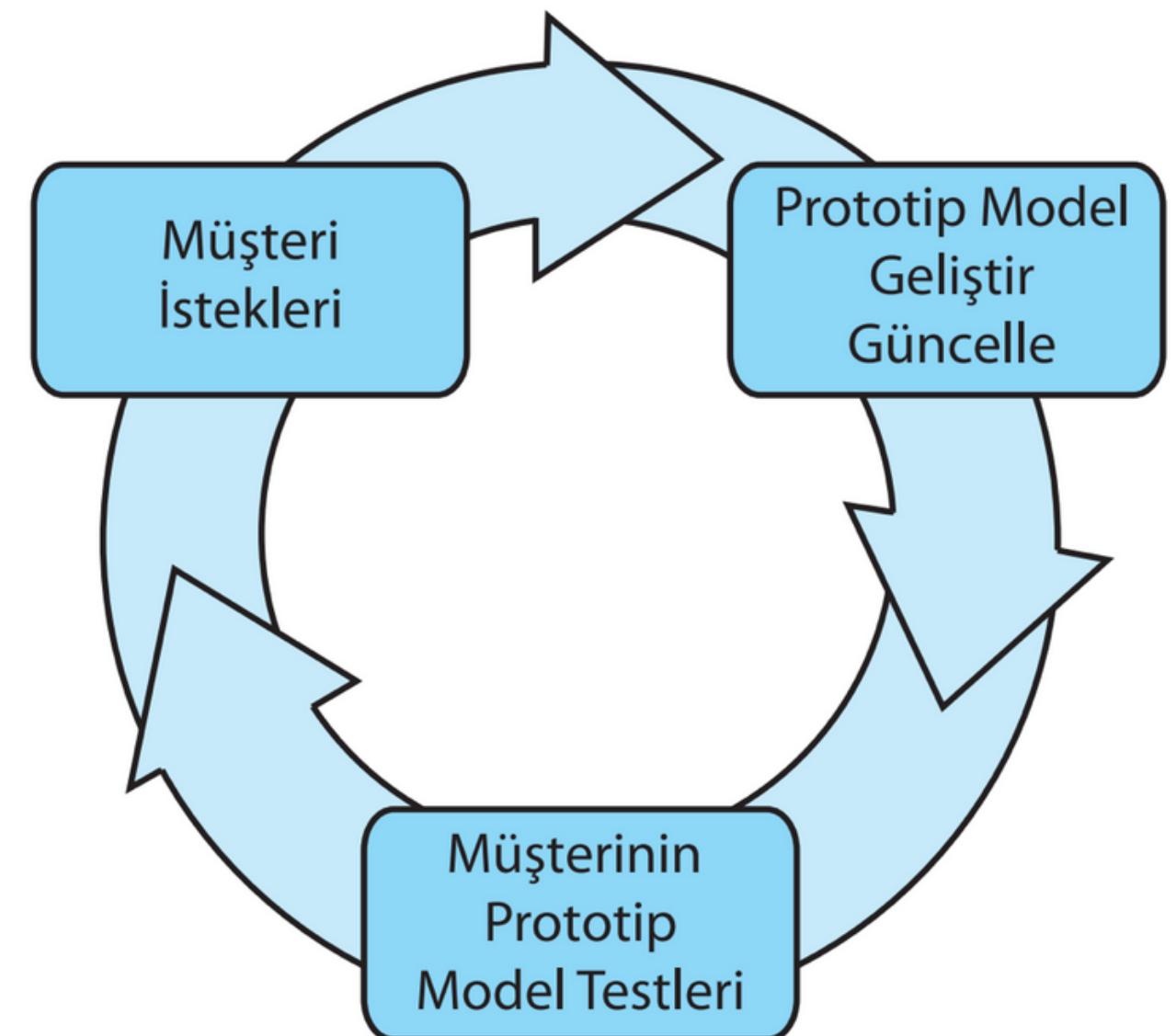
Prototip Model

Bazı projelerde müşteri, sistemin genel hedeflerini tanımlar fakat ayrıntılı girdi, iş süreçleri veya çıktı gereksinimlerini belirtmez. Bazı durumlarda da geliştirme ekibi kullanılacak işletim sisteminden, yazılım verimliliğinden ve insan makine arayüzlerinin nasıl olması gerekiğinden emin olmayabilir. Bu ve benzeri durumlarda gereksinim oluşturmada prototip yaklaşımı en uygun yöntem olacaktır. Prototip modelde ilk adım müşteri istekleri doğrultusunda gereksinimlerin belirlenmesidir. Geliştirme ekibi ve müşteri birlikte yazılımın tüm hedeflerini tanımlar, mümkün olabildiği kadarıyla gereksinimleri belirler ve açık kalan alanları kayıt altına alır.



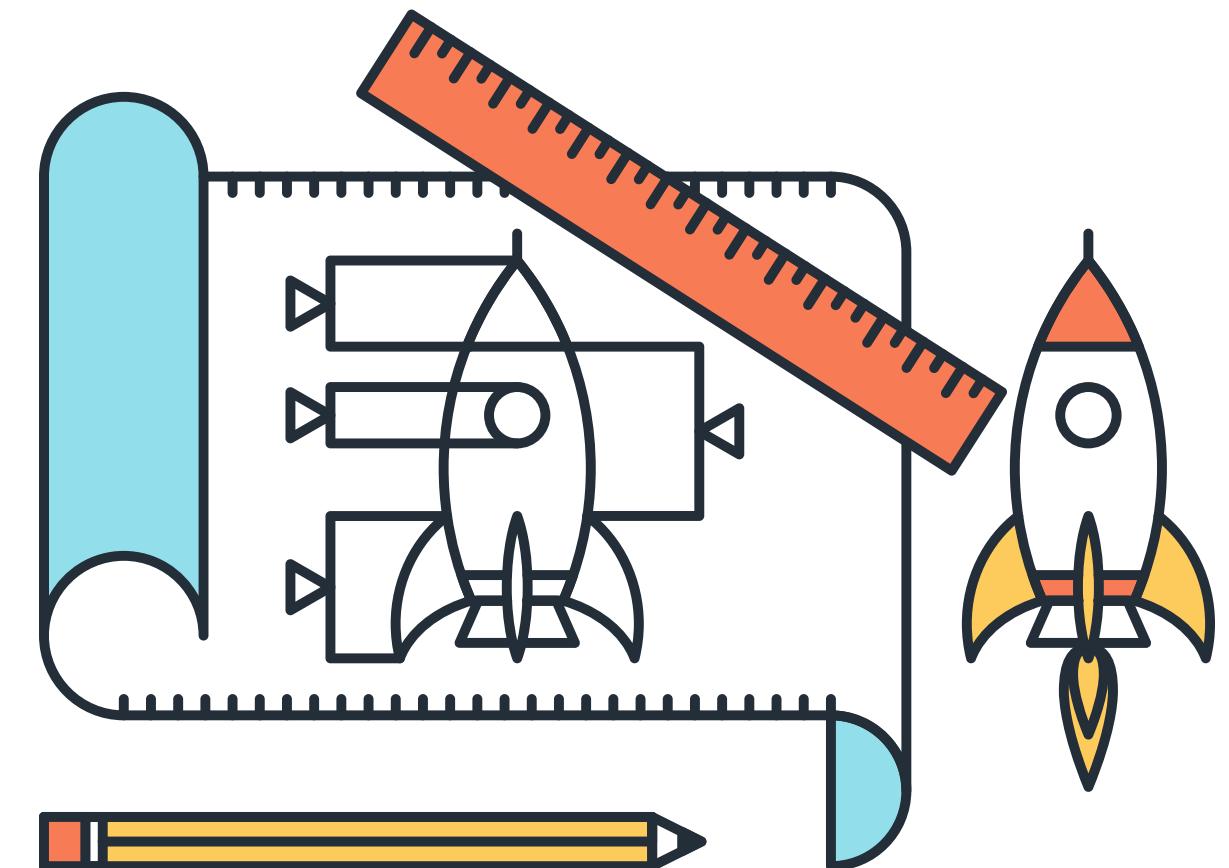
Prototip Model

Devamında sistemin kullanıcılar tarafından görülür kısımları için hızlı bir tasarım yapılır. Hızlı tasarımın devamında prototip geliştirilir. Prototip, kullanıcılar tarafından değerlendirildikten sonra gereksinimler tekrar gözden geçirilir ve süreç tekrar başlar. Prototip modelde (Şekil) iterasyon (döngü), ürünün gelişimi müşterinin gereksinimlerini oluşturuncaya ve geliştiricilerin ne yapılması gerektiğini anlamasına kadar devam eder.



Prototip Model

İdealde prototip, sistemin tüm gereksinimlerinin belirlenmesine yardımcı olur. Prototip modelde, başlangıçta müşteri ile geliştirilen prototipin gereksinimleri tanımlamak için bir mekanizma olduğu konusunda anlaşılmalıdır. Gereksinimler açık olarak tanımlandıktan sonra prototip atılmalı ve gerçek yazılım, kalite, bakım ve yapılabılırlik özellikleri ile yeni baştan ele alınmalıdır.



ARTIRIMLI MODEL

Bölüm 4

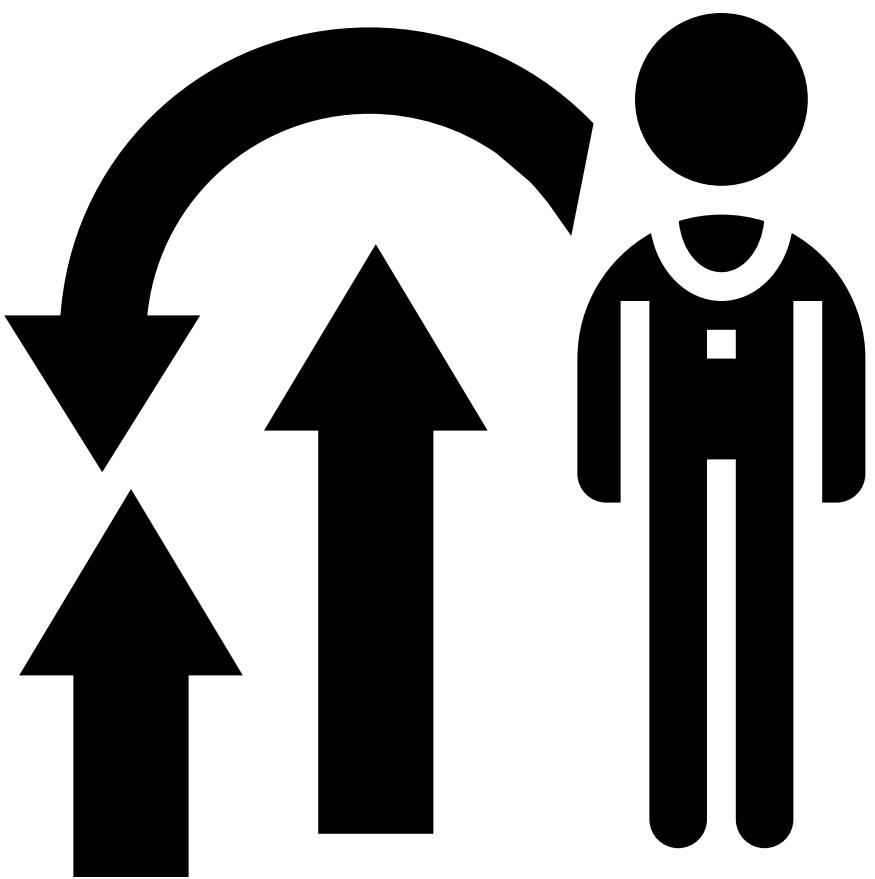
Artırımlı Model

Artırımlı model (Incremental model), ilk uygulamanın oluşturularak kullanıcı görüşüne başvurulması ve istenilen son sistem geliştirilinceye kadar birçok sürüm oluşturulması fikrine dayanır. Tasarım ve Özellikleri oluşturma, Geliştirme ve Testler ayrı ayrı yapılmak yerine her bir ara sürüm için ayrı ayrı ve paralel yapılır.



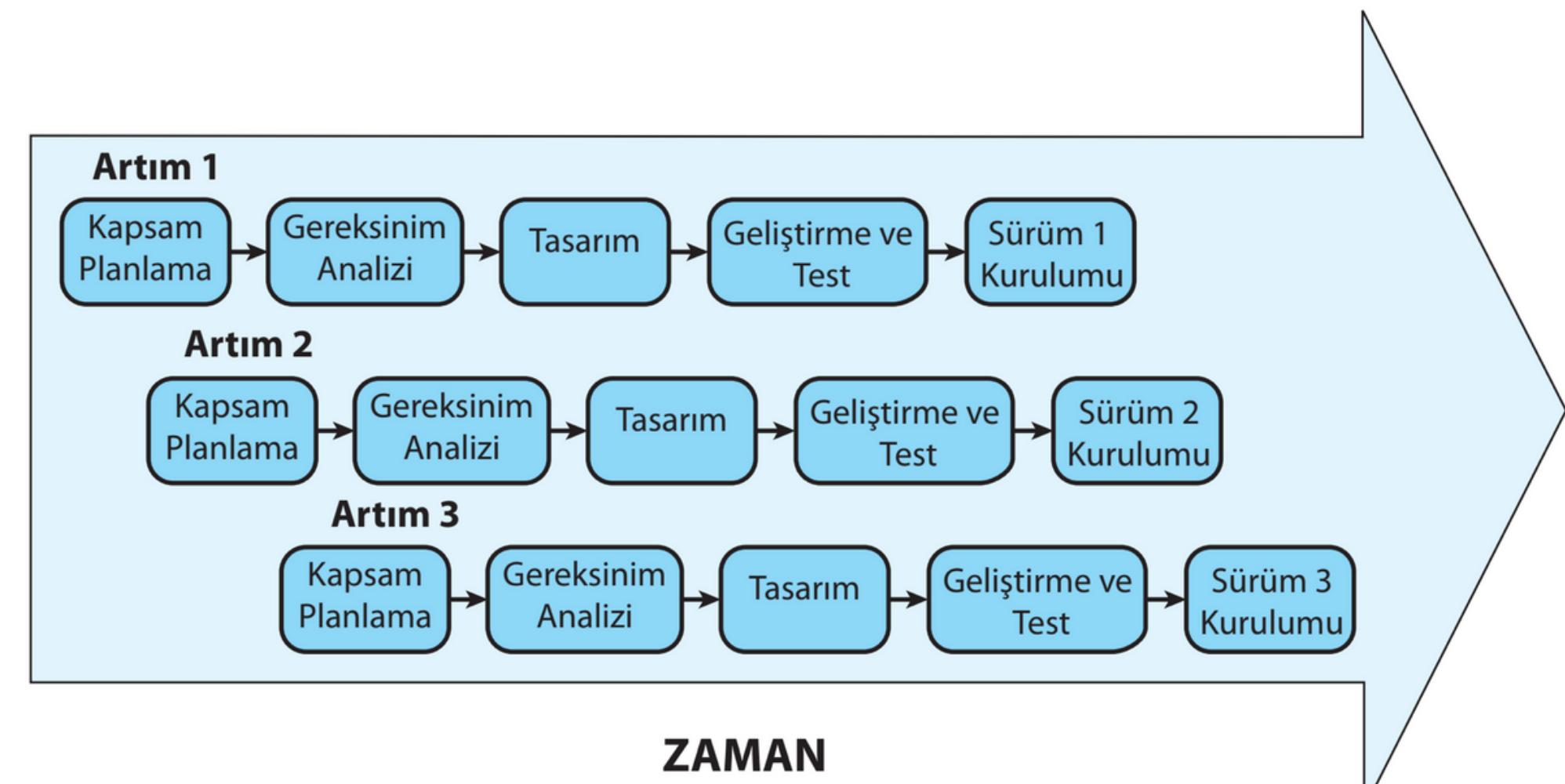
Artırımlı Model

Artırımlı model, Doğrusal ardışık model ile Prototip modeldeki iteratif yaklaşımın birleşimidir. İteratif yapısıyla e-ticaret ve masaüstü sistemlerin geliştirilmesinde Doğrusal ardışık modelden daha iyi bir yaklaşımdır. Artırımlı model geliştirilen sistemi birçok alt sürümden oluşacak şekilde Doğrusal ardışık modeldeki yaklaşımı kullanarak zaman içinde aşamalı olarak gerçekleştirir.



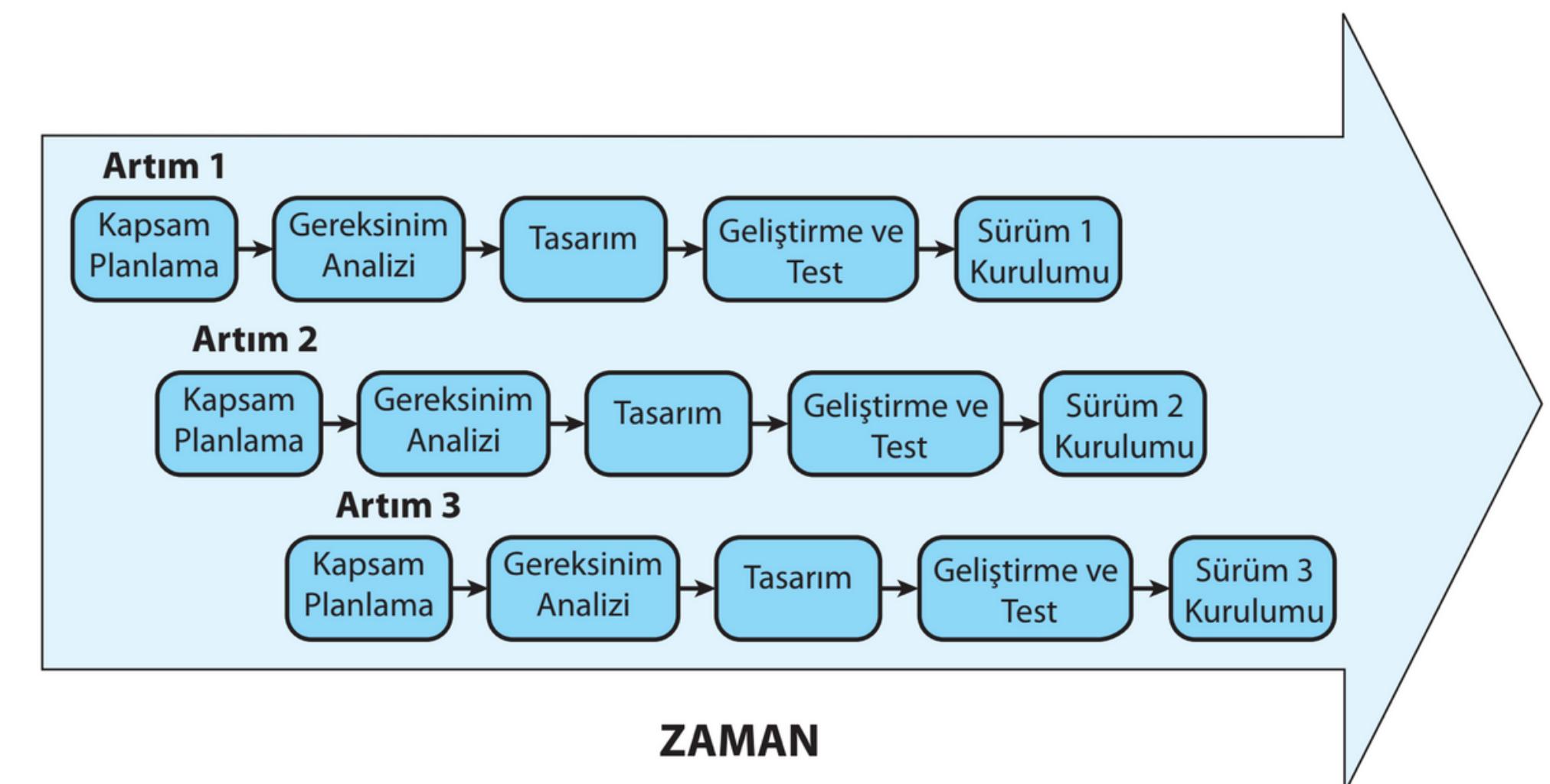
Artırımlı Model

Şekilde gösterildiği gibi her bir alt sürüm bir artımı oluşturur. Her bir yeni sürüm müşterinin ihtiyacı olan bir özelliği sistem içine ekler. Genelde sistemin ilk iterasyonları daha önemli ve acil özellikleri içerir. Yazılımın artan şekilde geliştirilmesi tasarım ve geliştirme aşamalarında değişikliklerin daha kolay ve az maliyetli yapılmasını sağlar.



Artırımlı Model

Bu iteratif yaklaşım, bir sürümde sisteme eklenilen özellik istenileni sağlamıyorsa izleyen sürümde değişiklik yapma ve ihtiyaçları karşılayan bir sürüm oluşturulmasına imkân verir.



Artırımlı Model

Doğrusal ardışık model ile kıyaslandığında Artırımlı model bazı önemli kazanımlar sağlar. Bunlar;

- i. Müşteri gereksinimlerinde değişikliğin maliyeti (analiz, tasarım, testler ve belgeleme) daha düşük olur.
- ii. Geliştirme aşamasında müşteri geri dönüşü daha kolay alınmış olur. Müşteri, tasarım belgeleri yerine yazılımın çalışan bir sürümü üzerinden daha kolay yorum yapar.



Artırımlı Model

- iii. Yazılımın tamamını olmasa da önemli özelliklerini taşıyan ilk sürümünü müşterinin kullanımına sunmak daha hızlı olacaktır. Böylece müşterinin yazılımı önceden kullanmaya başlaması iş süreçlerinde önemli kazanımlar elde etmesini sağlayabilir.
- iv. Sistemin son tarihinde teslim için yeterli sayıda geliştirme personeli yok ise, ilk sürümler (temel ürün) daha küçük bir ekip ile istenilen zamanda geliştirilir. Bir kez temel ürün geliştirildikten sonra gerekirse ekip büyütülerek artım ile izleyen sürümler gerçekleştirilir.



Artırımlı Model

Artırımlı model ile geliştirme, Doğrusal ardışık model yaklaşımıyla, Çevik model yaklaşımıyla (izleyen kesimde anlatılacak) ya da her ikisinin karışımı ile gerçekleştirilebilir. Doğrusal ardışık model yaklaşımıyla sistemin her bir artımı önceden belirlenirken, Çevik model yaklaşımında ilerleyen sürümlerin geliştirilmesi süreç ve müşteri önceliklerine göre gerçekleştirilir.



Artırımlı Model

Bununla birlikte Artırımlı modelin kullanımı da bazı problemler oluşturabilir:

- i. Süreç izlenemez. Yöneticiler ilerlemeyi izlemek için ara çıktılar ihtiyaç duyar. Sistemler çok hızlı geliştirildiğinden, erken ve ara sistem sürümlerini gösteren belgeleri uygun maliyetlerle üretmek mümkün değildir.
- ii. Sistem, sürümler ilerledikçe yapısal olarak geriler. Sistemde sürekli değişikliklerin uygulanması yapısal bozulmaya yol açar. Bu nedenle sistemin yapısal iyileştirilmesi için zaman ve para harcanması gereklidir. Aksi hâlde değişiklik uygulamasının sebep olduğu yapısal bozukluklar, yeni değişikliklerin uygulanmasının zorluğunu ve maliyetini artırır.



Artırımlı Model

- iii. Sistemlerin çoğu, alt parçalarında bazı ortak imkân ve kaynakları kullanır. Bütün gereksinimlerin başlangıçta açık olarak bilinmediği durumlarda, bu ortak imkânların öngörülmesi zor olacaktır.
- iv. Mevcut sistemin yerine yapılan sistemlerin geliştirilmesinde iteratif yaklaşımı kullanmak zor olabilir. Sistem kullanıcıları, mevcut/eski sistemin tüm özellikleri olmadan, kısıtlanmış özellikleriyle yeni sistemde çalışmayı istemez. Bu yüzden de müşteri geri beslemesini almak zorlaşır.



SPIRAL MODEL

Bölüm 5

Spiral Model

Spiral model, Prototip modelin iteratif yapısının, Doğrusal ardışık modelin sistematik yaklaşımıyla kontrol edilmesini sağlayan evrimsel bir modeldir. Sistem yazılımının artan sürümlerinin hızlı geliştirilmesine olanak tanır. Bu modelde de yazılım artan sürümler hâlinde geliştirilir. İlk iterasyonlardaki artan sürümler kâğıt üzerindeki gereksinimleri veya prototipi oluştururken ilerleyen iterasyonlarda sistemi tam olarak gerçekleyen artan sürümler geliştirilmiş olur. Spiral modelde sistem geliştirme tipik olarak 3 ile 6 arasında değişen sayıda bölümden oluşan çerçeve faaliyetlerine bölünmüştür.

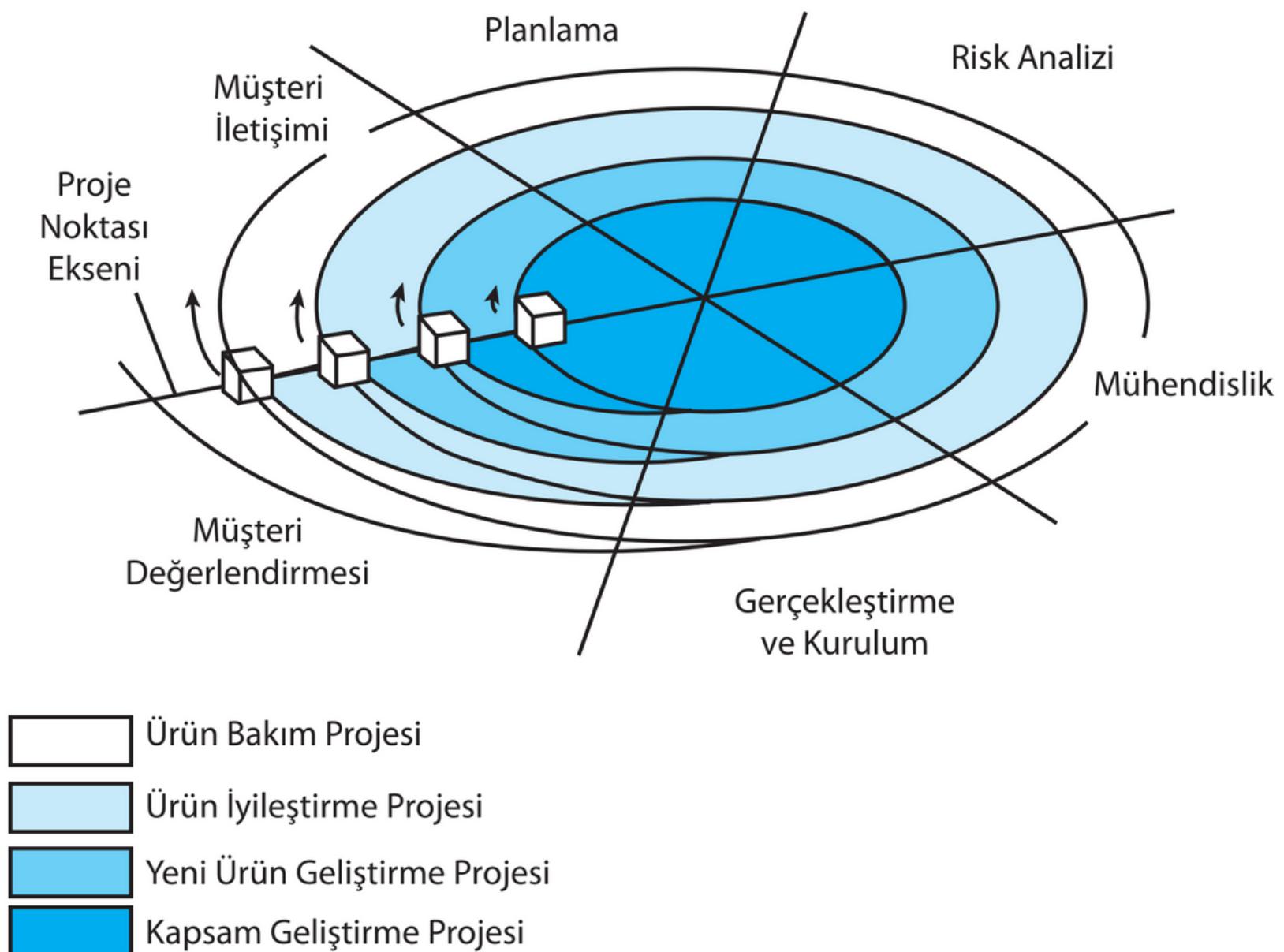


Spiral Model

Müşteri iletişim: Geliştirici ve müşteri arasında etkin bir iletişim kurulması görevlerini,

Planlama: Kaynakların, zaman çizelgelerinin ve proje ile ilgili diğer bilgilerin tanımlanması görevlerini,

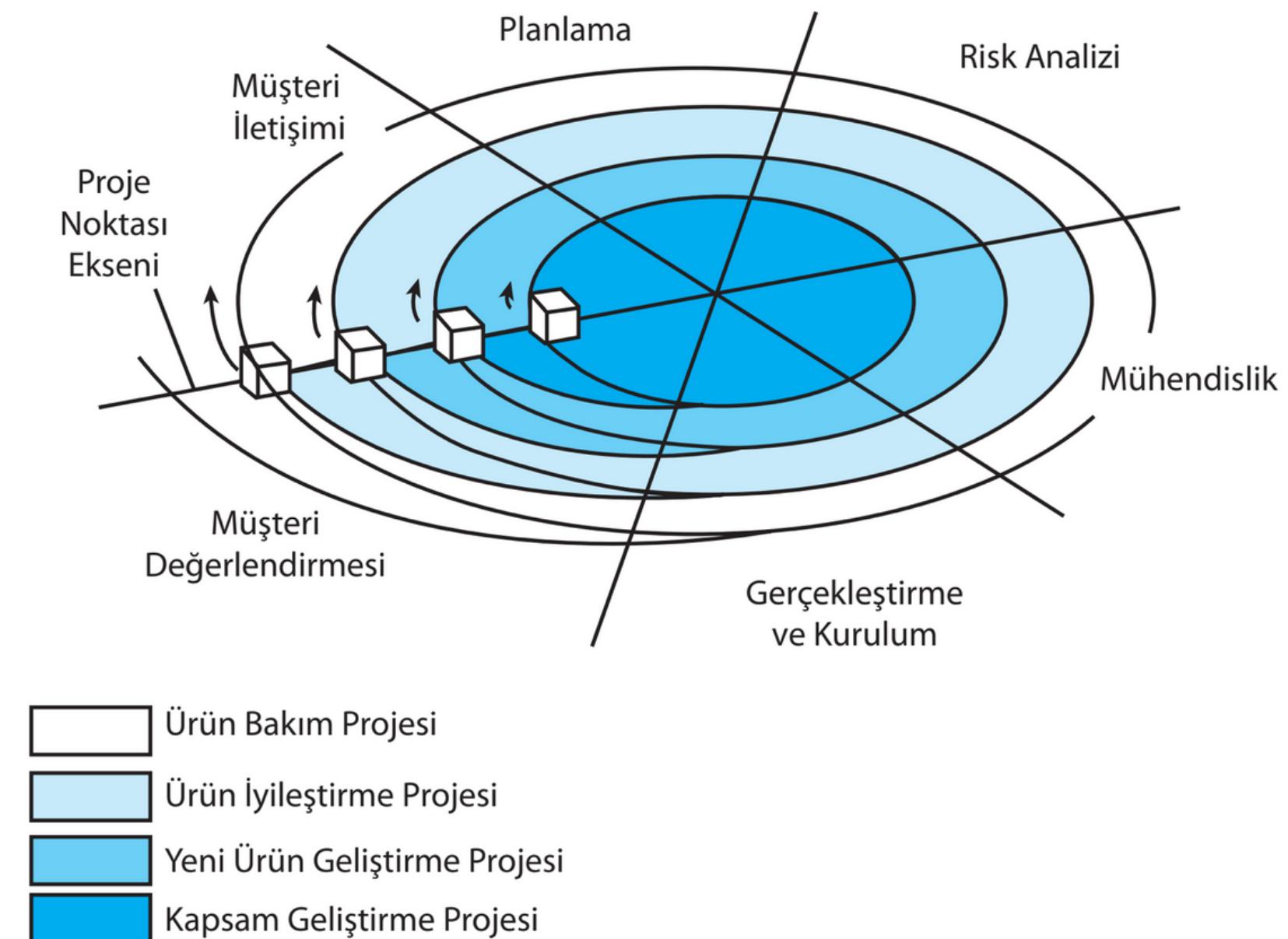
Risk Analizi: Teknik ve yönetim risklerinin belirlenmesi görevlerini,



Spiral Model

Mühendislik: Uygulamanın bir veya daha fazla çözümünün oluşturulması görevlerini,
Gerçekleştirme ve Kurulum: Sistemin inşa edilmesi, test, kurulum ve kullanıcı desteğini (belgeleme, kullanıcı eğitimleri) oluşturma görevlerini,

Müşteri Değerlendirmesi: Mühendislik çözümleri, gerçekleştirme ve kurulum sonrasında sistem hakkında müşteri değerlendirmesinin alınması görevlerini içerir.



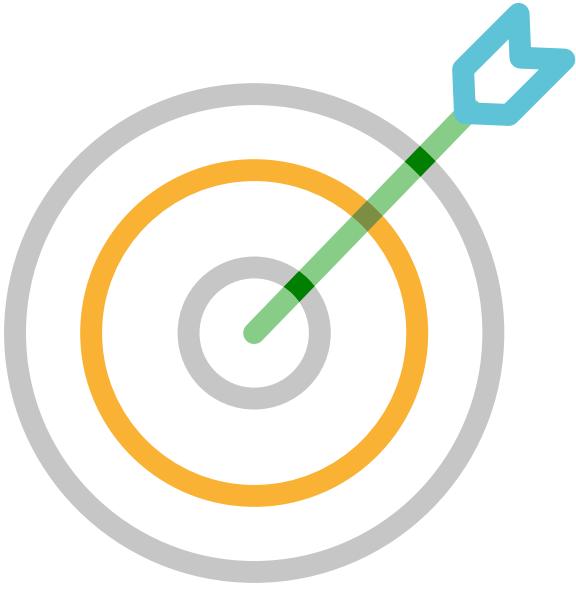
Spiral Model

Bu evrimsel süreçte, sistem geliştirme takımı merkezden başlayarak saat yönünde bölümler ile ilgili görevleri gerçekleştirir. Spiralde bir tur tamamlandığında ürün özellikleri tamamlanabilir; ikinci turda prototip geliştirilebilir; sonraki turlarda da adım adım sistemin daha gelişmiş sürümleri oluşturulabilir. Planlama bölmesinden her bir geçişte proje planında daha hassas düzenlemeler yapılır. Maliyet ve zamanlama müşterinin değerlendirmesine göre tekrar gözden geçirilerek ayarlanır.



Spiral Model

İlave olarak, proje yöneticisi sistemi tamamlamak için gereken planlanmış iterasyon sayısını ayarlar. Spiral model aynı zamanda sistemin yaşam döngüsü boyunca da uygulanabilir. Spiral model projenin her adımında teknik ve yönetim risklerini dikkate alarak risklerin büyük problemlere sebep olmadan giderilmesini sağlar.



ÇEVİK MODEL

Bölüm 6

Çevik Model

Çevik model (Agile Model) sistemin gereksinimlerini belirleme, tasarım, geliştirme ve kurulumunda artan yaklaşımı kullanır. Geliştirme sürecinde sistem gereksinimleri genelde değiştiği durumlarda Çevik model en uygun yaklaşılardan biridir. Çevik modelde çalışan yazılım müşterisiye hızlıca verilerek, yeni ve değişen gereksinimler ileri iterasyonlarda dâhil edilir. Burada amaçlanan, uzun zaman alan belgelemeyi azaltarak, tüm çabayı sistem geliştirmeye vermektedir. Çevik modelin felsefesi aşağıdaki Çevik Yazılım Geliştirme Manifestosunda verilmiştir.



Çevik Model

Bizler daha iyi yazılım geliştirme yollarını uygulayarak ve başkalarının da uygulamasına yardım ederek ortaya çıkartıyoruz.

Bu çalışmaların sonucunda:

*Süreçler ve araçlardan ziyade bireyler ve etkileşimlere,
Kapsamlı dokümantasyondan ziyade çalışan yazılıma,
Sözleşme pazarlıklarından ziyade müşteri ile işbirliğine,
Bir plana bağlı kalmaktan ziyade değişime karşılık vermeye
değer vermeye kanaat getirdik.*

Özetle sol taraftaki maddelerin değerini kabul etmekle birlikte, sağ taraftaki maddeleri daha değerli bulmaktayız.

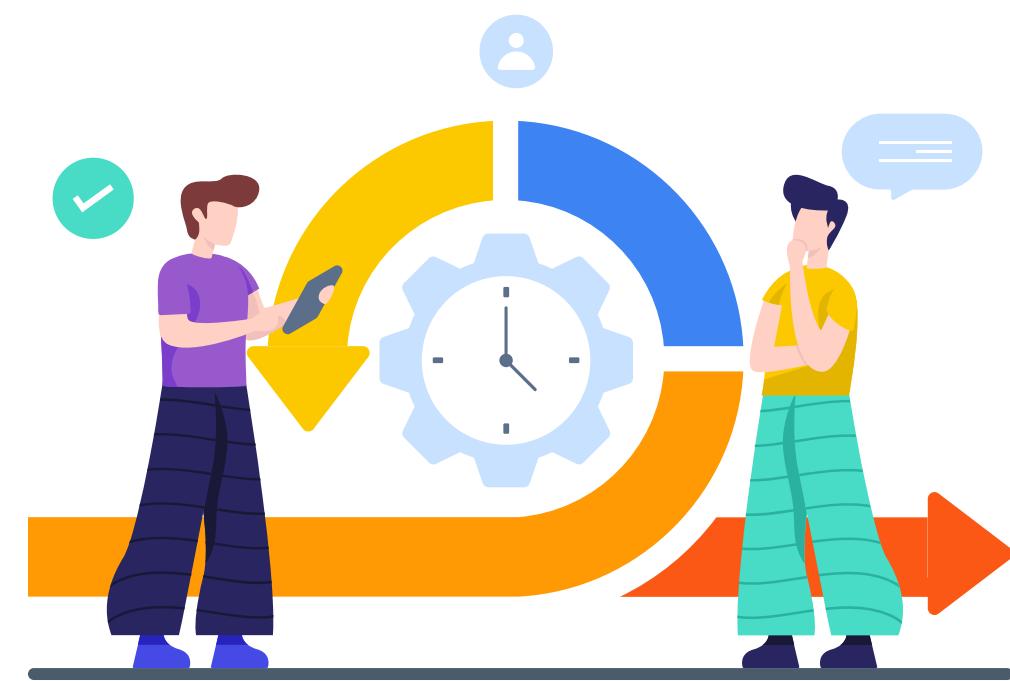
Kaynak: <http://agilemanifesto.org/iso/tr/> adresinden alınmıştır.



Çevik Model

Çevik modelin uygulanmasında dört teknik yaklaşım ön plandadır. Bunlar: kısa sürede sürüm teslimi (2 haftada bir), haftada 40 saat uygulama geliştirme (belgeleme yok), müşteri ile içi içe geliştirme ve ci kişi programlamadır.

Çevik model kullanan yaklaşımalar, Uç Programlama (eXtreme Programming), Adaptif Yazılım Geliştirme (Adaptive Software Development), Scrum ve Özellik Gündümlü Programlamadır (Feature Driven Development).



Çevik Model

Çevik yaklaşımların aşağıdaki tip sistemlerin geliştirilmesinde daha başarılı olduğu görülmüştür.

- i. Küçük ve orta büyülükte sistemleri geliştiren işletmelerin ürün geliştirmesinde
- ii. Müşterinin geliştirme süreci içinde olduğu ve sistemin gereksinimlerinde belirli standartların (kalite, güvenlik vb.) aranmadığı durumlarda, işletme içinde özelleş- tirilmiş sistemlerin geliştirilmesinde



Çevik Model

Özelliklerde ortaya çıkabilecek hataların önemli sorunlara yol açabileceğine kritik sistemlerde çevik yaklaşım kullanılmamalıdır. Çünkü bu durumlarda geliştirme başlamadan önceki fazlarda tüm gereksinimler tam ve ayrıntılı tanımlanmış olmalıdır. Günümüzde Çevik geliştirme yaklaşımlarından en yaygın kullanılan Scrum sonraki slaytta anlatılacaktır.

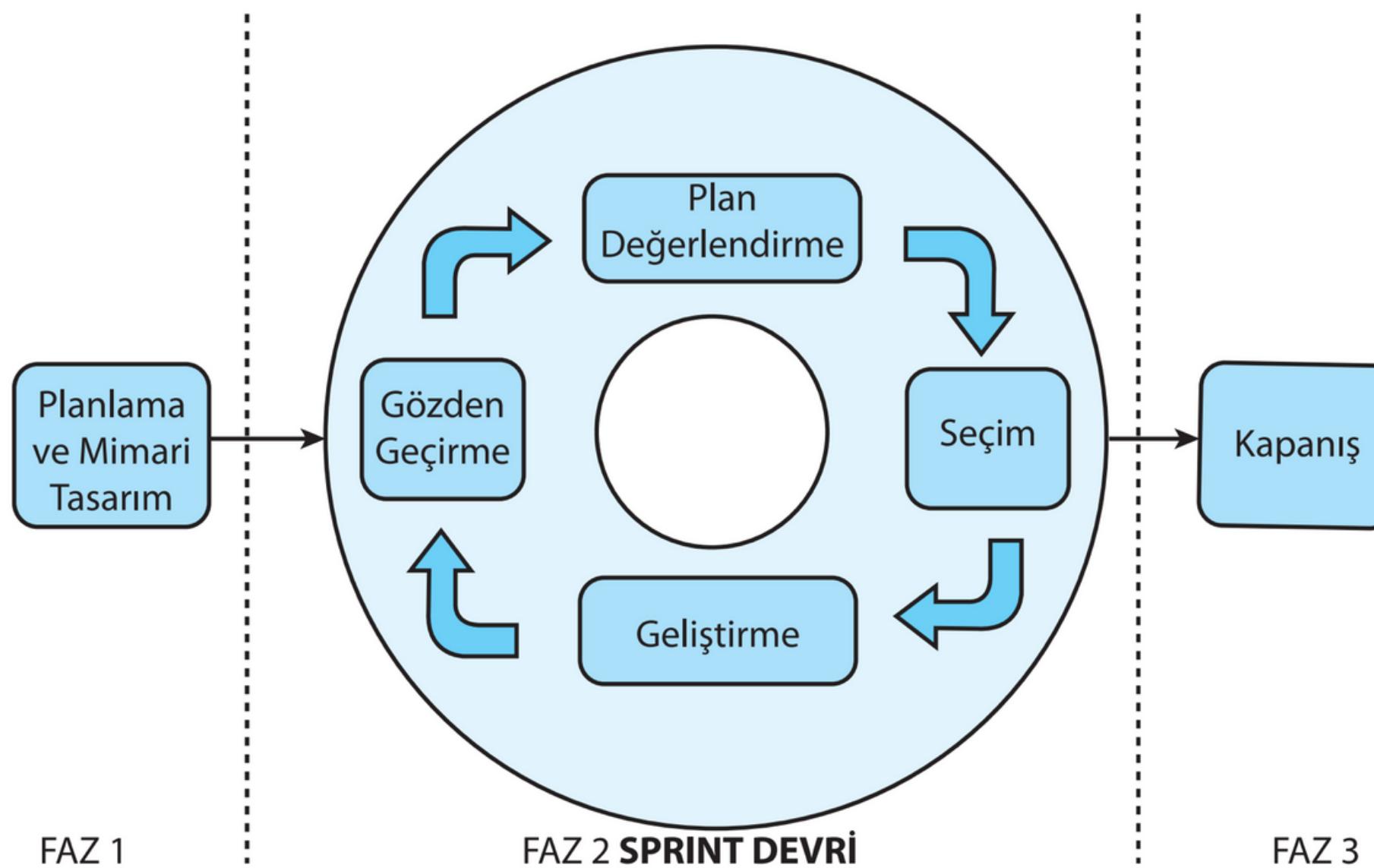


Scrum Model

Scrum Scrum yaklaşımı çevik yöntemlerin teknik yaklaşımları yerine iteratif geliştirme yöntemine odaklanmış bir çevik metottur. Şekilde Scrum yönetim süreci gösterilmektedir. Scrum' da 3 faz vardır. Birinci fazda proje genel hedefleri ve yazılım mimarisinin tasarımını oluşturmak üzere ana hat planlama yapılır. Bu aşamada oluşturulan projenin ana gereksinimleri önem sırasına göre bir liste (Product Backlog) olarak toplanır (Faz 1).

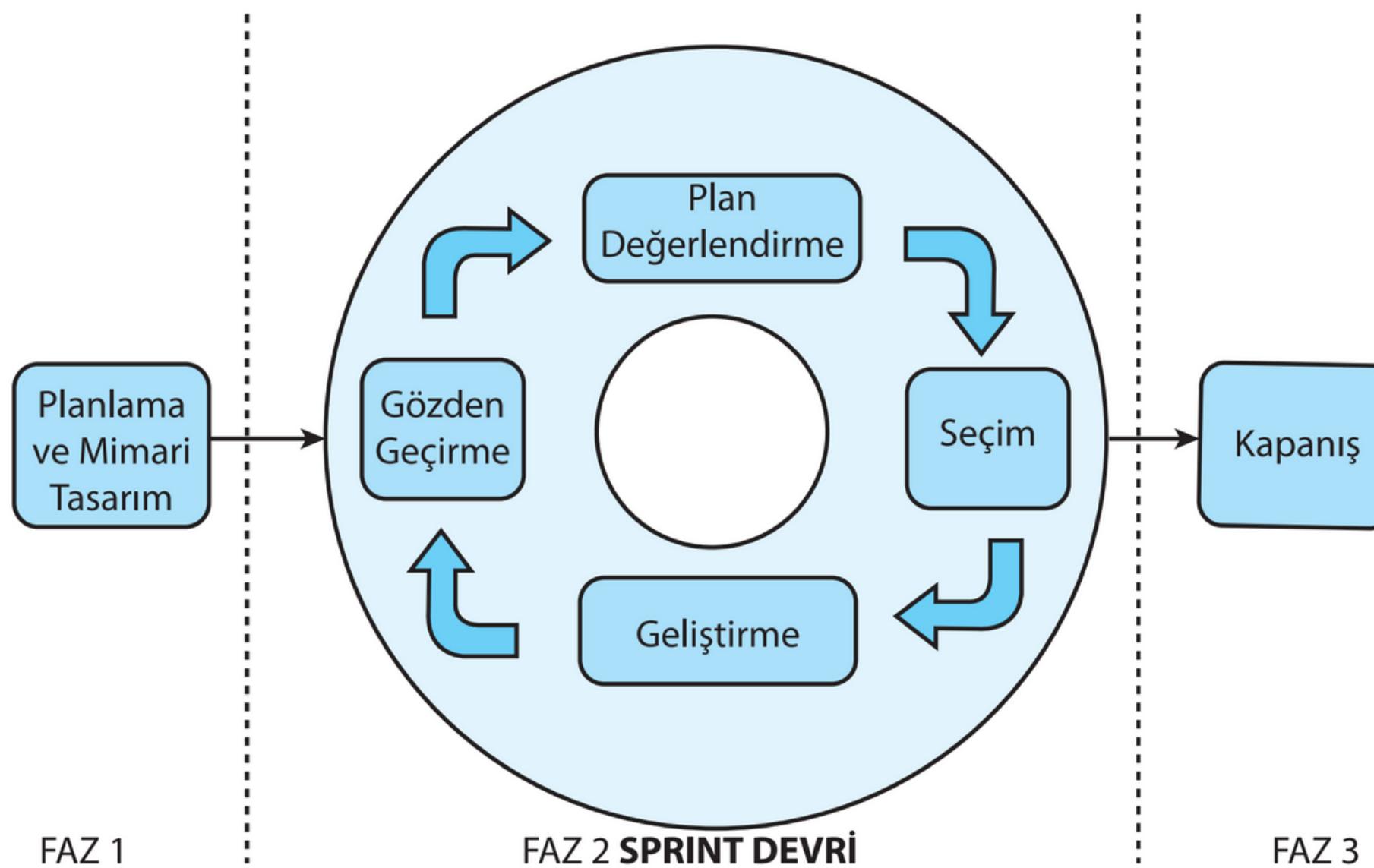


Scrum Model



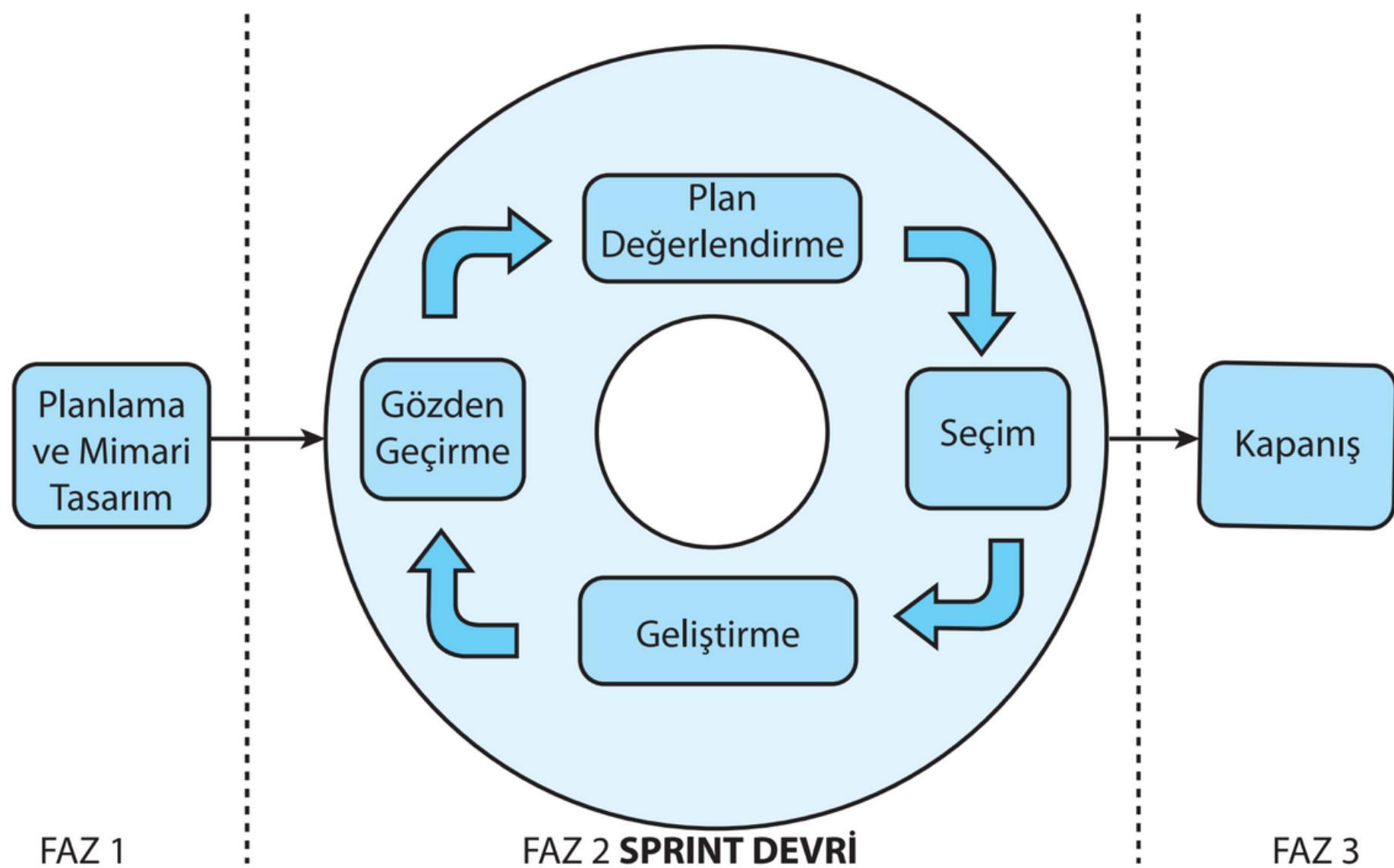
Bu aşamayı her turda sistemde bir artım sağlayan sprint devirleri (sprint cycle) takip eder (Faz 2). Son aşamada da ürün belgeleri (yardım belgeleri, kullanıcı el kitapları) ve projeden elde edilen deneyimlerin değerlendirildiği proje kapanış fazı vardır (Faz 3). Scrum yaklaşımının iteratif yapısını 2'nci fazdaki Scrum sprint devri oluşturur.

Scrum Model



Sprint devirleri sabit zaman aralıklarında yapılır (tipik olarak 2-4 hafta). Sprint devri plan değerlendirme ünitesiyle başlar. Ardından geliştirilecek özellikler seçilir ve yazılım geliştirilir. Sprint devrinde geliştirilecek özelliklerin seçimine tüm proje ekibi katılır.

Scrum Model



Yazılım geliştirme sırasında ise geliştirme ekibi müşteriden izole edilerek sadece yazılım ile ilgilenir. Sprint devrinin sonunda, tamamlanmış sürüm sistem paydaşlarına dağıtılarak gözden geçirilmesi sağlanır.

Kaynaklar

1. Öğr. Gör. Aslı Birol, Sistem Analizi ve Tasarımı, BIL3403
Ders Notları
2. Anadolu Üniversitesi, BIL206, Sistem Analizi ve Tasarımı
Ders Notları

