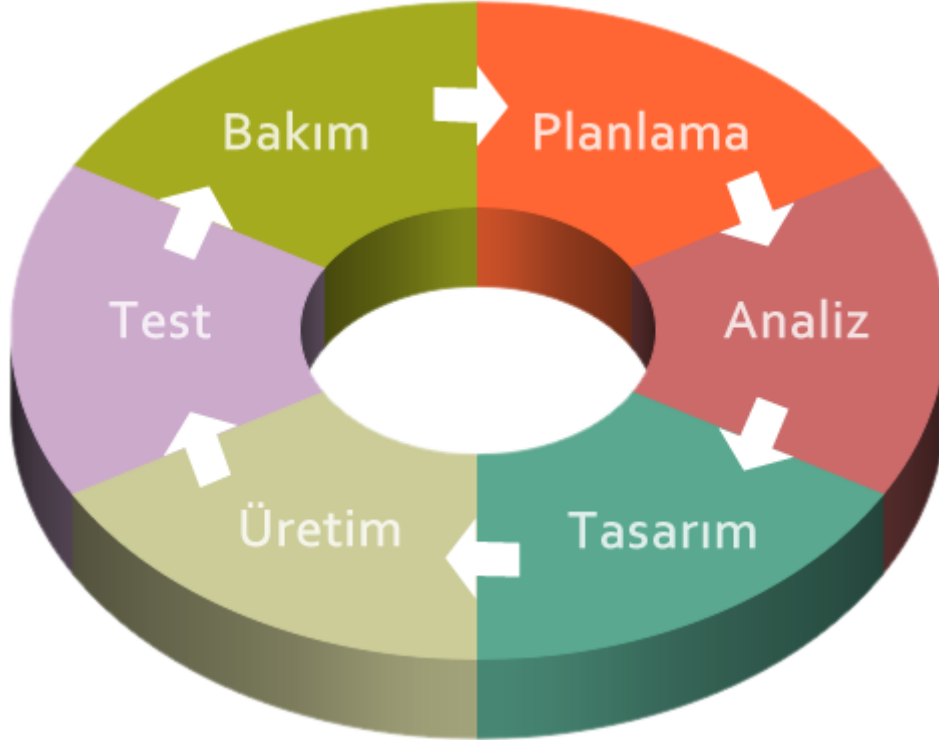


YAZILIM YAŞAM DÖNGÜSÜ MODELLERİ

İZMİR BAKIRÇAY ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ, Aydın ERALP, 210601030

ÖZET

Yazılım da tüm dünyada bizler için üretilen ürünlerden farksızdır aslında ve tüm ürünlerde olduğu gibi üretilen yazılım projelerinin de bir ömrü, bir yaşam süreci, yaşam döngüsü, vardır ve bu yaşam döngüsü tek yönlü olmadığı gibi içinde birçok da adım barındırır. Bir yazılım projesi geliştirmek sadece kodlama yapmak anlamına gelmez. Kodlamadan önce kullanılacak olan metodu, kodlama öncesi ve sonrası hangi adımları atacağımızı belirlememiz gerekir. En temeline baktığımızda bu adımlar ise aşağıda gösterildiği gibidir.



PLANLAMA: Yazılım yaşam döngüsünün başladığı, projenin hedefi olan müşteri, kullanıcı isteklerini ve tüm ihtiyaçları belirleyip tespit ettiğimiz adımdır. Bu adım tüm evrenin içinde küçük bir başlangıç adımı gibi gözükse de aslında baştan sona tüm süreç döngüsünü ve planlamaları kapsar.

ANALİZ: Planlamadan sonra karşımıza çıkan bu adımda proje artık somutlaşmıştır. Ne gibi riskler olacağı, projenin geliştirilmesinin ne kadar süreceği ve projenin tüm işlevleri detaylı olarak belirlenir. Kısaca analiz adımıyla projenin tüm detayları çıkarılır. Proje bu adımla birlikte artık plan olmaktan çıkmış, somut bir hal almaya başlamıştır.

TASARIM: Analiz adımıyla çıkartılan detaylar doğrultusunda proje planı oluşturulup yazılım projesinin hangi adımlarla, nasıl oluşturulacağı belirlenir yani bu yazılım projesinin temel yapısı ve şekli oluşturulur. Proje planının yanısıra tasarım dokümanları da oluşturulur.

ÜRETİM: Planlama, analiz ve tasarım aşamaları bitirilmiş ve bu aşamada belirtilen plan çerçevesinde artık proje gerçekleştirilmeye başlanmıştır.

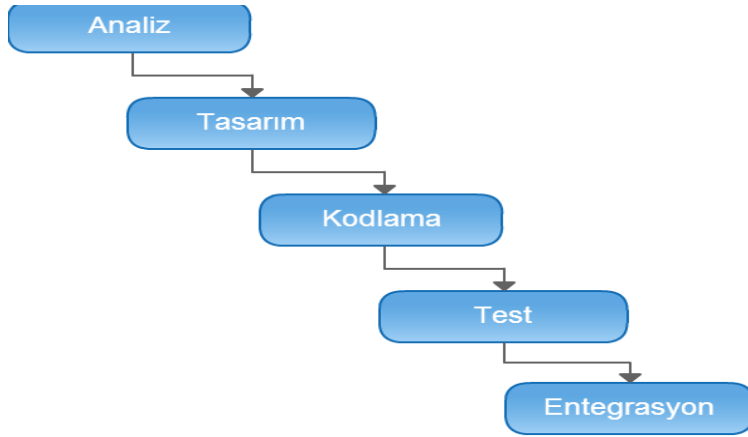
TEST VE BAKIM: Proje tamamlandıktan sonra müşteriye teslim edilmeden önce test ekibi tarafından denetlenir ve hatalar giderilir. Proje sunulduktan sonra ise ortaya çıkan hatalar giderilir ve kullanıcılardan gelen istekler doğrultusunda projede birtakım geliştirmeler söz konusu olabilir.

MODELLER

GELİŞİGÜZEL MODEL: Aslında bu yöntemi bir model olarak belirtmek doğru değildir. Çünkü diğer modellerle karşılaştırdığımızda herhangi bir yöntem gözetmeksizin yapılır. Tamamen ilgili kişiye bağlıdır. Genel olarak tek bir kişi tarafından yapılır. Bakımı oldukça zordur fakat çok basit bir programlamaya sahiptir.

BAROK MODELİ: Yaşam döngüsü temel adımları doğrusal bir biçimde ele alınır. Belgeleme diğer modellerden farklı olarak ayrı bir süreç olarak ele alınır. Döngü yoktur nasıl geri dönüş yapılacağı tanımlanmamıştır ve en çok gerçekleştirim adımına ağırlık verilmiştir bundan dolayı günümüzde kullanımı tavsiye edilmemektedir.

ÇAĞLAYAN YAŞAM DÖNGÜ MODELİ(ŞELEALE MODELİ): Bu model diğer güncel modellerin temeli diyebiliriz. Buna rağmen barındırdığı sorunlar nedeniyle günümüzde kullanımı gittikçe azalmaktadır. Barok modelinin aksine belgeleme ayrı olarak ele alınmamıştır, sürecin içerisinde. Kullanıcıyı sürece dahil etmediği için bazı sorunlar çok geç fark edilebilir ve bu geç fark edilen sorunlar pahalıya patlayabilir, süreci daha da uzatabilir. Geliştirilmeye hiç açık değildir. Analiz ve tasarım aşamalarının bu döngü modeli için hayati önem arz etmesi vaktin çoğunun bu iki aşamaya harcanmasına sebep olmaktadır. Çok iyi tanımlanan ve kısa sürede gerçekleştirilebilecek projeler için uygun denilebilir.



V SÜREÇ MODELİ: Şelale modelinin geliştirilmiş hali de diyebiliriz. Resimde de görüleceği üzere sol tarafa üretim sağ tarafa ise sınama bölümleri diyebiliriz. Avantajları:

-Şelale modelinin aksine geliştirmeler yapılabilir.

-Kullanımı ve takibi kolaydır.

Dezavantajlarına gelecek olursak:

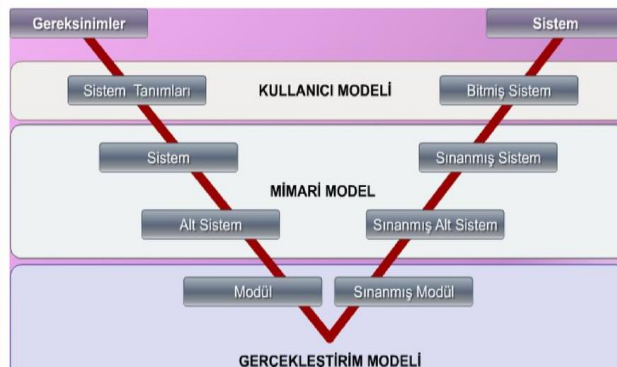
-Risk çözümleme için bir şey yapılmaz.

-Aşamalarda tekrarlar bulunmaz.

-Belirsizliklerin az olması ve iş tanımlarının belirgin olması bu modelin kullanımını kolaylaştıracaktır.

V süreç modeli genelde Bilgi teknolojileri için geliştirilen projelerde kullanılır.

V Modeli

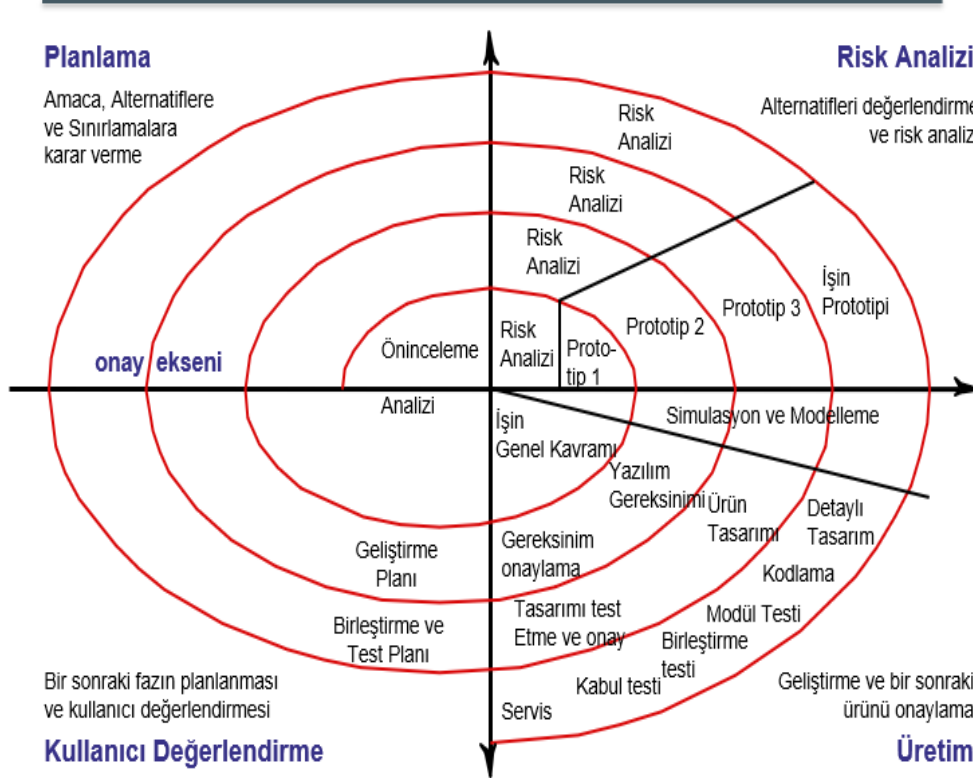


KULLANICI MODELİ: Sürecin kullanıcı ile ilişkili olan bölümünü tanımlar. Sistemin ne şekilde kabul edeceği doğrultusunda planlar yapılır.

MİMARİ MODEL: Sistemin tasarımı yapılır ve tüm sistemin sınanmasına yönelik işlemler yapılır.

GERÇEKLEŞTİRİM MODELİ: Kodlama ve sınama işlemleri yapılır.

HELEZONİK(SPIRAL) MODEL: Bu modeli diğer tüm modellerden ayıran en önemli özellikler risk analizinin ve çözümlenmesinin oldukça ön planda tutulması ve prototipler oluşturulmasıdır diyebiliriz. Risk analizinin ön planda tutulmasının en büyük avantajı da hataların erken fark edilerek Şelale Modeli gibi modellerin aksine geliştirmeler ve düzeltmeler için fazla maliyet çıkmasına engel olmasını sağlamasıdır. Prototip oluşturulmasının avantajı ise kullanıcı her aşamada projeye dahil olmuş olur ve yine ileride çıkacak olan sorunların birçoğu erkenden rahat bir şekilde halledilebilir. Pek çok yazılım modelini içinde bulunduran bir modeldir. Çoğu modelin aksine geliştirmeye gayet açık ve elverişli bir modeldir. 4 Temel aşamadan(fazdan) oluşur. Bunlar aşağıdaki görselde ve açıklamalarda verildiği gibidir.



PLANLAMA: Her aşamada ara ürün için bir amaç belirlenir, planlama yapılır. Oluşturulan ara ürün bir önceki adımda ara ürün üretilmişse onunla birleştirilir.

RİSK ANALİZİ: Olası risklerin belirlenmesi, analiz edilmesi ve çözümlenmesi aşamasıdır.

ÜRETİM: Ara ürünün veya ürünün üretilmesi aşamasıdır

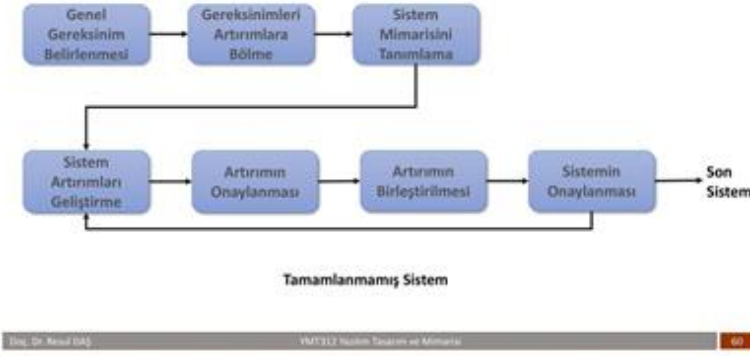
KULLANICI DEĞERLENDİRMESİ: Oluşturulmuş olan ara ürünlerin veya ürünün sonucunda müşterilerden/kullanıcılardan geri dönüt alınması ve ardından bu geri dönütlerin değerlendirilme aşamasıdır. Değerlendirme ardından diğer adıma geçilir.

Her yönüyle iyi bir model gibi gözükse de dezavantajları da vardır:

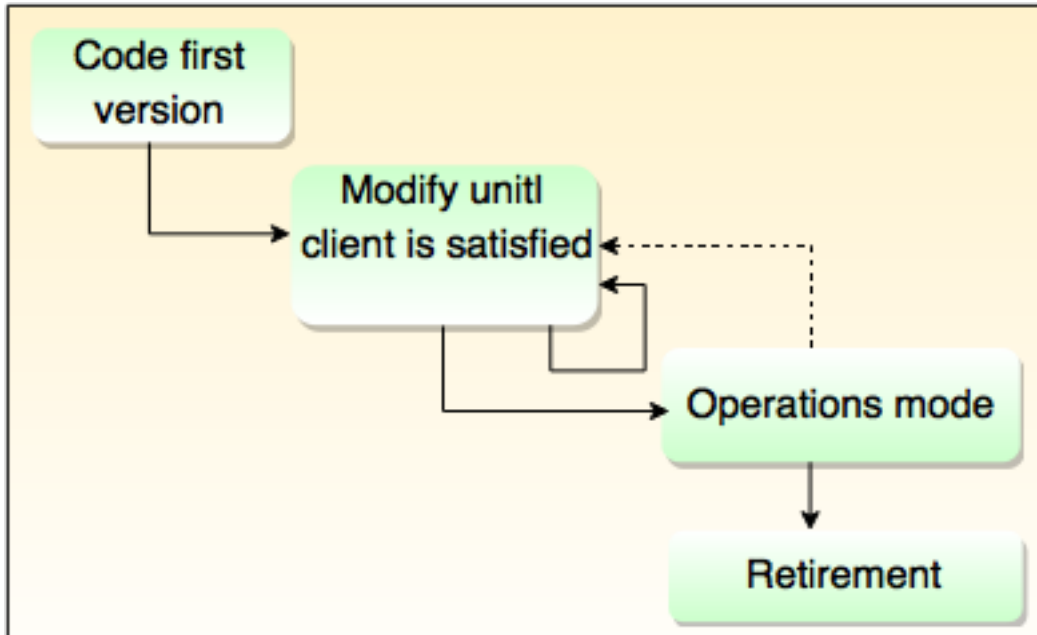
- Çok büyük olmayan ve düşük bütçe gerektiren projeler için fazla masraflıdır.
- Karmaşık bir sistemdir.
- Ara adımlar fazla olduğundan dolayı çok fazla dokümantasyon gerektirir bu da projenin uzamasına neden olmaktadır.
- Yalnızca büyük ölçekli projeler için kullanılması gerekmektedir.

ARTIRIMSAL GELİŞTİRME SÜRECİ MODELİ: Her bir adımda gittikçe işlevi arttırmaya, geliştirmeye, modernleştirmeye dayalı bir modeldir. Belli bir takvime, programa bağlı olarak gelişmektedir. Bundan dolayı eğer müşteriler/kullanıcılar ürünlerinde değişikliğe, gelişime ihtiyaç duyarsa bu modeli tercih etmeleri iyi bir seçim olacaktır. Kısıtlı sayıda çalışanla iş yapılır. Çünkü iş bölümlere ayrılmıştır ve adım adım ilerlenir. Ürün tüm geliştirmelerin sonunda ortaya çıkar. Doğal olarak yinelenmeli bir modeldir. Uzun zaman alan ve tüm işlevselliğini barındırmadığı halde çalışabilen sistemler bunun için uygundur.

Artırımsal Geliştirme Modeli



KODLA VE DÜZELT YAŞAM-DÖNGÜ MODELİ: Kısa ömürlü küçük projeler için kullanılabilen çok kolay fakat yüksek maliyetli bir modeldir. Maliyetli olmasının sebebi değişim yapmanın çok zor olması, esnek bir model olmaması. Bakımı vardır fakat bakımını yapmak çok zordur. Fazla tecrübe gerektirmedikinden küçük firmaların tercihi olabilir.



SCRUM NEDİR VE NEDEN GÜNÜMÜZDE POPÜLER: Scrum bir Agile Metodudur. Öncelikle Agile metodunun ne olduğundan bahsedelim. Agile metodu, kullanıcıları projeye daha fazla dahil ederek masrafları, giderleri kısan ve değişikliklere hızlı cevap verilebilmesini sağlayan bir methodur. Scrum da bir agile yöntemidir. Popüler olma sebebiyse kullanıcılarla projenin iç içe olması ve çok az hata ve masraf hedeflemesidir.

SCRUM PRENSİPLERİ:

SEFFAFLIK: Ekiptekiler birbirlerinin ne yaptığını bilip gördüğü bir ortamda çalışılmalıdır. Ekip üyeleri

DENETLEME: Projenin ilerleyişinin düzenli bir şekilde kontrol edilmesi, denetlenmesidir.

ADAPTASYON: Proje, yapılan tüm değişikliklere ayak uydurabilecek seviyede olmalıdır.

SCRUM YAPISI VE KAVRAMLARI

PRODUCT BACKLOG: Basit tanımıyla, proje için gerekenleri barındıran bir listedir. Üretilmek istenen şeyin ne olduğuna dair sorulan soruyu cevaplar. Product owner tarafından müşteriden gereksinimler toparlanılıp alınır ve önemine göre sıralanır. Ekleme veya çıkarmalar yapılabilir.

PRODUCT BACKLOG ITEM: Proje için gerekli olanların listesindeki her bir maddeye verilen isimdir.

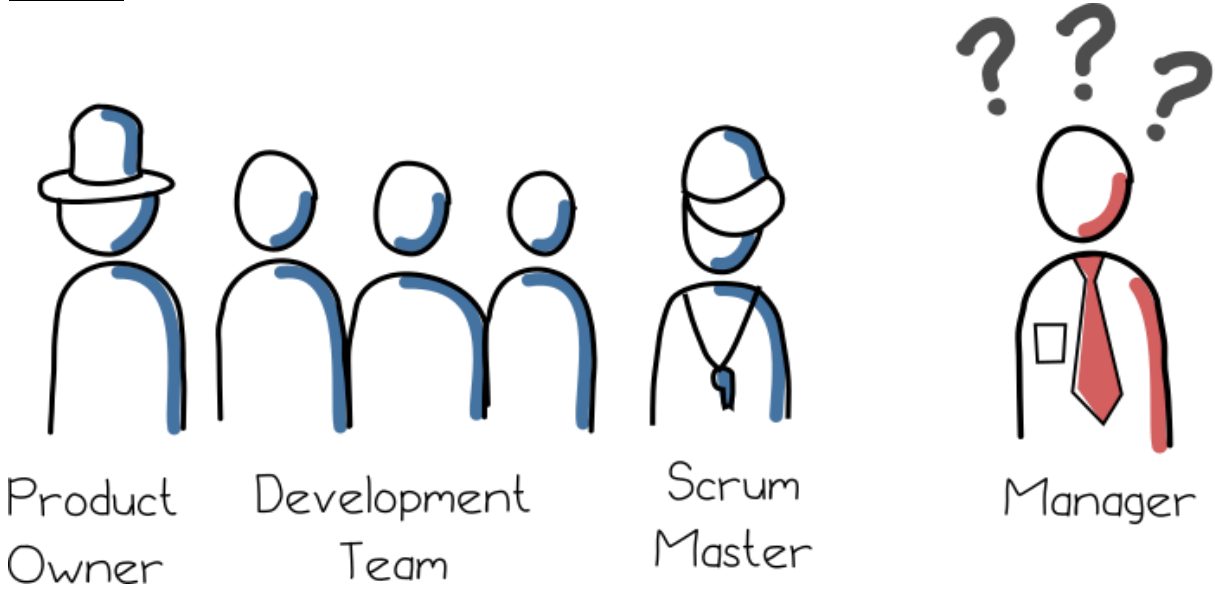
SPRINT: Proje bu sprint dediğimiz parçalara ayrılır. Tüm aktiviteler burada gerçekleşir.

SPRINT BACKLOG: İşlerin hangi sırada ve zamanlarda yapılacağı hakkında detaylı zaman çizelgesi oluşturulması.

SCRUM BOARD: Yapılacak işler birnevi sınıflandırmaya sokulur. Bu işler; TO DO(yapılacak olanlar), IN PROGRESS(yapım aşamasında olanlar), TO VERIFY(yayına hazır olanlar). DONE (tamamıyla bitirilenler). Şeklinde sınıflandırılır.

BURNDOWN CHART: Günlere göre yapılan ve kalan işleri gösteren bir grafikdir.

ROLLER



SCRUM TAKIMI: Fikirleri, düşünceleri sonuca bağlayan ve ürüne dönüştüren kişiler topluluğuna, ekibine denir.

PRODUCT OWNER (ÜRÜN SAHİBİ): Geliştirme takımıyla müşteri/kullanıcı arasındaki iletişimi canlı tutar.

Müşteriden aldığı geri dönüşleri geliştirme takımına ileterek ne gibi geliştirmeler, değişiklikler yapılması gerektiğini bildirmelidir. İsminden tek başına kararlar alan bir rol gibi görünse de öyle değildir. Müşteri ve takım arasında bir köprü görevi üstlenir.

DEVELOPMENT TEAM (GELİŞTİRME TAKIMI): Ürünü oluşturan kişilerin bulunduğu takımdır. Müşteriden gelen dönüşler üzerine hareket ederler. Bundan dolayı bir yöneticiye (manager) ihtiyaçları yoktur. Ürün hakkındaki bilgiler tüm ekip tarafından bilinir.

SCRUM MASTER: Bu rolü üstlenecek kişiler rolün adından da anlaşılacağı üzere scrum kurallarını çok iyi bilen kişilerden olmalıdırlar. Ve bu bilgileri, kuralları takım üzerinde uygulamakla sorumlu kişidir. Fakat yine isminden yanlış anlayacağımız gibi bir yönetici değildir.

SCRUM TOPLANTILARI

SPRINT PLANLAMA: Sprint, planlama toplantısı yapıldıktan sonra başlar. Bu toplantılar scrum master, ürün sahibi ve geliştirme ekibinin katılımıyla gerçekleşir. Product backlog'un bir kısmı geliştirme ekibiyle paylaşılır. Projeye dair bir plan çizilir.

DAILY SCRUM MEETING (GÜNLÜK SCRUM TOPLANTISI): Geliştirme Takımının ve Scrum Master'ın zorunlu Ürün sahibinin isteğe bağlı katılımıyla birlikte gerçekleşen günlük toplantılardır. Takım arkadaşlarının birbiriyle yaptıkları iş hakkında günlük olarak bilgi paylaştığı bir ortamdır.

SPRINT REVIEW (SPRINT DEĞERLENDİRME): Scrum Master, Ürün sahibi ve geliştirme takımının katılımıyla birlikte ürünün özelliklerinin tanıtılması, hatta reklamının yapılmasıdır.

SPRINT RETROSPECTİVE: Geliştirme takımı, ürün sahibi ve scrum masterın katılımıyla yapılan toplantılardır. Neler daha iyi yapılabilir, neler düzeltilmeli, neler iyi yapıldı gibi konular hakkında konuşulur.

KAYNAKÇA

<https://osmanozaydin.com/yazilim-yasam-dongusu-ve-agile-yazilim-gelistirme/>

<https://medium.com/@omerharuncetin/yazilim-yasam-dongu-modelleri-543c7879a742>

<https://www.codex.com.tr/yazilim-gelistirme-modelleri>

<https://tr.linkedin.com/pulse/yazilim-yasam-dongusu-nedir-veysel-ugur-kizmaz>

<https://www.mshowto.org/agile-ve-scrum-nedir.html>

<https://medium.com/@secilcor/scrump-nedir-6a4326951dd8>

HESAPLAR

Linkedin: <https://www.linkedin.com/in/aydin-eralp-193790218/>

Medium: <https://medium.com/@veyselaydinalp/-2c0e875e992f>

Github: <https://github.com/aydinalp>