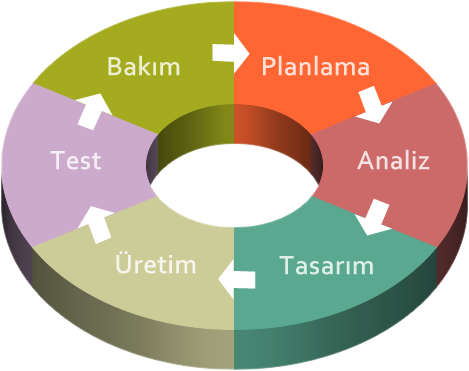
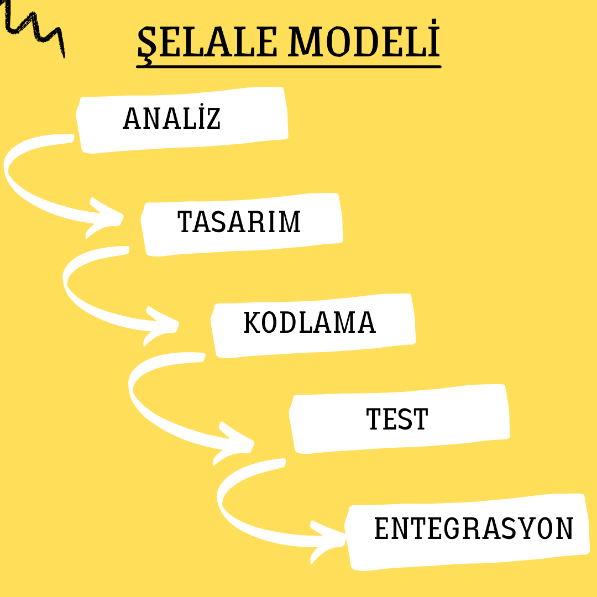
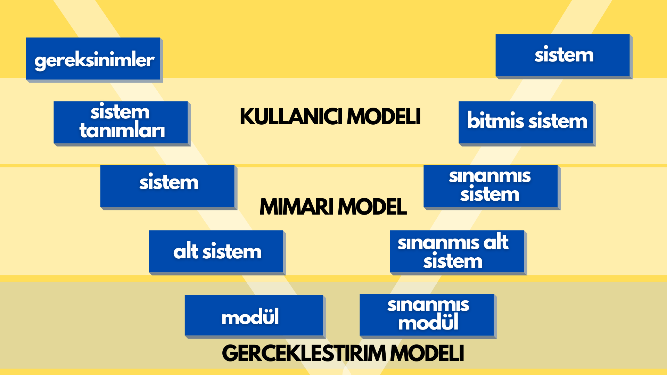
**Yazılım Yaşam Modelleri**

**Yazılım Yaşam Döngüsü Nedir?**

Bir yazılım geliştirmek sadece kodlamadan ibaret değildir. Yazılım bir üründür ve her ürün gibi yazılımın da bir yaşam süreci vardır. Yazılım yaşam döngüsü bir yazılımın planlama aşamasından kullanıcıya teslimatına kadar olan tüm aşamaları kapsayan döngüdür. Yazılım yaşam döngüleri Planlama, Analiz, Tasarım, Gerçekleştirim, Bakım aşamalarından oluşur.  
  
Planlama: Yazılım projesi planlanır, kullanıcıların istekleri göz önüne alınır. Aynı zamanda görev dağılımı da bu aşamada yapılır.  
  
Analiz: Projenin maliyeti, projeye ayrılması gereken zaman ve sistem gereksinimleri gibi detaylar analiz aşamasında düşünülür.  
  
Tasarım: Projenin nasıl sürdürüleceği tasarlanır.  
  
Gerçekleştirim: Kodlama ve test aşamaları gerçekleştirilir.  
  
Bakım: Proje müşteriye sunulduktan sonra oluşabilecek hataların düzeltilmesi için gerekli güncellemelerin ve bakımların yapılmasından oluşan aşamadır. Ayrıca yeni taleplere göre yazılıma eklemeler de yapılabilir.

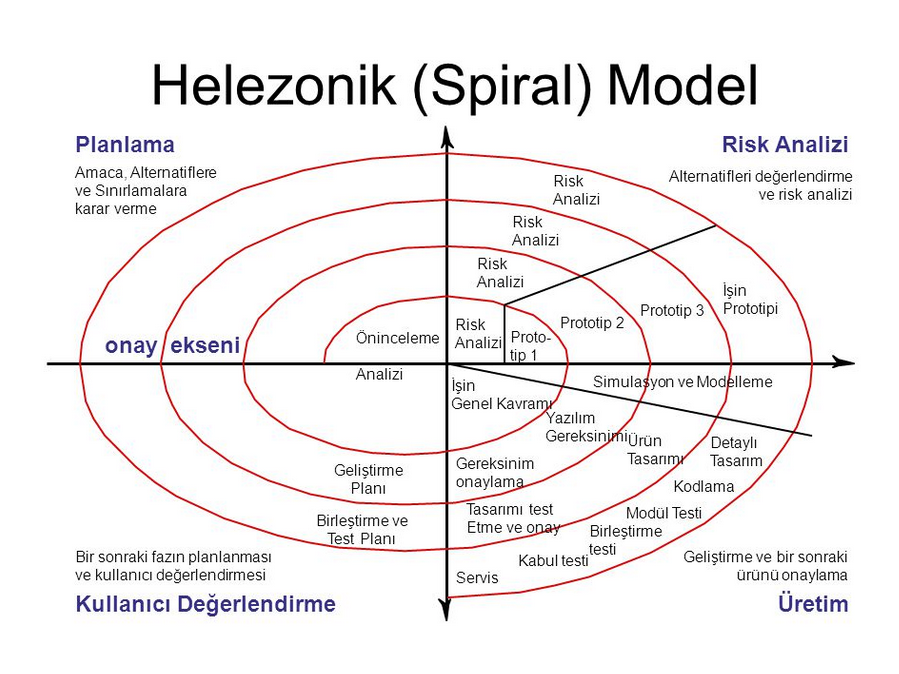
**Yazılım Yaşam Döngüsü Modelleri**

* Gelişigüzel Model: Bir yöntem veya model yoktur. Bakımı çok zordur. Günümüzde kullanılması önerilmez.
* Barok Modeli: Genel olarak gerçekleştirim aşamasına ağırlık verilmektedir ve günümüzde kullanılması önerilmez. Ayrıca döngü bulunmamaktadır. Belgelemeyi ayrı bir işlev olarak göz önüne almaktadır.
* Şelale (Waterfall) Modeli: Bu model diğer güncel yaşam döngülerinin temelidir.

1. Bir sonraki aşama önceki aşama tamamlanmadan başlamaz.
2. Gerektiği durumlarda bazı aşamalarda tekrarlamalar olabilir.
3. Tekrarlamalar genellikle art arda konumlanmış 2 aşama arasında olur. Daha uzak adımlar arasında tekrarlama bu modelde pek görülmemektedir.
4. Çok iyi tanımlanmış ve kısa sürede bitecek projeler için iyi bir seçim olabilir.
5. Bir faz tamamlanmadan diğerine geçilmesi riski artırır.
6. Barok modelle kıyaslayacak olduğumuzda belgeleme şelale modelinin içinde bulunur.
7. Kullanıcı katılımı yalnızca başlangıç aşamasında mümkündür. Sonraki aşamalarda kullanıcı ile iletişim kurulmaz.

* V Süreç Modeli:

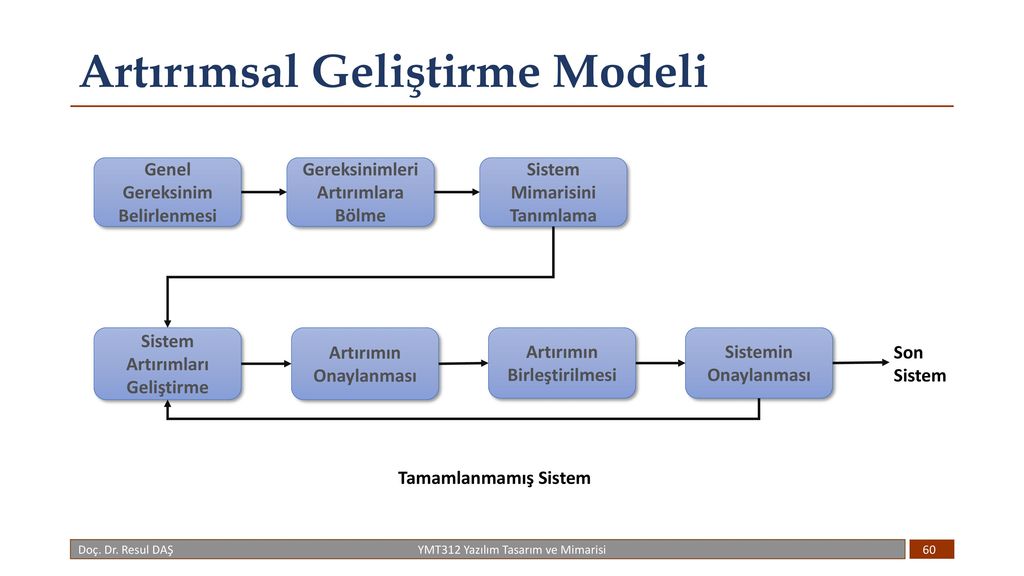
1. Sol taraf üretim sağ taraf ise sınama sürecidir.
2. Belirsizliklerin az ve tanımların belirgin olduğu iş modellerinde kullanılır.
3. Kullanıcının projeye katkısını artırmaktadır.
4. Şelale modelinin gelişmiş hali olarak da düşünülebilir.
5. Belirsizliklerin az olduğu ve iş tanımlarının yeterince açık olduğu projelerde kullanılır.
6. Bu model asıl olarak 3 aşamadan oluşmaktadır.
   * Kullanıcı Modeli, Mimari Model, Gerçekleştirim Modeli
   * Kullanıcı Modeli: Kullanıcının istekleri ve proje için tanımları alınır. Ardından tamamlanmış hali teslim edilir.
   * Mimari Model: Projenin tasarımı yapılır ve proje sınanır.
   * Gerçekleştirim Modeli: Kodlama evresinden ve bunların sınanmasında oluşur.

* Helezonik (Spiral) Model:

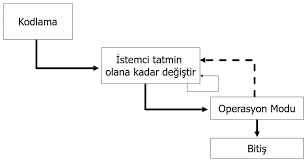
Bu model temel olarak 4 aşamadan oluşmaktadır. Planlama, Risk Analizi, Üretim ve son olarak da Kullanıcı Değerlendirmesi.

1. Planlama: Üretilecek ara ürün için işin planlanması; amacın, kısıtlamaların ve alternatiflerin belirlenmesi ve aynı zamanda önceki aşamada üretilen ara ürünle bütünleştirme yapılması gibi görevler yapılır.
2. Risk Analizi: Alternatiflerin aranması, risklerin araştırılması, belirlenmesi ve çözülmesinden oluşur.
3. Üretim: Planlanmış olan ara ürünün üretildiği aşamadır.
4. Kullanıcı Değerlendirmesi: Kullanıcıların ara ürünü test etmesi ve değerlendirmesi.

Spiral model risk analizi ve prototip üretme üzerine kurulmuş bir yazılım yaşam döngüsü modelidir. Her döngü öncesi içinde bulunulan fazın risk analizi yapılır ve o faz için planlanmış olan prototip üretilir. Risk analizi bu modelde ön planda olduğundan sorunlar daha çabuk çözülebilir.

* Artırımsal Geliştirme Süreç Modeli:

1. Sistem tek seferde teslim edilmez. Parça parça teslim edilir ve her seferinde daha fazla işlevsellik kazanır.
2. Kullanıcı gereksinimleri öncelik sıralarına göre sınıflandırılır ve en öncelikli gereksinimler erken Bir geliştirme başladığında ona ait gereksinimler dondurulur ve bu sırada herhangi bir değişiklik meydana gelirse sonraki teslimlerde dikkate alınır.
3. Erken teslimler prototip görevi gördüğünden projenin tamamen yanma olasılığı azalır.
4. Öncelikli gereksinimleri içeren sistem işlevleri daha fazla test edilmektedir.



* Kodla ve Düzelt Modeli:

1. Çok karmaşık olmayan ve birkaç yüz satırdan oluşan programlarda kullanılır.
2. Direkt olarak yazılım ürünü geliştirilmektedir.
3. Yazılımın bakım safhası vardır fakat çok zordur çünkü herhangi bir belge, plan vs. bulunmamaktadır.
4. Yazılımın bir emeklilik safhası vardır.

**Yazılım Yaşam Döngülerinin Avantaj ve Dezavantajları**

Çağlayan (Şelale Modeli)  
  
*Avantajları*:  
1- Kullanımı ve anlaması basittir.  
2- Yönetimi kolaydır.  
3- Projenin safhaları ayrı olduğundan iş bölümü ve iş planı projenin en başında net bir şekilde bellidir. Bu durum projenin yönetimini de oldukça kolay hale getirir.​​   
  
*Dezavantajları*:  
1- Proje ekiplerinin tasarım ve analiz aşamalarında çok detaylı çalışması gerekmektedir. Ancak yazılım ekipleri bir an önce programı yazıp sonucu görme eğiliminde olduklarından bu modelle yapılan projelerde ekip mutsuzlaşır.  
2- Karmaşık projeler için uygun değildir.  
3- Nesne yönelimli projeler için de uygun değildir.  
4- Yapılan her değişik maliyeti çok büyük oranda artırır.  
5- Müşteri gereksinimlerine tamamen uygun bir sonuç çıkarmak çok zordur çünkü gereksinimler zamanla değişebilir ve bu model gelişim ve değişimlere pek açık bir model değildir.  
6- Ürün ancak son aşamada müşteri karşısına çıktığından büyük sorunlar çıkabilmektedir.

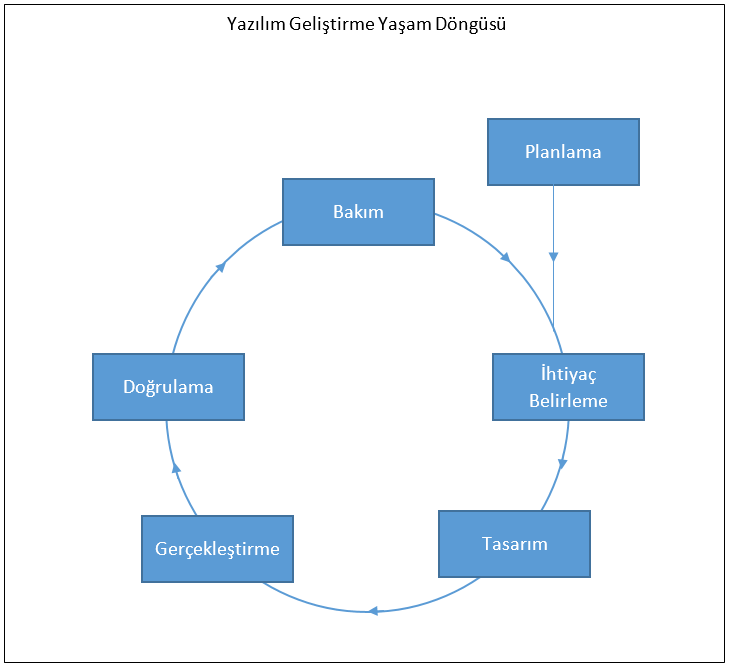
V Süreç Modeli:

*Avantajları*:  
1- Kullanıcının projeye katkısını artırmaktadır.  
2- Hataların bir sonraki aşamaya geçmesi önlenmektedir.  
3- Basittir ve kullanımı kolaydır.

*Dezavantajları*:  
1- Yazılımın erken prototipleri üretilmez.  
2- Uygulama şekli kesin kurallara bağlıdır.  
3- Fazlar arasında tekrarlamaları kullanmaz.

Helezonik (Spiral) Model:  
  
*Avantajları*:  
1- Her döngünün başında risk analizi yapıldığı için zaman ve maliyet bileşenleri kolay tahmin edilebilmektedir.  
2- Kullanıcı geribildirimine önem verilir.  
3- Daha sonraki aşamalarda eklemeler veya değişiklikler yapılabilir.  
4- Prototip oluşturma küçük parçalar halinde olduğundan daha temkinli ilerlenebilir.  
  
*Dezavantajları*:  
1- Düzgün çalışması için model aşamalarına birebir uymak gerekmektedir.  
2- Küçük projeler için spiral model çok fazla harcamaya sebep olabilir.

Artırımsal Geliştirme Süreç Modeli:   
  
*Avantajları*:  
1- Hızlı ve erken çalışan bir yazılım oluşturur.  
2- Riski yönetmek kolaydır.  
3- Müşteriden tüm sürümler için değerlendirme alınabilmektedir.  
  
*Dezavantajları*:  
1- Toplam maliyet şelale modelinden daha yüksektir.  
2- Tüm sistemin geliştirmesi başlamadan önce net ve iyi şekilde tanımlanmış olması gerekir.  
  
Kodla ve Düzelt Modeli:   
  
*Avantajları*:  
1- Yazılım geliştirmenin en kolay yoludur.   
2- Birkaç yüz satırdan oluşan programlarda işe yarar bir çözüm olmaktadır.  
  
*Dezavantajları*:  
1- Bakım aşaması çok zordur.  
2- Projenin emeklilik safhası vardır.  
3- Yazılım geliştirmenin en kolay yolu olmasına karşın en pahalı yoludur.

**AGILE (ÇEVİK) YAZILIM GELİŞTİRME**  
- Agile yöntem yazılım geliştirmedeki proje yönetimine özel bir yaklaşımdır.  
- Yazılım sektöründe işlerin zamanında ortaya çıkarılamaması, yazılımın müşteri gereksinim değişimlerine ayak uyduramaması ve zaman içinde sistemin sorunlarına çözüm bulunamaması sonucu agile (çevik) yöntemler açığa çıkmıştır.  
- Çevik yazılım geliştirme metotları, verimliliği yüksek, esnek, hata oranı düşük , hızlı ve ucuz çözümler sağlamaktadır.  
- Çevik yazılım geliştirme metodolojisinde projenin ölçeği fark etmeksizin her proje daha küçük yinelemelere ayrılır ve bu yinelemeler başlı başına bir projeymiş gibi ele alınarak geliştirilir. Projenin küçük parçalara ayrılarak geliştirilmesi oluşan veya oluşabilecek hataların daha çabuk fark edilmesini ve çözülmesini sağlar.

- Ayrıca her yineleme sonrası müşteriye projenin son durumu hakkında bilgi verilmektedir.  
- Çevik ile yinelemeler ortalama 2 – 4 hafta kadar sürmektedir.  
- Çevik geliştirmenin hızlı olma sebebi ekip üyelerinin sürekli iletişim halinde olmasıdır.

Agile Metodunun Genel Prensipleri:

* Müşteriyi memnun etmek ve sürekli yazılım geliştirmek.
* Teslimatı mümkün olan en kısa sürede yapmak.
* Projeler motive olmuş insanlarla devam etmelidir. Onlara uygun ortam ve ihtiyaç duydukları destek sağlanmalıdır. İşlerini yapmak için güvende olmalıdırlar.
* Mümkün olduğunda yüz yüze iletişim kurulmalıdır.
* Teknik mükemmellik ve iyi tasarıma sürekli dikkat etmek çevikliği artıracaktır.
* Düzenli aralıklarla, takımın nasıl daha etkili olacağına dair düşünülmeli ve davranışlar buna göre düzenlemelidir.

En Yaygın Uygulanan Çevik Metodolojiler

* Extreme Programming (XP)
* SCRUM
* Agile Unified Process
* Feature Driven Development (FDD)
* Test Driven Development (TDD)
* LEAN Development
* Dynamic System Development Methodology (DSDM)
* Microsoft Solution Framework (MSF)

**SCRUM**

SCRUM, Agile proje yönetim metodolojilerinden biridir. Karışık yazılım süreçlerinin yönetilmesinde kullanılmaktadır. Karmaşık yazılım işlerini küçük parçalara (sprint) bölerek geliştirir. Düzenli geri bildirim ve planlamalarla hedefe ulaşmayı sağlar. Bu anlamda ihtiyaca yönelik ve esnek bir yapısı vardır. Müşteri ihtiyacına göre şekillendiği için müşterinin geri bildirimine göre yapılanmayı sağlar. İletişim ve takım çalışması çok  
önemlidir. 3 temel prensip üzerine kurulmuştur.

1. Şeffaflık: Projenin ilerleyişinin herkes tarafından görülebilir olmasıdır.
2. Denetleme: Projenin ilerleyişinin düzenli olarak kontrol edilmesidir.
3. Uyarlama: Projenin yapılabilecek değişikliklere uyum sağlayacak yapıda olmasıdır.

Scrum Kavramları:

1. Product Backlog: Yüksek seviyeli gereksinim listesidir.
2. Sprint Backlog: Sprint Backlog Development Team’in oluşturduğu ve Development Team’in sprint’te almış olduğu işleri ve bu işleri nasıl tamamlayacaklarına ilişkin stratejilerini içeren bir çıktıdır.
3. Scrum Daily Meeting: Takımla her gün yapılan maksimum 30 dakika süren toplantılardır.
4. Scrum Board: Yapılması gerekenlerin yazıldığı ve yapılanların işaretlendiği tahtadır.
5. Burndown Chart: Sprintin günlerini ve kalan işi gösteren grafiktir. Scrum’ın şeffaflık ilkesini sağlar.

Scrum’da Roller:

1. Product Owner (Ürün Sahibi): Takım ve müşteri arasındaki iletişimi sağlayan kişidir. Proje özelliklerini tanımlar. Projenin iş değeri açısından geri dönüşünden sorumludur. Projeyi iptal etme yetkisine sahiptir. Ürünün çıkış tarihini belirler.
2. Scrum Master (Scrum Yöneticisi): Takımın Scrum’ın temel ilkelerine uymasını sağlar. Ayrıca takımdan tam verim alınabilmesi için olumsuz durumları ortadan kaldırır.
3. Scrum Team (Scrum Takımı): 5-9 kişi arasında birbirleriyle sürekli iletişim halinde olan ve tek bir hedefe ulaşmaya çalışan takımdır.

Toplantılar: Sprint Planlama, Sprint Gözden Geçirme, Günlük Scrum Toplantıları, Sprint Değerlendirme Toplantısı

1. Sprint Planlama:
   * + - Bir gereksinim listesi çıkarılır.
       - Ne? ve Nasıl? Sorularının sorulduğu kısımdır.
       - Risk değerlendirmesi ve risk kontrolleri yapılmaktadır.
     1. Sprint Gözden Geçirme:
        + Sprint Gereksinim Listesi oluşturulur.
        + Takım, ürün sahibi tarafından önceliklendirilmiş gereksinimlerden ne kadarını yapacağını belirler.
     2. Günlük Scrum Toplantıları:
        + Her iş gününde gerçekleştirilen tüm takımın katıldığı kısa süreli toplantılardır.
        + Dün ne yapıldığı, bugün ne yapılacağı gibi konular konuşulur.
     3. Sprint Değerlendirme Toplantısı:
        + Toplantıyı yöneten kişi Product Owner’dır.
        + Önceki Sprint’te yapılanlar takım dışındaki kişilerle paylaşılır.
        + Takım üyeleri bitirilen işleri açıklar ve soruları cevaplar.
        + Takım üyeleri nelerin iyi gidip gitmediğini paylaşır. Problemler için çözüm önerisinde bulunabilirler.

Bileşenler:

Ürün Gereksinim Dokümanı (Product Backlog): Proje için gerekli olan gereksinimler listesidir. Devamlı bakım yapılması gerekmektedir. Gereksinimler ürün sahibi tarafından önem sırasına göre alınır. Değişen gereksinimlere göre ürün sahibi gereksinimlere göre ekleme çıkarma yapabilir. Ayrıca takımları daha küçük parçalar haline de getirebilir. Ürün gereksinim dokümanı kullanıcı gözünden bakılarak oluşturulmalıdır.

Sprint (Koşu) Dokümanı (Sprint Backlog): Product Backlog’dan elde edilen iş ve görevleri kapsar. Amacı son ürünün bir işlevselliğini geliştirmektir. Sprint Backlog yalnızca takım tarafından değiştirilebilir.

Sprint Kalan Zaman Grafiği: Yapılan işleri ve yapılması gereken zamanı birlikte gördüğümüz grafiktir.

**REFERANSLAR**:

* <http://www.aspmvcnet.com/tr/m/yazilim-muhendisligi/selale-yontemi-waterfall-modeli.html>
* <https://medium.com/@gucalaliriza/scrum-dc15d9c53eae>
* <http://www.yilmazcihan.com/scrum-toplantilari-ve-rituelleri/>
* <https://medium.com/@secilcor/scrum-nedi%CC%87r-6a4326951dd8>
* <https://www.linkedin.com/pulse/sprint-backlog-nedir-ceyhun-yavuz/?originalSubdomain=tr>
* <https://akademi.bilgeadam.com/scrum-nedir/?gclid=CjwKCAjwxOCRBhA8EiwA0X8hi37s9SAzGml960J49zxr-EQbQ1YczTEuh7a4qnWGtc5GVPQNr8h_6BoCiDIQAvD_BwE>
* <https://toptalent.co/agile-nedir-agile-metodu-nasil-uygulanir>
* <https://medium.com/@omerharuncetin/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BC-modelleri-543c7879a742>
* <https://www.yazilimtestmerkezi.com/post/yazilim-gelistirme-modeli-nedir-cesitleri-nelerdir#:~:text=V%2Dmodelinin%20avantajlar%C4%B1%3A,Hatalar%C4%B1n%20bulunmas%C4%B1%20erken%20a%C5%9Famada%20bulunur>.
* <https://tr.csstricks.net/8222532-spiral-model-when-to-use-advantages-and-disadvantages#menu-4>
* <https://medium.com/@denizkilinc/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-temel-a%C5%9Famalar%C4%B1-software-development-life-cycle-core-processes-197a4b503696>
* <http://www.aspmvcnet.com/tr/m/yazilim-muhendisligi/selale-yontemi-waterfall-modeli.html>
* <http://www.aspmvcnet.com/tr/m/yazilim-muhendisligi/v-model.html>
* <http://www.aspmvcnet.com/tr/m/yazilim-muhendisligi/helezonik-tasarim-spiral-tasarim.html>
* <https://medium.com/@esra_/spiral-model-nedir-avantajlar%C4%B1-dezavantajlar%C4%B1-89c7ddffaae0>
* <https://www.pngwing.com/tr/search?q=%C5%9Eelale+modeli>