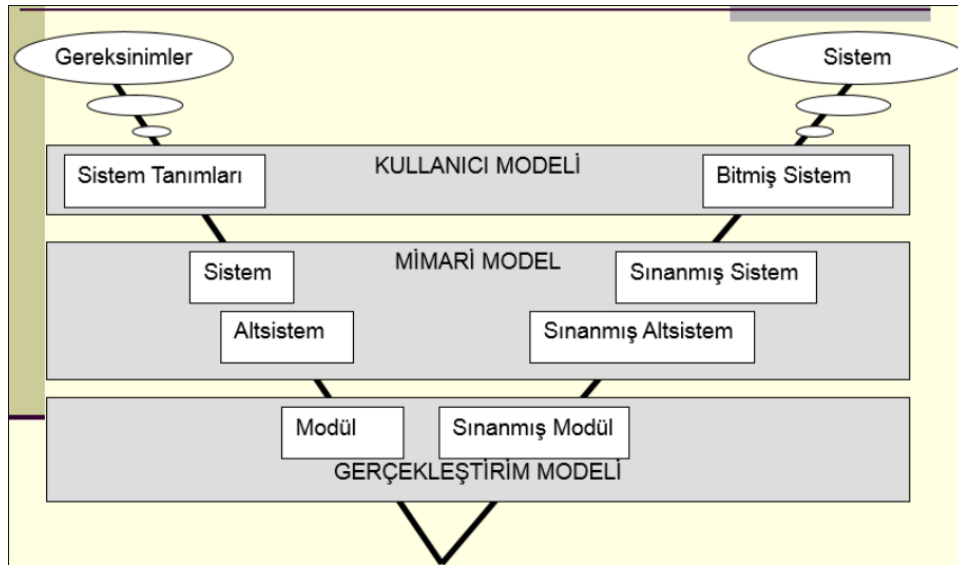


**V Süreç Modeli:** Gelişmiş bir şelale modeli gibidir. İş bölümü olan, tanımı iyi projeler için kullanımı kolaydır. Üretim ve deneme süreçleri kendi içinde döngüleştiği bir modeldir. Üç farklı modelden oluşur



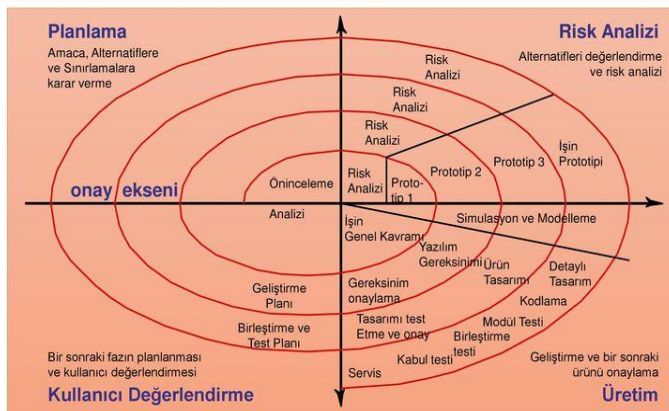
Kullanıcı modeli, projenin analizi ve işlevlerini belirlediğimiz modeldir. Projenin bitmemiş hali teslim edilir.

Mimari modeli, projenin tasarımı ve bileşenleri tamamlanır. Alfa testleri burada yapılır (Sınanmış Sistem).

Gerçekleştirim modeli, kodlamanın ve kod testlerinin yapıldığı bölümdür (Sınanmış Modül). V Modelinin sorunları ise, risk analizinin olmaması ve süreçte karşılaşılan sorunların çözümü için geç kalınabilmesidir.

**Helezonik(Spiral) Model:** Risk analizinin ön planda olması ve her aşamada prototip oluşturulması nedeniyle günümüzde oldukça yaygın kullanılan bir modelleme biçimidir. İyi tasarlanmış ve iş bölümü yüksek olan projeler için uygundur. Dört kısımdan oluşur.

## Helezonik Model (Spiral Model)



- Planlama: Her aşama sonunda oluşanlar için yeniden planlama yapılır.
- Risk Analizi: Riskler analiz edilir
- Üretim: Ana ürün ve ara ürünler oluşturulur
- Kullanıcı Değerlendirme: Her aşama sonunda oluşan ara ürünler kullanıcı yorumu alınır. Ve bu yorumlar ardından gerekli iyileştirmeler yapılarak diğer aşamaya geçilir.

Küçük projeler için maddi bakımdan ve süre açısından zorluklar yaratır.

**Artımsal Geliştirme Modeli:** Bu modelde proje parçalanır ve kullanıcının isteğine göre bu parçalar sıralanır. Bu parçalar birleşerek ara ürünü oluşturur ve kullanıcı tarafından kullanılır. Ara ürünler olduğu gibi durmaz. Sürekli yenilikler getirilir. Bu modeldeki amaçlanan şey üretim sürerken, kullanıcının kullanıp, yorumlayıp geri bildirim vermesini amaçlar. Bu model daha çok uzun süreli ve belirsizliğin fazla olduğu projelerde kullanılır.

**Evrimsel Geliştirme Modeli:** İlk tam ölçekli modeldir. Büyük alanlara yayılmış, büyük firmalar için önerilir. Her aşamada üretilen ürün tam işlevselliğe sahiptir. Modelin başarısı ilk düzenin başarısına bağlıdır.

**Çevik modeller:** Ekip içi diyalogu arttırmak ve dinamik bir grup oluşturmak için kurulan modellerdir. Başlıca 2 tane çevik grup bulunur. Günümüzde modelleme için kullanılan en popüler kavramlar bu başlık altındadır.

**Extreme Programming:** 1996 yılında kurulan sistem, projeyi 4 başlıkta ele alır.

- Basitlik: Yazılan kodun ve yapılan işin sade, anlaşılır ve karmaşık olmadan yapılmasını gerektirir. Uzun uzun dokümantasyondan uzak durulur.
- Cesaret: Projeye devam edilirken cesur olunmalıdır. Projede cesurca karar almayı bilmek gerekmektedir.
- Geri Dönüş: Müşteri ve ekip arasındaki iletişim sürekli korunmalıdır. Müşterinin isteği veya karşılaştığı hata karşısında hemen geri dönüş yapılabilmelidir.
- İletişim: Ekip içi diyalogu koruyup çalışma ortamını ekip için daha kolay bir hale getirmek amaçlanır.

**SCRUM:** Projeyi “Sprint” denilen parçalara ayıran bu modelde her çevik modelde olduğu gibi ekip içi iletişim önemlidir. SCRUM temelde 3 ögeden oluşur. Bunlar rol, toplantı ve araçlardır.

**Rol:** Ürün sahibi, Scrum beyni ve Scrum ekibinden oluşur. Ürün sahibi kısaca projenin lideridir. Scrum beyni ekibi Scrum kurallarına göre kontrol eder. Scrum ekibi ise projeyi yapan 5-11 kişilik yazılım ekibidir.

Toplantı: Scrum'ın olmazsa olmazı toplantılardır. Toplantılar günlük ve haftalık olur. Her sprint için ayrı toplantılar yapılır. Beynin ekibine o gün yapacaklarını, dün yaptıklarını anlattığı toplantılardır.

Araç: Envanter ve gereksinimler oluşturulur. Her sprinte ait dokümantasyon yapılır. Zamanlama ve çizelge oluşturulması da yapılır

Scrum günümüzde en çok kullanılan yöntemdir. Sadece yazılımda değil ekip içeren her olguda kullanılabilir.

### **Hangi Yazılım Geliştirme Modeli?**

Gelişigüzel ve Barok modelleri geri dönüt olmaması ve döngüden mahrum olmaları nedeniyle artık kullanılmamaktadır.

Şelale modeli küçük projeler için oldukça kullanışlıdır. Her adımda döngüye girmesi nedeniyle hala kullanılabilir bir modeldir.

V Süreç modeli, gelişmiş bir şelale modeli olarak adlandırılabilir ve kullanışlılığını hala korumaktadır. Üretim ve Deneme süreçleri kendi içinde döngülenen bir model olduğu için uygulaması ve takibi kolaydır. Orta büyüklükteki projeler için uygundur. Risk analizi basamakları olmaması gibi sorunlar içerir.

Spiral modelde kullanıcının her adıma olması bir avantajdır. Büyük projeler için idealdir. Hatalar erken giderilir. Risk analizi yapılır. Karmaşık olması, uzun sürmesi ve dokümantasyonun fazla olması başlıca dezavantajlarıdır.

Artımsal modelde kullanıcı projeyi kendi isteğiyle projeyi parçalar ve bu kullanıcı memnuniyeti sağlar.

Çevik modelde, müşterilerle iletişim ve ekip içindeki iletişim yüksektir, hatalar en aza indirgenir, uyarlanması kolaydır, değişiklik yapmak için açık ve esnektir, ekip önemli bir konumdadır, ancak dokümantasyon çok ayrıntılı olmadığı için talep nedeniyle çok sıra dışıdır. Sürekli değişiyor, bu yüzden çalışma saatleri aşırı olabilir. Bugün yaygın olarak kullanılan bir model. "Evrimsel geliştirme yönteminde", kullanıcılar gereksinimleri daha iyi anlayabilir ve hataları azaltabilir, ancak süreç daha az açıktır ve bakımı zordur.

### **Hangi Model Benim Projeme Daha Uygun?**

V modeli, düşük belirsizliği ve net iş tanımları olan bilgi teknolojisi projeleri için uygun bir modeldir.

Evrimsel gelişim yöntemleri geniş bir kitleyi çekecek projeler için kullanılabilir.

Büyük, pahalı ve uzun vadeli projelerde spiral model veya artımlı geliştirme modeli uygundur.

Çevik model, uzun süre dayanmayan küçük ve orta ölçekli projeler için uygundur.

Şelale modeli küçük ve iyi tanımlanmış projeler için kullanılabilir.

## **SCRUM Modeli Günümüzde Neden Daha Yaygın?**

SCRUM günümüzde en yaygın kullanılan yazılım geliştirme yöntemidir. Sadece yazılım geliştirme için değil, aynı zamanda birçok sistemin geliştirilmesi için de kullanılır. Sebebi zaman ve para tasarrufu, o yüksek teknolojiye ve en son gelişmelere kolayca uyum sağlayabilme, tam olarak tanımlanmamış karmaşık projeler için çok uygun olmasıdır. Ekip içinde yüksek düzeyde iletişim sağlamak,ve hataları erkenden tespit etme ve düzeltme, kullanıcılardan sürekli geri bildirim alması süreci uzatır. Ancak sorun azalır. Diğer yazılım geliştirme yöntemleri gibi, değişen ihtiyaçlara hızlı cevap verir.