**YAZILIM YAŞAM DÖNGÜSÜ VE MODELLERİ**

**(SOFTWARE DEVOLEPMNENT LİFECYCLE AND MODELS)**

Yazılım yaşam döngüsü, (SDLC) geliştirmekte olduğumuz projenin planlanmasından itibaren teslimine kadar belirli aşamaların uygulandığı sürece verilen addır. Yazılım yaşam döngüsü 6 aşamaya bağlıdır fakat bu aşamalar farklılık gösterebilir. Aşamalar şu şekildedir; Planlama, gereksinim, analiz, tasarım, gerçekleştirme, bakım. Sürecin tamamlanması için her aşamayı bir kere uygulamak yeterli olucaktır.



* **Aşama 1: Planlama**
* **Aşama 2: Gereksinim**
* **Aşama 3: Analiz**
* **Aşama 4: Tasarım**
* **Aşama 5: Gerçekleştirme**
* **Aşama 6: Bakım**

**Planlama:** İlk aşamadır. Alınan işin projelendirilmesi ve proje için yapılan planların oluşturulma aşamasıdır. Ekip içi görev dağılımı bu aşamada gerçekleşir.

**Gereksinim:** İş için gerekli olan bilgiler Gereksinim Analizi teknikleri tarafından belirlenerek bu doğrultuda fizibilite çalışması yapılır.

**Analiz:** İşimiz doğrultusunda çıkardığımız gereksinimler analiz edilerek iş hakkında daha detaycı ve hatacı yaklaşılır.

**Tasarım:** Oluşturulacak yazılımın ilk ve temel halinin oluşturulması ve sürecin planlanması. Kodlama olarak herhangi bir durum söz konusu yoktur

**Gerçekleştirme:** Gereksinimler doğrultusunda tasarımı oluşturulan sistem bu aşamada kodlanır. Kodlamayı içeren bir aşama olduğu için kaliteli ve basit bir kodlama için önem gerektiren bir aşamadır. Test kısımlarınıda barındırır.

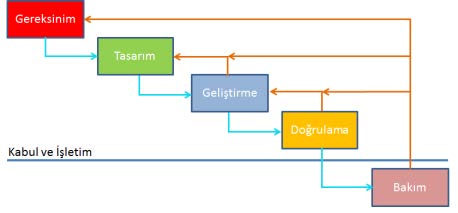
**Bakım:** Tamamlanmış ürünün fiziksel ve yazılımsal olarak bakımı gerçekleştirilir. Oluşacak ya da oluşan hataların çözümü ve güncellemeleri kapsar.

**Yazılım Geliştirme Süreci Modelleri**

**Gelişigüzel Model:** Herhangi bir model veya yönteme bağlı kalınmadan yapılır. Proje konusunda sorumlu kişiye bağımlıdır. Anlaşılabilirlik ve bakım konusunda zorluklar yaşanabilir.1960’lı yıllarda kullanılmaktaydı.

**Barok Modeli:** Yazılım yaşam döngüsü başlıca adımlarının direkt olarak gerçekleştiridiği modeldir. Bu model belgelemeyi ayrı bir süreç olarak ele alır. 1970’lı yıllarda kullanılmaktaydı.

**Çağlayan Şelale (Waterfall) Modeli:** Yazılım geliştirme yöntemlerinin temeli sayılmasına rağmen günümüzde en kötü modellerden biri olarak kabul edilmektedir. Aşamalar birbirini takip eder bir aşama bitmeden diğer aşamaya geçilmesi mümkün değildir. Her aşamada test gerçekleştirilmek zorundadır. Şelale modelinde yazılım geliştirme süreci analiz, tasarım, kodlama, test, ve bakım gibi aşamalardan oluşmaktadır.

****

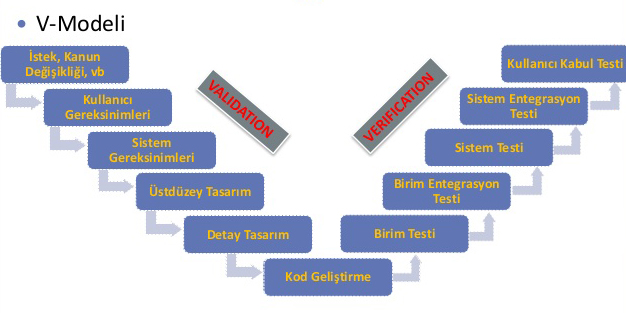
Modelin verimli kullanımı içini çok küçük ve gereksinimleri iyi anlanmış projeler gereklidir.

Müşteri memnuniyeti sağlamak çok zordur çünkü gelişim ve değişime açık bir model değildir.

Oluşabilecek sorunlar geç fark edilirse maliyetini büyük oranda arttırır. Bu durum istenmeyen bir durumdur.

**V Süreç Modeli:**

V ModeliŞelale (Waterfall) modelinin gelişmiş hali gibi düşünülebilecek bir yazılım süreci sunar.Doğrusal bir modeldir. Her aşama kendi kontrol aşamasıyla eşleşmiştir ve bundan dolayı “V” harfine benzeyen bir şekil ortaya çıkmıştır. Belirsizlikleri az, yapılacak işlerin belirgin olduğu projeler için uygundur, örneğin Bilgi Teknolojileri projeleri için uygundur.



Proje yönetimi tarafından takibi kolay olan bir proje olarak kabul edilmektedir.

Geliştirme süreci devam ettikçe iş ve ürün gereksinimleri de değişkenlik gösterebilir.

**Helezonik (Spiral) Model:**

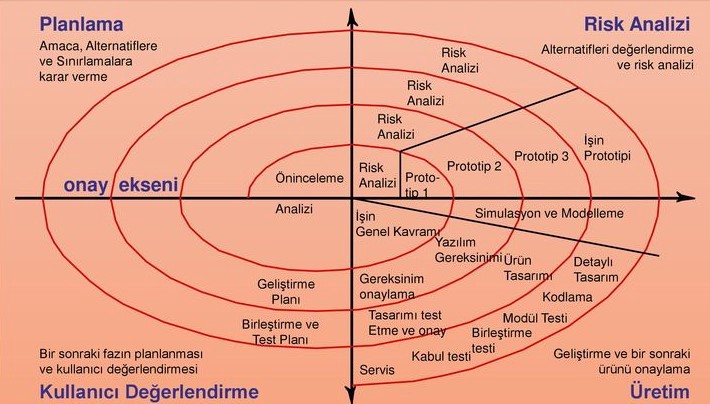
Bu modeli diğer modellerden ayıran en önemli unsurumuz risklerin analizinin en ön planda olması ve prototip oluşturulmasıdır. Temel olarak dört ana bölümü içerir Bunlar; planlama, risk yönetimi, üretim ve kullanıcı yorumu olarak tanımlanabilir bu ana bölümler ;

Planlama: Her ara ürün için iş planlanır.

Risk Yönetimi: Alternatifler değerlendirilerek risk analizi yapılır.

Üretim: Ara ürün ya da ürünün üretilmesi.

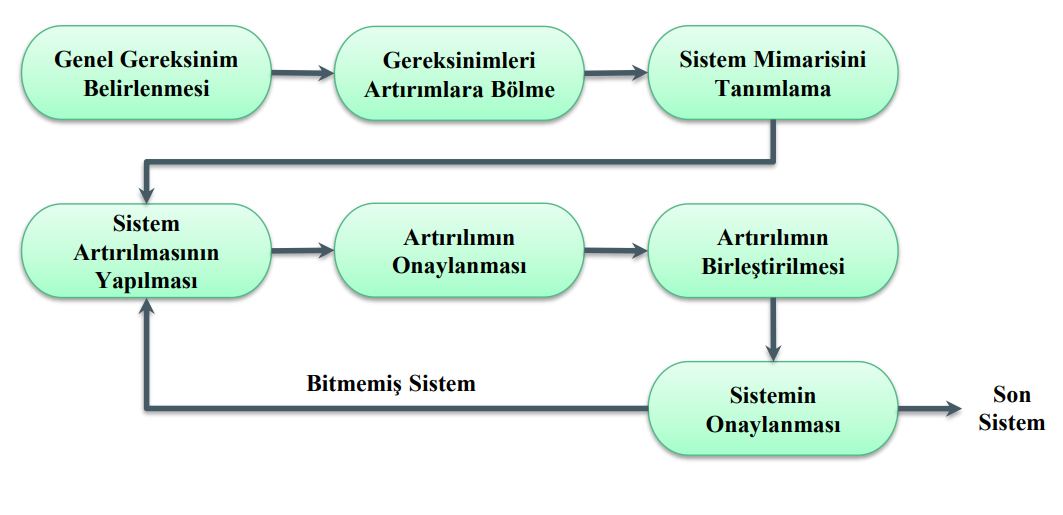
Kullanıcı Yorumları: Üretim sonucunda oluşturulan ara ürünün kullanıcı tarafından alınan yorumların değerlendirilmesi yapılır

****

Dezavantajları; Komplekstir(karmaşık), spiral sonsuza gidebilir, risk değerlendirilmeleri özneldir.

**Artımsal Geliştirme Modeli:**

Sistem tek seferde teslim edilmek yerine, parçalara bölünerek teslim edilir. Divide and Conquer (Böl ve Yönet) yaklaşımıdır. Oluşturulacak bu parçaların sıralamasını ise kullanıcı belirler. Bölünen parçalar sırasıyla tamamlanarak her parça için bir ara ürün oluşur ve oluşan bu ürünler kullanıcı tarafından kullanılır. Üretilen her yazılım sürümü zamanla yeni işlevsellikler ve özellikler kazanmak zorudadır.

****

Bir tarafta kullanıcılar tarafından deneyimlenirken aynı zamanda üretimde devam etmektedir.

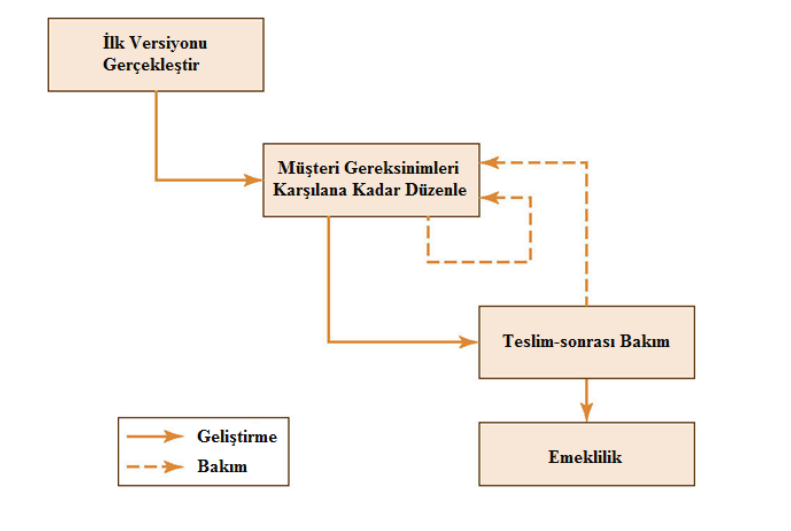
Projenin başarısız olma riski azdır.

Bazı dezavantajları ise; Deneyimli personel gerektirir, Parçaların kendi içinde tekrarlanmalarına izin vermez.

**Kodla ve Düzelt Yaşam-Döngü Model:**

Küçük projeler için kullanılabilir. Direkt olarak yazılım ürünü oluşturulur. Bakım safhası vardır fakat gerçekleştirilmesi oldukça zordur sebebi ise dökümantasyon olmayışıdır.

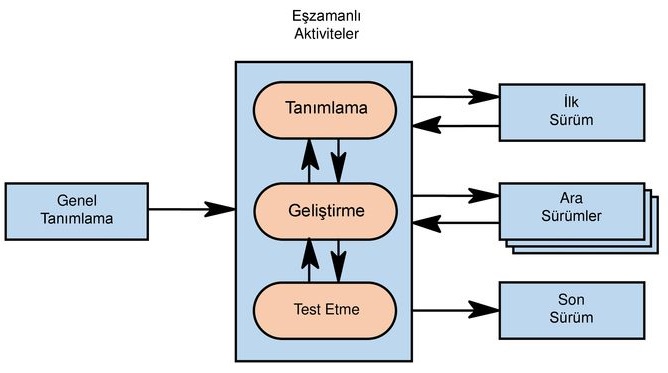
Ayrıca emeklilik (retirement) safhası vardır.



**Evrimsel Geliştirme Modeli:**

Diğer modeller ile kıyaslandığında süreç daha yavaş bir şekilde işlemektedir. Modelin başarısı ilk evrimin başarısına bağlıdır. Gereksinimleri Anlamak kolaydır. Risk ve Hata azdır.

Dezavantajları ise; Bakım zor, düzeli ürün oluşumu yok, gereksinimi sürekli yenilemek gerekebilir.



**Çevik Modeller:**

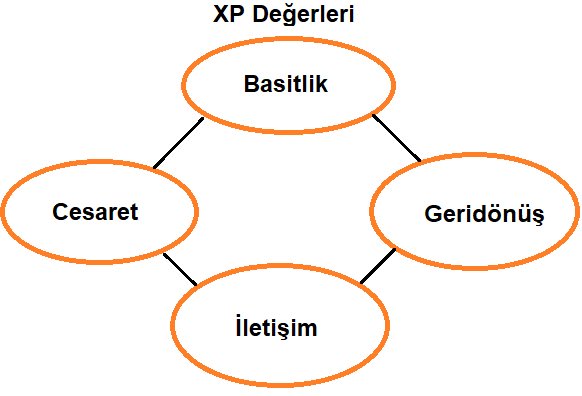
Çevik modelleme, yazılım sistemlerini etkili ve verimli bir şekilde modellemeye ve dökümantasyonunu yapmaya yönelik ortaya çıkmıştır. Ekip içi iletişim oldukça fazladır bir diğer amacıda budur. Bazı çevik yazılım geliştirme modelleri şunlardır:

**Extreme Programming (XP):**

En popüler çevik süreçlerden birisidir. Kent Beck ve arkadaşları tarafından 1966 yılında oluşmuştur.

Bu çevik süreçte müşteri ve gereksinimleri merkezi bir rol oynamaktadır. Tam olarak belirlenemeyen ve çabuk değişen müşteri gereksinimlerine ayak uydurabilen bir programlama sürecidir.

XP dört değer üzerine kurulur: Basitlik(Simplicity), İletişim(Communacition), Geridönüş(Feedback) ve Cesaret(Courage).



Basitlik(Simplicity): Yazılan kodun ve yapılan işin sade, anlaşılır ve kompleksli (karmaşık) yapıdan uzak olması gerekir.

İletişim(Communacition): Projeler için önemli bir rol oynayan iletişim XP için sorun olmaması açısından üzerinde önemle durulması gereken bir noktadır.

Cesaret(Courage): Projede ilerlenirken cesaret önemli bir noktadır cesur bir şekilde korkmadan ilerlenmesi gerekmektedir.

Geridönüş(Feedback): Geridönüşler sayesinde program en iyi halini almaktadır ve hatalardan arınmaktadır. Müşteri ile iletişim halinde olmamız gerekir.

**SCRUM:**

Scrum günümüzde kullanılan en popüler çevik modellerden biridir. Scrum adı rugby oyunundan gelmektedir. Scrum, rugbyde topu almaya çalışan bir grup oyuncuyu kasteder. Bundan dolayı genelde büyük ve karmaşık projelerde kullanılmaktadır.

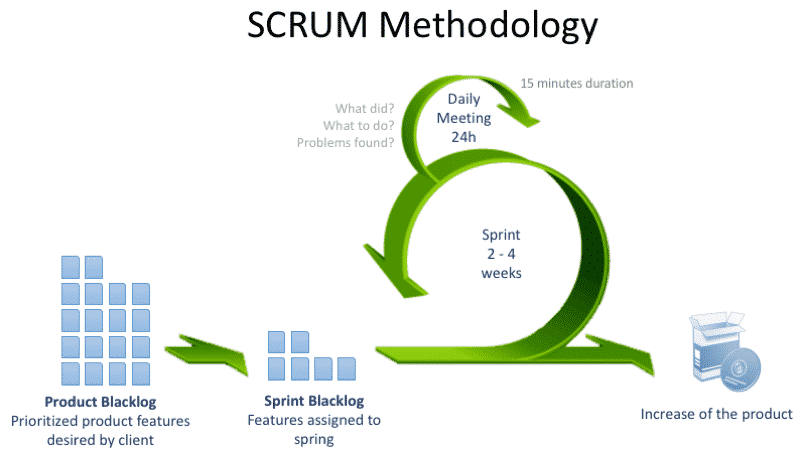
Scrum’ın kendi değerleri vardır. Bunlar; Taahhüt, cesaret, odaklanma, açıklık ve saygıdır. Bu değerler projenin işleyişi, verimi ve ekip üyeleri arası iletişimde büyük rol oynamaktadır.

**Scrum’ın çalışma mantığı:**

**Ürün İş Listesi (Product Backlog):** Müşteride alınacak dönütlere göre ihtiyaçların tutulacağı bir listedir. Ürünler hakkında maliyet ve amaçları hakkında bilgiler verir.

**Sprint İş Listesi (Sprint Backlog):** Belirli bir versiyon planı için çıkartılan iş listesidir. Listedeki işler için 4 hafta koşulur. Her sprint sonrası dışarıya verilebilecek bir increment çıkarılır.

**Scrum Günlük Toplantı (Scrum Daily Meeting):** Her gün scrum takımıyla ilerleyiş, yapılacaklar ve oluşan oluşabilecek sorunlar hakkında kısa toplantılar yapılır. Genellikle 15-30dk sürer



Scrum’da üç temel kavram bulunmaktadır. Bunlar; Roller, toplantılar, bileşenler/araçlar.

**Roller (Roles):**

Roller, ürün sahibi(product owner), scrum yöneticisi(scrum master) , scrum takımı(scrum team) olarak 3’e ayrılmaktadır. Ürün sahibi, projenin iş kısmından sorumlu olan kişidir ve büyük bir rol oynar. Scrum yöneticisi, takımının scrum değerlerine ve pratiklerine bağlılığını kontrol eder. Scrum takımı, 5-9 kişiden oluşan verilen hedefe odaklanan ve onun için mücadele eden takımdır.

**Toplantılar (Meetings):**

Toplantılar, Sprint(Koşu) planlama, sprint(koşu) gözden geçirme, günlük scrum toplantısı olmak üzere 3’e ayrılmaktadır.

* Sprint(Koşu) Planlama: Gereksinimlerden yararlanılarak geniş kapsamlı bir gereksinim listesi çıkartılır. Çıkartılan bu liste birimlere ayrılarak ekip üyelerine dağıtılır. Risk değerlendirmeleri ve risk kontrollerinnin belirlenmesi bu koşularda yapılmaktadır. Planlama kısmı olduğu için önemli ve itinayla yapılması gereken bir sprinttir.
* Sprint(Koşu) Gözden Geçirme: Her sprint sonunda yapılan toplantıdır. Yapılan sprint değerlendirilerek oluşan ürün karşılaştırılarak gözden geçirilir. Müşterinin istediği ürüne tam anlamıyla ulaşabilmek için önem teşkil etmektedir.
* Günlük Scrum Toplantısı (Daily Scrum Meeting): Her gün scrum takımıyla ilerleyiş, yapılacaklar ve oluşan oluşabilecek sorunlar hakkında kısa toplantılar yapılır. Genellikle 15-30dk sürer.



**Bileşenler / Araçlar (Artifacts):**

Ürün gereksinim dökümanı, sprint(Koşu) dokümanı ve sprint kalan zaman grafiği oluşturulur. Ürün gereksinim dökümanında proje boyunca yapılması gerekenler basit bir şekilde yazılır. sprint(Koşu) dökümanı, product backlog’dan elde ettiğimiz iş ve görevleri kapsamaktadır. Sprint kalan zaman grafiği ise yapılan işin planlanan ile karşılaştırılması ve zaman konusunda değerlendirilmesi sağlanılır.

Scrum günümüzde kullanılan en popüler çevik model olmasının bazı sebepleri şunlardır :

* Sorunları günlük toplantılarda ele alıp çözülebilir.
* Eşit iş paylaşımından dolayı motivasyon ekip üyelerinde yüksek tutulur.
* Müşteriden alınan geri bildirimlere önem verilir ve sürekli geri bildirim alınır.
* Zaman ve parasal açıdan çok fayda sağlar.
* Son gelişmelere kolay uyum sağlayabilmesi.
* Büyük ve karmaşık gereken projeler için biçilmiş kaftan olması.

**MODELLERİN KARŞILAŞTIRILMASI:**

Gelişigüzel ve barok modelleri günümüzde kullanılmamaktadır. Kullanılmamaların bazı sebepleri ise şunlardır; Takip edilebirlik ve bakım oldukça zordur, adımlar arası geri dönüşlerin tanımlanmaması ,dökümantasyon ve yineleme içermemeleri gibi sebepler bu modellerin günümüzde kullanılmama sebepleri arasında örnek verilebilir.

Çağlayan (Şelale) modeli, eskiden oldukça popülere olan bir modeldir fakat günümüzde kullanıcı ile iletişimin düşüklüğü ,ürünün kullanıcıya uzun zaman diliminde ulaşması, hataların düzeltilmesi için gerekli maliyetlerin fazlalığı gibi sebepler günümüzde kullanımını düşürmüştür.

V Süreç modeli , kullanımı ve takibi kolay olan bir modeldir ancak aşamalar arası tekrarlanmalara ve yinelemelere imkan sağlamadığı için günümüzde zorluk sağlamaktadır.

Spiral modeli, ara ürünlere yer verdiği ve bu ara ürünler ise müşteri ile etkileşimde olduğu için müşterinin etkileşimi bu modelde diğer modellere kıyasla daha fazladır. Böyle olması müşterinin ihtiyaçları doğrultusunda doğru ilerlemeyi sağlayarak istenilen ürünün çıkmasında önemli rol oynar.

Artımlı Model, parçalar halinde teslim edilir. Böyle olması modelin geri dönütlerinin hızlı alınmasına ve erken olarak çözüm bulunabileceğini işaret eder. Gereksinimler müşteri ile ortak çalışmalarla belirlenir ve özellikler çok kez tekrarlanarak en iyi hali bulunabilir sonradan eklemeler yapılabilir.

Kodla ve düzelt modeli küçük projelerde kullanılmaktadır oldukça öznellik içerir bundan dolayı bakımı zordur. Bu modelin emeklilik safhası bulunur.

Çevik modeller ekip içi iletişime ve müşterinin verdiği gereksinimlerin doğru bir şekilde uygulanmasına önem gösterir. Bundan dolayı hatalar minimumdur, verim çoktur ve istenilen gereksinimlere en yakın ürünler verilir. Günümüzde en çok kullanılan modeldir diyebiliriz.

Evrimsel geliştirme modeli bakımı ve sürece hakimiyet oldukça zordur.

**HANGİ PROJEDE HANGİ MODEL ?**

Şelale modeli ,sonradan özellikler ve eklentiler eklenmeyecek sektörlerde oldukça kullanılır. İnşaat sektörü örnek olarak verilebilir.

Çevik modeli, büyük şirketlerin karmaşık çözüm isteyen projelerinde kullanılabilir. Böl ve fethet düşüncesine göre hareket etmektedir.

Artımlı model, sürekli güncel tutulması gereken ve özelliklerin sonradanda eklenebileceği projelerde kullanılır. İhtiyaçlar doğrultusunda güncellenebilr. E-Ticaret siteleri örnek olarak verilebilir.

Kodla ve Düzelt küçük projelerde kullanılır.

V Modeli büyük projelerde kullanılabilir. İnşaat sektörü örnek verilebilir.

**KAYNAKLAR:**

* [**https://fikirjeneratoru.com/yazilim-proje-yonetimi-yontemleri/**](https://fikirjeneratoru.com/yazilim-proje-yonetimi-yontemleri/)
* [**https://medium.com/@omerharuncetin/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BC-modelleri-543c7879a742**](https://medium.com/@omerharuncetin/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BC-modelleri-543c7879a742)
* [**https://denizkilinc.com/scrum-nedir/#**](https://denizkilinc.com/scrum-nedir/)
* [**https://www.codex.com.tr/yazilim-gelistirme-modelleri**](https://www.codex.com.tr/yazilim-gelistirme-modelleri)
* [**https://zeynepaygun.wordpress.com/2017/05/29/what-is-sdlc-sdlc-nedir/**](https://zeynepaygun.wordpress.com/2017/05/29/what-is-sdlc-sdlc-nedir/)
* [**https://medium.com/@tunaytoksoz/yazilim-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-sdlc-ve-modelleri%CC%87-c3fe40f6e4e8**](https://medium.com/@tunaytoksoz/yazilim-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-sdlc-ve-modelleri%CC%87-c3fe40f6e4e8)
* [**https://www.nutcache.com/blog/what-is-scrum-methodology-and-project-management/**](https://www.nutcache.com/blog/what-is-scrum-methodology-and-project-management/)
* [**https://www.microsoft.com/tr-tr/microsoft-365/business-insights-ideas/resources/how-to-choose-the-best-agile-methodology-for-your-project**](https://www.microsoft.com/tr-tr/microsoft-365/business-insights-ideas/resources/how-to-choose-the-best-agile-methodology-for-your-project)
* [**https://medium.com/@denizkilinc**](https://medium.com/@denizkilinc)

HALİL ERDEM ÜNGÖR – 200601072