

Yazılım Geliştirme Ortam ve Araçları

2. Hafta

YAZILIM SÜREÇ ADIMLARI ve MODELLERİ

Kaynaklar:

1-) Yrd. Doc. Dr. Yunus Emre SELÇUK – Yıldız Teknik Üniversitesi – Yazılım Mühendisliği Temelleri Ders Notları, Şubat 2017

2-) <http://cemtopkaya.blogspot.com.tr/2008/06/yazlm-mhendislii.html>

3-) <https://www.slideshare.net/okkesemin/agile-proje-ynetimi-10625113>

4-) <http://kod5.org/scrum-metodolojisine-giris/>



Yazılım Süreç Adımları

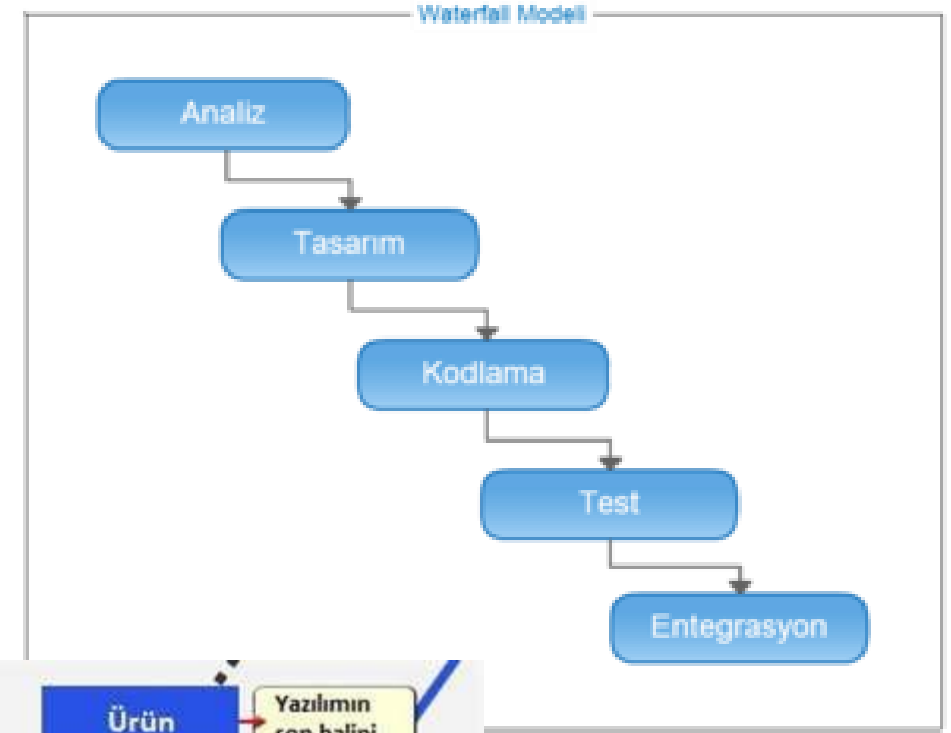
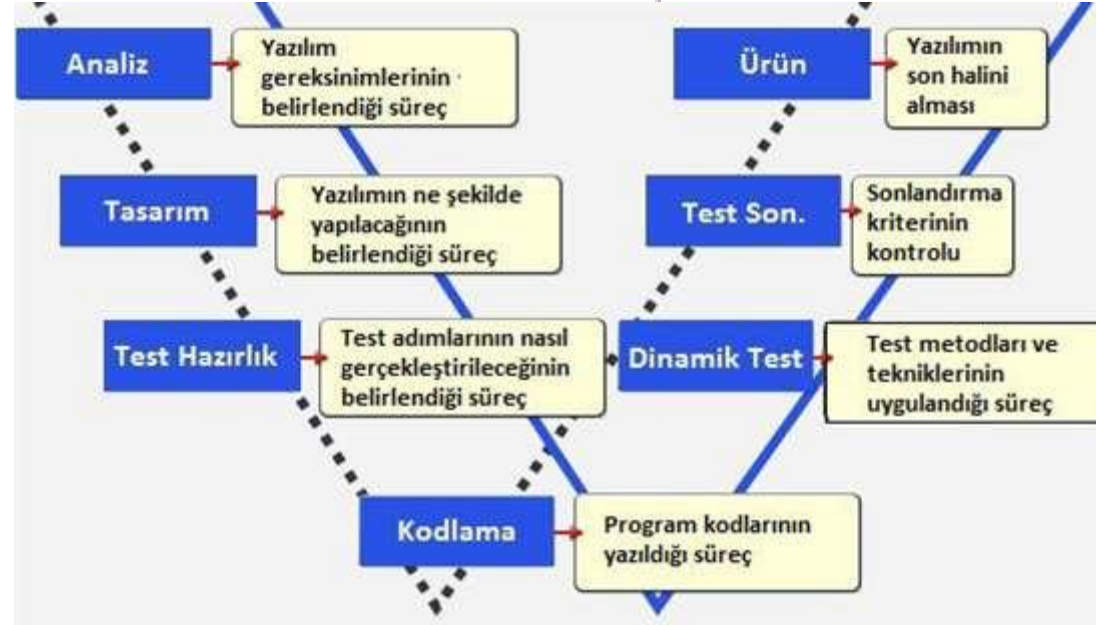
- Çözümleme (Analiz)
- Tasarım (Dizayn)
- Gerçekleme (Geliştirme)
- Sınama (Test)
- Bakım

YAZILIM SÜREÇLERİNİN GENEL ADIMLARI

- Çözümleme (Analysis)
- Tasarım (Design)
- Gerçekleme (Implementation)
- Sınama (Testing)
- Bakım (Maintenance)

Yazılım Süreci:

Yazılım Geliştirmek
İçin Kullanılan
Aktivite Kümesi



ÇÖZÜMLEME

- Çözümleme: Bir şeyi anlayabilmek için parçalarına ayırmak.
- Gerçeklenecek sistemi anlamaya yönelik çalışmalardan ve **üst düzey** planlama eylemlerinden oluşur.
 - Uygulama alanı
 - Kullanıcı gereksinimleri
 - Program parçaları arasındaki üst düzey ilişkiler ve etkileşimler (NYP'deki parçalar: sınıflar ve nesneler)
- “Bir sorunu anlamadan çözemezsiniz.”



TASARIM

- Tasarım: Bir araştırma ve/veya geliştirme sürecinin çeşitli dönemlerinde izlenecek yol ve işlemleri tasarlayan çerçeve.
- Çözümleme ile anlaşılan sorun tasarım aşamasında kağıt üzerinde (!) çözülür.
- Yazılım ↔ Tasarıma yönelik şemalar (NYP'de bazı tür UML şemaları), elektronik ↔ devre şemaları, mimari ↔ kat planları



GERÇEKLEME

- Eldeki tasarım, bir programlama dili ile kodlanır.



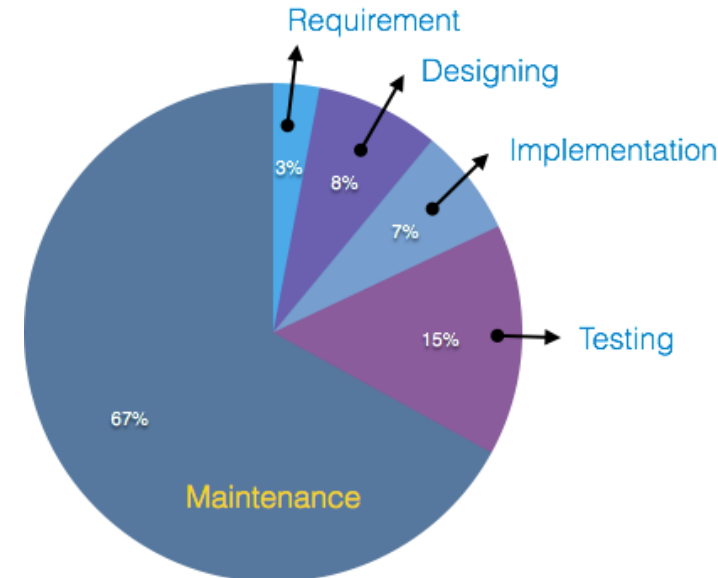
SINAMA

- Sinama neden önemlidir?
 - Yazılım sürecinde ilerledikçe, ortaya çıkabilecek hataların giderilme maliyeti üstel olarak artar.
 - Aksi gibi, hataların büyük çoğunluğunun kökenleri isteklerin belirlenmesi ve tasarım aşamalarındaki sorunlara dayanır.
 - Bu yüzden: Erkenden, sık sık ve kolay sinama yapın.



BAKIM

- Yazılımın faaliyete geçirilmesinden sonra sistemde yapılan değişikliklerdir.
 - Yazılım hatalarının düzeltilmesi:
 - Kodlama hataları
 - Tasarım hataları (!)
 - Gereksinim ve analiz hataları (!!)
 - Sistemin işlevlerini değiştirme veya işlevlere eklemeler/çıkarmalar,
 - Yazılımın farklı bir ortama taşınması (programlama dili, işletim sistemi, donanım, iklim, vb.) (porting)
- Yeniden mühendislik (Refactoring / Software re-engineering)
 - Teknik bakış açısı: Yazılımın işlevini değiştirmeden iç yapısını değiştirmek.
 - Olası eylemler:
 - Yazılımın belgelendirilmesi
 - Tasarımın iyileştirilmesi/değiştirilmesi
 - Yazılımın farklı bir ortama taşınması



Yazılım Süreç Modelleri

- Şelale (Waterfall)
- Sarmal (Spiral)
- Çevik (Agile)

YAZILIM GELİŞTİRME SÜREÇ (MODEL)LERİ

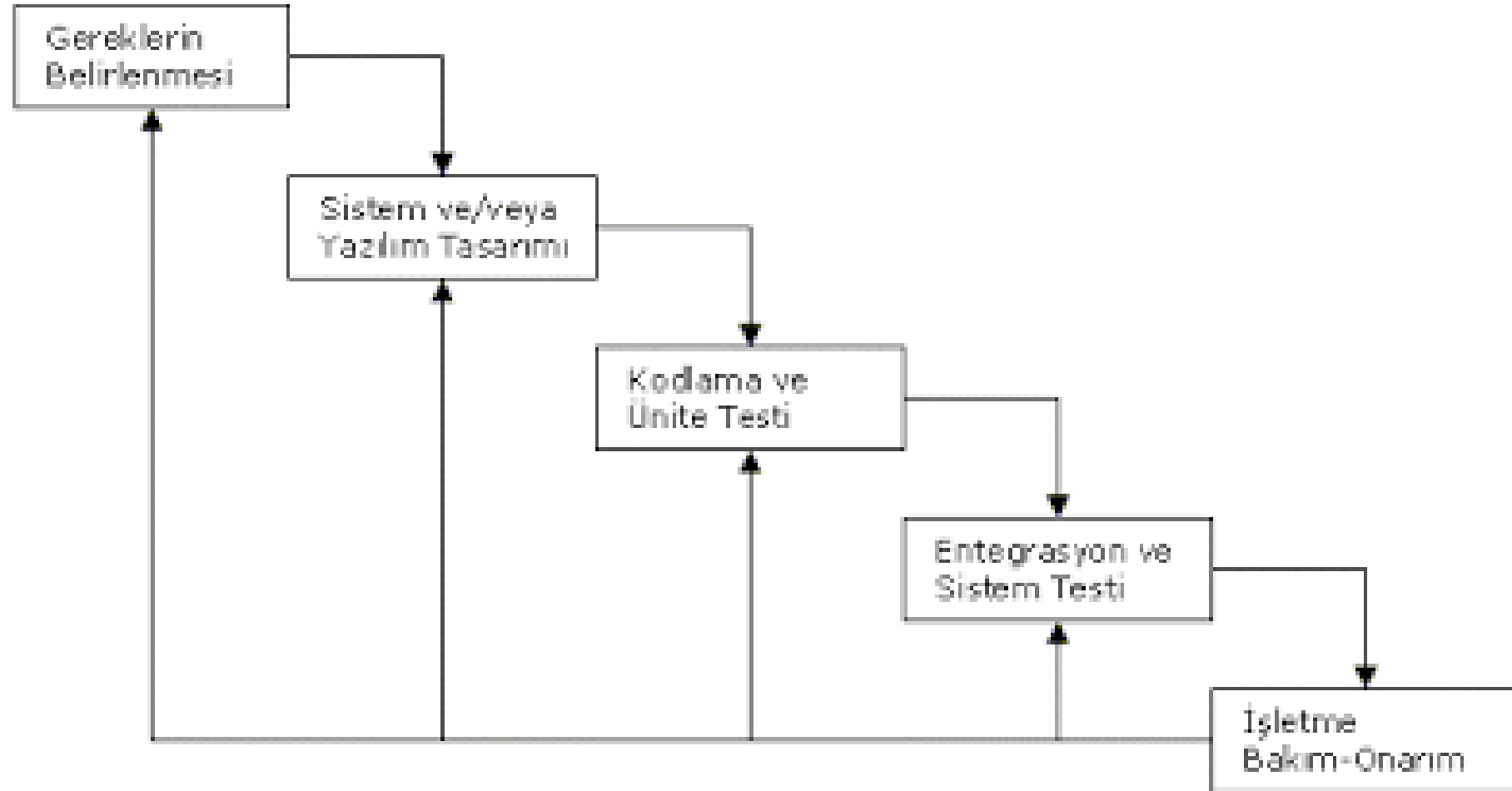
- Yazılım geliştirme bir süreç olarak ele alınmalıdır.
 - Süreç: Önceden belirlenmiş adımlardan oluşan iş akışı.
- Süreç modelleri, yazılım geliştirme sürecinin yapısını ve adımlarını belirler.
 - Önceden ve iyi planlanmış bir süreç, zamanında ve 'kaliteli' bir 'ürün' elde edilmesini sağlar.
- Çeşitli modellerin kendine özgü avantaj ve dezavantajları vardır.
 - Gerçeklenecek projeye uygun modelin seçilmesi gerekir.

ŞELELE MODELİ

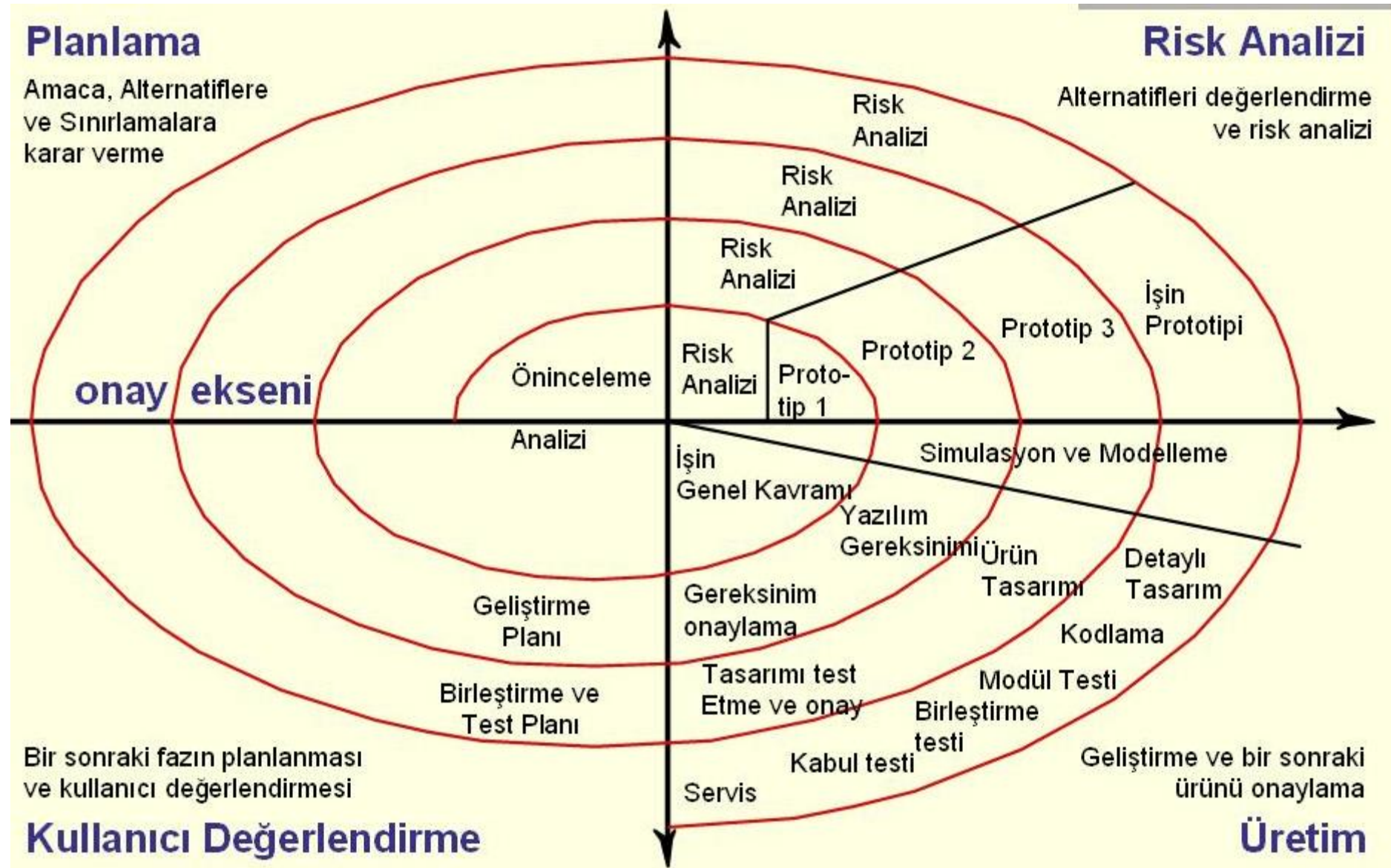
- Ardışıl Model / Şelale Modeli (Sequential / Waterfall)
 - Adımlar: Çözümleme → Tasarım → Kodlama → Sınama → Bakım.
 - Bir adımın tamamlanmasından sonra diğerine geçilir.
 - Eksiklikler veya hatalar fark edilirse bir önceki adıma geçilir.
- Artılar:
 - En eski model, yaygın kullanımda.
 - İyi tanımlanmış adımlar.
- Eksiler:
 - Son ürünün eldesi uzun süreceğinden müşteri sabırlı olmalıdır.
 - Adımları geride bıraktıkça, ilerleyen aşamalarda karşılaşılan hataların düzeltilmesi üstel olarak zorlaşmaktadır.
 - Bir çok 'müşteri' de gereksinimleri eksiksiz ve kesin belirtmekte zorlanmaktadır.
- Sonuç: Hiç model kullanmamaktan iyidir!
 - Önceden bir çok kez başarıya ulaştırılmış projelere benzer yeni projelerin yürütülmesi için kullanılabilir (rutin projeler).

ŞELELE MODELİ

Şekil 1 - Şelale Model İşleyişi



SARMAL (Spiral) MODEL



SARMAL (Spiral) MODEL

- Süreç Adımları:
 - Planlama: Üretilcek ara ürün için planlama, amaç belirleme, bir önceki adımda üretilen ara ürün ile bütünleştirme
 - Risk Analizi: Risk seçeneklerinin araştırılması ve risklerin belirlenmesi
 - Üretim: Ara ürünün üretilmesi
 - Kullanıcı Değerlendirmesi: Ara ürün ile ilgili olarak kullanıcı tarafından yapılan sına ve değerlendirmeler
- Avantajları:
 - Kullanıcı Katkısı
 - Üretim süreci boyunca ara ürün üretme ve üretilen ara ürünün kullanıcı tarafından sınalanması temeline dayanır.
 - Yazılımı kullanacak personelin sürece erken katılması ileride oluşabilecek istenmeyen durumları engeller.
 - Yönetici Bakışı
 - Gerek proje sahibi, gerekse yüklenici tarafındaki yöneticiler, çalışan yazılımlarla proje boyunca karşılaştıkları için daha kolay izleme ve hak ediş planlaması yapılır.
 - Yazılım Geliştirici (Mühendis) Bakışı
 - Yazılımın kodlanması ve sınalanması daha erken başlar.
 - Risk Analizi Olgusu ön plana çıkmıştır.
 - Her döngü bir fazı ifade eder.
 - Doğrudan tanımlama, tasarım,... vs gibi bir faz yoktur.
 - Yinelemeli artımsal bir yaklaşım vardır.
 - Prototip yaklaşımı vardır.

ÇEVİK (Agile) SÜREÇLER

- Değişen gereksinimler, teknik riskler gibi önceden belirlenemeyen durumlara ve yazılım ürününü etkileyebilecek her tür değişikliğe karşı esneklik sağlayan süreçlerdir.
- Bireyler ve etkileşimler
- Çalışan yazılım
- Müşterinin sürece katılımı
- Değişikliklere uyum sağlamak
- Süreçler ve gereçler
- Ayrıntılı belgeler
- Sözleşme pazarlığı
- Bir planı izlemek
- Çevik süreçler, sağ taraftaki maddelerin yararını kabul etmekle birlikte, sol taraftaki maddelere daha çok önem vermektedir.
- Bir ilerleme olmaksızın yalnızca sürekli uyum sağlamak başarı değildir.
 - Yazılımın artımsal gelişimi
 - Müşteriye erken ve sık ürün teslimi
 - Başarımın birincil ölçütü doğru çalışan yazılımdır.

ÇEVİK (Agile) SÜREÇLER

- Çevik süreci yürütecek ekibin özellikleri:
 - Yüz yüze görüşme, en etkili bilgi aktarım yoludur.
 - Takım üyeleri çevik yaklaşım hakkında eğitilmelidir.
 - Ekip üyelerinin ortak amacı, çalışan yazılım üreterek müşteriye zamanında teslim etmek olmalıdır.
 - Ekip üyeleri birbirleriyle ve müşteriyle işbirliği içinde olmalıdır.
 - Ekip üyeleri karşılıklı saygı ve güven içerisinde olmalıdır.
 - Ekipler hem teknik, hem de tüm proje hakkında kararlar verebilmelidir.
 - Boşuna harcanan çaba yoktur: Çözülen bir sorun gereksizleşse bile, çözüm sürecinde edilen deneyim ekibe ileri aşamalarda yararlı olabilir.
- Kendi kendini düzenleme:
 - Ekibin kendisini yapılacak işe göre uyarlaması,
 - Ekibin kullanacağı süreci yerel ortama uyarlaması,
 - Üstünde çalışılan artımsal yazılım parçasını teslim etmek için gerekli çalışma zamanlamasını ekibin kendisinin belirlemesi.

ÇEVİK (Agile) SÜREÇLER

- Çevik süreçlerin dezavantajları:
 - Uygun olmayan ekiple çevik çalışılamamaktadır.
 - Kalabalık ekip veya büyük ölçekli projeler için uygun görülmemektedir.
 - Bir dış denetleyicinin dahil olduğu ve ayrıntılı kuralların gerektiği denetlemelerin zorunlu olduğu projelerde yetersiz kalmaktadır.
 - Çevik çalışmak disiplinsizlik olarak yorumlanmamalıdır.

FARKLI AGILE FRAMEWORKLER

Bu frameworkler agile prensiblerini benimsemiş farklı çalışma şekilleridir.

- Scrum
- Extreme Programing
- Lean Software Development
- Feature Driven Development
- Dynamic Systems Development Method

Scrum

Çevik Framework Aracı

SCRUM

- Scrum ilk olarak kompleks yazılım projelerinin yönetilmesinde kullanılan, proje ekibinin çevik davranarak kompleks süreçleri sadeleştirmesini ve üretkenliğini artırmasını hedefleyen bir proje yönetim frameworküdür. Bunu yaparken bütünü parçalayan tekrara dayalı bir yöntem kullanır.
 - Temelde ürünün daha hızlı, kaliteli ve ucuz olarak üretilmesi hedeflenmektedir.
 - Çoğunlukla yazılım projelerinde kullanılmasına rağmen, farklı disiplinlerde de Scrum uygulanmaktadır.
- Projelerdeki karmaşıklığı en aza indirmek, şeffaflığı artırmak ve riskleri minimize etmek için Scrum 3 temel prensip üzerine kurulmuştur.
 - **Şeffaflık(Transparency)** : Projenin ilerleyişi, gelişmeler, sorunlar, kısaca herşeyin herkes tarafından görülebilir olmasıdır.
 - **Denetleme(Inspection)** : Projenin ilerleyişinin her açıdan düzenli olarak kontrol edilmesidir.
 - **Uyarlama(Adaptation)** : Projedeki süreçlerin, pratiklerin yada yapılan işin kabul edilebilir sınırlar dışarısına çıktığında değiştirilebilmesidir.

SCRUM

- Roller:

- **Product Owner** : Projede ortaya çıkarılacak olan ürünün sahibidir. Müşteriyi/bağlı olduğu üst yönetimi temsil eder.
- **Scrum Master** : Projede herşeyin yolunda gitmesinden sorumlu olan kişidir. Takım ve product owner arasında köprü vazifesi görür.
- **Development Team** : Projede ortaya çıkarılacak olan ürünü yapan/geliştiren ekiptir. Çeşitli alanlarda uzman kişiler bulunur.

SCRUM

- Toplantılar:
 - **Sprint Planlama Toplantısı** : Gerçekleştirilecek olan sprintin içeriğinin planlandığı toplantıdır.
 - **Daily Stand Up Toplantısı** : Günlük olarak takımın bir araya geldiği, yaptıkları işleri birbirlerine anlatarak senkronize olduğu toplantıdır.
 - **Sprint Review Toplantısı** : Sprint sonunda çıkartılan ürünün değerlendirildiği toplantıdır.
 - **Sprint Retrospective Toplantısı** : Sprint boyunca yapılan işlerin kalitesinin, doğruların ve yanlışların değerlendirildiği toplantıdır

SCRUM

- Terminoloji:
 - **Ürün İş Listesi (Product Backlog)** : Ürünün tüm özelliklerinin listesidir.
 - **İterasyon İş Listesi (Sprint Backlog)** : Ürünün özelliklerinden hangilerinin sprint içerisinde yapılacağının listesidir.
 - **Ürün İş Listesi Elemanı (Product Backlog Item)** : Ürünün özelliklerinin bulunduğu listedeki her bir ürüne verilen isimdir.
 - **Eriyen Grafik (Burndown Chart)** : Yapılacak olan işlerin eriyerek azaldığının gözlemlendiği grafiktir. Sprint burndown chart ve release burndown chart olarak 2 ye ayrılır.
 - **Tamamlandı Kriteri (Definition Of Done – DoD)** : Her bir product backlog item'ın tamamlanma kriteridir.

SCRUM PROJE YÖNETİMİ SÜRECİ



Inputlar
Yönetimden Gelir



Product Owner



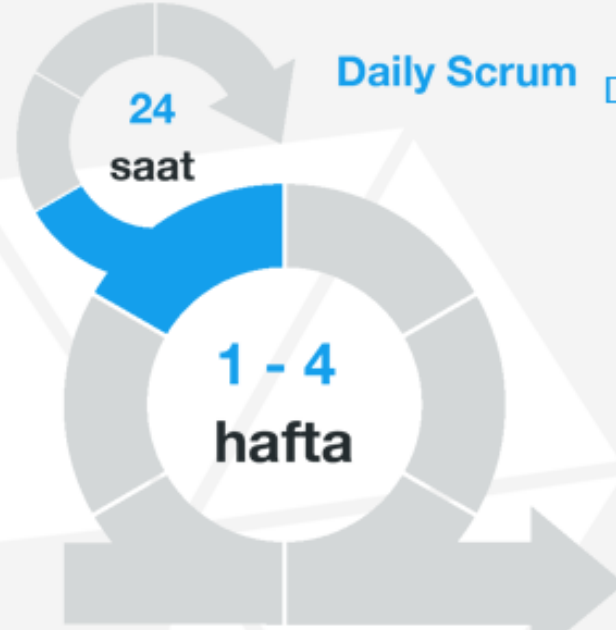
Product
Backlog



Takım ile yapılır
Sprint
Planlama
Toplantısı



Sprint
Backlog



24
saat

Daily Scrum



Daily Stand Up
Toplantısı

1 - 4
hafta

Sprint

Sprint bitiş süresi ve
içeriği değiştirilemez



Sprint Review
Toplantısı



Tamamlanan İş



Sprint Retrospective
Toplantısı

Temel Prensipler

Eğer bunları yapıyorsanız, listenin geri kalanını göz ardı edebilirsiniz. Süreciniz gayet iyidir.

- ☐ Her 4 veya daha az hafta da çalışan ve test edilmiş yazılım teslimi yapılmaktadır.
- ☐ İş biriminin en çok ihtiyacı olan fonksiyonlar teslim edilmektedir.
- ☐ Süreç sürekli olarak geliştirilmekte ve iyileştirilmektedir.

- ☐ Takımda açıkça tanımlanmış ürün sahibi (ÜS) rolü vardır.
 - ☐ ÜS önceliklendirme yapmaya yetkilidir.
 - ☐ ÜS önceliklendirme yapmak için gerekli bilgiye sahiptir.
 - ☐ ÜS takımla direk olarak iletişim halindedir.
 - ☐ ÜS paydaşlarla direk olarak iletişim halindedir.
 - ☐ ÜS'nin bir takım olması durumunda bir konuda aynı fikri beyan etmektedir.
- ☐ Takım'ın bir sprint backlog'u vardır.
 - ☐ Kolay ulaşılabilir
 - ☐ Günlük güncellenir
 - ☐ Yalnızca takım'a aittir
- ☐ Günlük Scrum Toplantıları gerçekleştirilmektedir.
 - ☐ Bütün takım katılmaktadır.
 - ☐ Sorunlar ve engeller konuşulmaktadır.
- ☐ Her sprint sonunda demo gerçekleştirilir.
 - ☐ Çalışan ve test edilmiş yazılım gösterilir.
 - ☐ ÜS ve paydaşlardan geri bildirimler alınır.
- ☐ Tamamlandı Kriteri (TK) tanımlanmıştır.
 - ☐ TK, her iterasyon içerisinde başarılabılır durumdadır.
 - ☐ Takım, TK'ya saygı duyar.

Scrum Temelleri

Bu maddeler Scrum'ın merkezindedir. Bu maddeler olmadan büyük olasılıkla uyguladığınızı sürece Scrum diyemezsiniz.

- ☐ Her sprint sonunda gözden geçirme toplantısı yapılmaktadır
 - ☐ Somut iyileştirme önerileri oluşturulur
 - ☐ Bazı öneriler fiilen uygulanmaktadır
 - ☐ Bütün Takım + ÜS katılmaktadır
- ☐ ÜS'nin bir ürün backlogu (ÜBL) vardır
 - ☐ ÜBL'de üstte bulunan maddeler iş değerine göre önceliklendirilmiştir
 - ☐ ÜBL'de üstte bulunan maddeler için tahminleme yapılmıştır
 - ☐ Tahminler takım tarafından yapılmaktadır
 - ☐ ÜBL'de üstte bulunan maddeler sprint kapsamına alınabilecek kadar küçük boyuttadır.
 - ☐ ÜS bütün backlog maddelerinin amacını anlamaktadır.
- ☐ Sprint Planlama Toplantıları gerçekleştirilmektedir.
 - ☐ ÜS katılmaktadır
 - ☐ ÜS güncel ÜBL getirmektedir
 - ☐ Bütün takım katılmaktadır
 - ☐ Toplantı sonunda sprint planı hazırlanmaktadır.
 - ☐ Bütün takım planın başarılabılır olduğuna inanır.
 - ☐ ÜS önceliklikler konusunda memnundur
- ☐ İterasyonların maksimum zaman sınırı vardır
 - ☐ İterasyon uzunluğu 4 veya daha az haftadır.
 - ☐ Her zaman zamanında bitirilir
 - ☐ Takım, dışarıda kalan kişiler tarafından kontrol edilemez ve değiştirilemez
 - ☐ Takım genellikle taahhüt ettiği şeyi teslim eder
- ☐ Takım üyeleri aynı yerde oturmaktadır
 - ☐ Her takımda en fazla 9 kişi vardır

Scrum Kontrol Listesi



Resmi Olmayan

Önerilen fakat her zaman gerekli olmayanlar

Bu maddelerin bir çoğu genellikle gereklidir, fakat her zaman hepsi gerekli olmayabilir. Deneyin!

- ☐ Takım backlog maddelerini TK'ya uygun hale getirmek için gerekli bütün yeteneklere sahiptir.
- ☐ Takım üyeleri belirli rollere kilitlenmemiştir.
- ☐ Başarısız olacağı belli olan iterasyonlar erken bitirilir
- ☐ ÜS'nin ÜBL ile uyuşan bir ürün vizyonu bulunur.
- ☐ ÜBL ve ürün vizyonuna kolay ulaşılabilir.
- ☐ Takımdaki herkes tahminlemeye katkıda bulunur.
- ☐ Takım tahminleme yaparken ÜS ulaşılabilir olmalıdır
- ☐ Tahminleme zaman yerine göreceli boyut (hikaye puanı) kullanılarak yapılır.
- ☐ Bütün takım en öncelikli 1-3 engeli bilmelidir
 - ☐ SM'in en öncelikli engeli nasıl çözeceğine dair bir stratejisi vardır
 - ☐ SM, engelleri kaldırmaya odaklanmalıdır
 - ☐ Takımın çözemediği durumda yönetime iletilir
- ☐ Takımda bir Scrum Master (SM) vardır.
 - ☐ SM takımla aynı yerde oturmaktadır
- ☐ ÜBL maddeleri, sprint içerisinde görevlere bölünmüştür
 - ☐ Sprint görevleri tahminlenmiştir
 - ☐ Devam eden görevler için tahminler günlük güncellenir
- ☐ Hız ölçülür
 - ☐ Sprint'teki bütün maddeler için tahmin yapılmıştır
 - ☐ ÜS hızı yaygınlaştırma planlaması için kullanır
 - ☐ Hız sadece TK'ne uyan maddeleri içermektedir
- ☐ Takım'ın bir sprint burndown grafiği vardır.
 - ☐ Kolay ulaşılabilir
 - ☐ Günlük güncellenir
- ☐ Günlük Scrum Toplantıları her gün aynı saate ve aynı yerdedir.
 - ☐ ÜS en az bir kaç kere katılmaktadır
 - ☐ En fazla 15 dakika gerçekleştirilir
 - ☐ Her takım üyesi diğerlerinin ne yaptığından haberdardır

Ölçekleme

Bunlar herhangi bir Scrum ölçekleme çabası için oldukça temel olan maddelerdir.

- ☐ Bir çok ÜS olduğu durumda Baş Ürün Sahibi bulunur.
- ☐ Bağımlı takımlar Scrum'ın Scrum'ı toplantısı yaparlar.
- ☐ Birbirine bağlı takımlar her sprintte entegre olurlar

Pozitif Göstergeler

İyi bir Scrum uygulamasının öncü göstergeleri

- ☐ Eğlencelidir. Enerji seviyesi yüksektir.
- ☐ Fazla mesai çalışması nadirdir ve gönüllü olarak yapılır.
- ☐ Süreçle ilgili tartışmalar, eleştiriler ve deneyler yapılmaktadır.