PRATİKTE BT VE BS UYGULAMALARI 1. ÖDEV

<u>SORU1:</u> Proje yönetim süreçlerinde kullanılan metodolojileri inceleyiniz, örnekleyiniz, karşılaştırınız ve yorumlayınız.

Metodoloji, bir BT projesi ya da yazılım yaşam döngüsü aşamaları boyunca kullanılacak ve birbirleriyle uyumlu yöntemler bütünüdür.

Bir metodoloji,

- Bir süreç modeli,
- Belirli sayıda belirtim yöntemini içerir.

Günümüzdeki metodolojiler genelde Çağlayan ya da Helezonik modeli temel almaktadır.

Bir Metodolojide Bulunması Gereken Temel Bileşenler

- ✓ Ayrıntılandırılmış bir süreç modeli,
- ✓ Ayrıntılı süreç tanımları,
- ✓ İyi tanımlı üretim yöntemleri,
- ✓ Süreçlerarası arayüz tanımları,
- ✓ Ayrıntılı girdi tanımları,
- ✓ Ayrıntılı çıktı tanımları,
- ✓ Proje yönetim modeli,
- ✓ Konfigürasyon yönetim modeli,
- ✓ Maliyet yönetim modeli,
- ✓ Kalite yönetim modeli,
- ✓ Risk yönetim modeli,
- ✓ Değişiklik yönetim modeli,
- ✓ Kullanıcı arayüz ve ilişki modeli,
- ✓ Standartlar

Metodoloji bileşenleri ile ilgili bağımsız kuruluşlar (IEEE, ISO, vs.) ve kişiler tarafından geliştirilmiş çeşitli standartlar ve rehberler mevcuttur.

Kullanılan süreç modelleri ve belirtim yöntemleri zaman içinde değiştiği için standart ve rehberler de sürekli güncellenmektedir.

Bir kuruluşun kendi metodolojisini geliştirmesi oldukça kapsamlı, zaman alıcı ve uzmanlık gerektiren bir faaliyet olup, istatistikler yaklaşık 50 kişi/aylık bir iş gücü gerektirdiğini göstermektedir.

Yazılımın hem üretim hem de kullanım süreci boyunca geçirdiği tüm aşamalar yazılım geliştirme yaşam döngüsü olarak tanımlanır.

Yazılım işlevleri ile ilgili gereksinimler sürekli olarak değiştiği ve genişlediği için söz konusu aşamalar sürekli bir döngü biçiminde ele alınır.

Döngü içerisinde herhangi bir aşamada geriye dönmek ve tekrar ilerlemek söz konusudur.

Yazılım yaşam döngüsü tek yönlü ve doğrusal değildir.

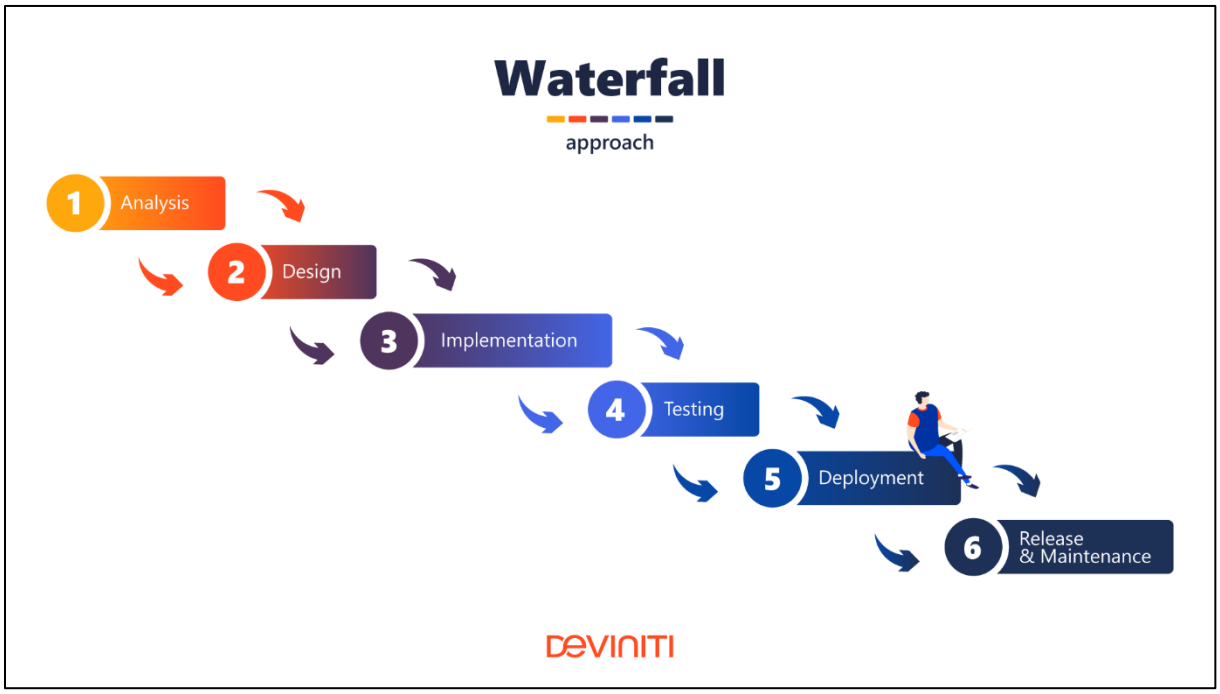
Yazılım Yaşam Döngüsü Temel Adımları

Yaşam döngüsünün temel adımları çekirdek süreçler (core processes) olarak da adlandırılır. Bu süreçlerin gerçekleştirilmesi amacıyla:

- **Belirtim (specification) Yöntemleri**: Bir çekirdek sürece ilişkin fonksiyonları yerine getirmek amacıyla kullanılan yöntemlerdir.
- **Süreç (process) Modelleri**: Yazılım yaşam döngüsünde belirtilen süreçlerin geliştirme aşamasında, hangi düzen ya da sırada, nasıl uygulanacağını tanımlayan modeller kullanılır.

Günümüzün proje yönetimi dünyasında ileri görüşlü yöneticiler ve liderler tek bir metodolojiye bağlı kalmazlar, birçoğunda bilgili hale gelirler ve projenin gerektirdiği her şeye uyum sağlamak için çeşitli uygulamaları nasıl birleştireceklerini öğrenirler.

1-Şelale Metodolojisi (Waterfall)



İlk olarak Dr. Winston Royce tarafından 1970 yılında, yazılım geliştirmenin giderek karmaşıklaşan doğasını yönetmeye bir yanıt olarak özetlenmiştir. O zamandan beri, en belirgin şekilde yazılım endüstrisinde yaygın olarak benimsenmiştir.

Waterfall metodolojisi sıralıdır. Aynı zamanda yoğun bir şekilde ihtiyaç odaklıdır. Daha fazla ilerlemeden önce projenin ne talep ettiği konusunda kristal netlikte bir fikre sahip olmanız gerekir. Proje başladıktan sonra düzeltme kapsamı yoktur.

Waterfall yöntemi ayrı aşamalardan oluşmaktadır. Gereklilikleri toplayıp analiz ederek, çözümü ve yaklaşımınızı tasarlayarak, çözümü uygulayarak ve varsa sorunlar çözülerek başlanır.

Bu süreçteki her aşama bağımsızdır; başka bir aşamaya geçmeden önce bulunduğunuz aşamayı tamamlarsınız.

Avantajlar

- Bu modelin anlaşılması ve kullanılması kolaydır. Aşamalar arasındaki ayrım sezgiseldir ve önceki deneyimlerden bağımsız olarak anlaşılması kolaydır.
- Waterfall metodolojisinin sertliği bir sorumluluktur. Aşamalar arasındaki net sınır, işi organize etmeye ve bölmeye yardımcı olur. Geriye dönemeyeceğiniz için her aşamada "mükemmel" olmanız gerekir ve bu yaklaşım genellikle daha iyi sonuçlar verir.
- Gereksinimlerin toplanması ve anlaşılmasına yönelik keskin odak, Waterfall modelini büyük ölçüde belgelere bağımlı hale getirir. Bu, yeni kaynakların taşınmasını ve gerektiğinde proje üzerinde çalışmasını kolaylaştırır.

Dezavantajlar

- Gerçek yaşamdaki projeler genelde yineleme gerektirir.
- Genelde yazılımın kullanıcıya ulaşma zamanı uzundur.
- Gereksinim tanımlamaları çoğu kez net bir şekilde yapılamadığından dolayı, yanlışların düzeltilme ve eksiklerin giderilme maliyetleri yüksektir.
- Yazılım üretim ekipleri bir an önce program yazma, çalıştırma ve sonucu görme eğiliminde olduklarından, bu model ile yapılan üretimlerde ekip mutsuzlaşmakta ve kod yazma dışında kalan (ve iş yükünün %80'ini içeren) kesime önem vermemektedirler.
- Üst düzey yönetimlerin ürünü görme süresinin uzun oluşu, projenin bitmeyeceği ve sürekli gider merkezi haline geldiği düşüncesini yaygınlaştırmaktadır.

2.Çevik Metodoloji (Agile)

Yazılım geliştirme odaklı başka bir proje yönetim metodolojisi olan **Agile**, karmaşık projeleri yönetmek için **Waterfall** yönteminin başarısızlığına bir yanıt olarak ortaya çıktı.

Agile proje yönetim fikirleri uzun bir süredir yazılım endüstrisinde kullanılmasına rağmen, 2001 yılında birkaç BT temsilcisi **Agile Manifesto**sunu yayınladığında resmen ortaya çıktı.

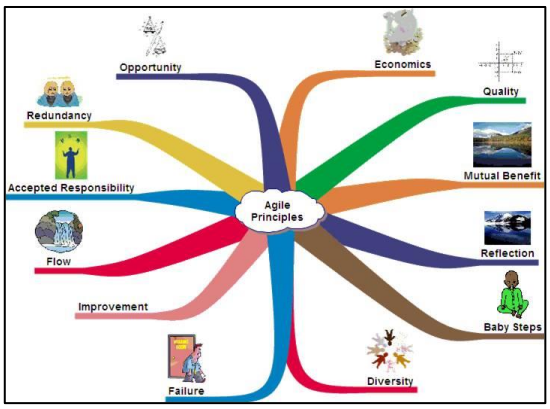
Yaklaşım ve ideolojide Agile, Waterfall yönteminin tam tersidir.

Adından da anlaşılacağı gibi, bu yöntem hızlı ve esnek bir yaklaşımı tercih eder. Üst düzey bir gereksinim toplama yoktur. Aksine, değişen gereksinimlere yanıt veren küçük artımlı değişikliklerle yinelemelidir.

Çevik yazılım geliştirme metodu, tekrarlanan yazılım geliştirme metodu taban alınarak geliştirilmiş, sık aralıklarla parça parça yazılım teslimatını ve değişikliği teşvik eden bir yazılım geliştirme metodolojisidir.

Çevik geliştirme;

- Değişimi,
- Takım içerisindeki iletişimin arttırılmasını,
- Parça parça yazılım teslimatını,
- Test odaklı yazılım geliştirilmesini,
- Uyumlu planlamayı teşvik eder.



Özellikleri

- Tanımlama (specification), tasarım (design) ve gerçekleme (implementation) süreçleri eş zamanlıdır. Detaylı tanımlama yoktur ve tasarım dokümantasyonu minimize edilmiştir.
- Sistem bir sıra ekleme şeklinde geliştirilir. Son kullanıcılar her artışı dener ve daha sonraki ekleme için teklifte bulunurlar.
- Sistem kullanıcı ara yüzü genellikle bir etkileşimli geliştirme sistemi kullanılarak geliştirilir.
- En yüksek iş değeri- en kısa süre
- Tekrarlı uygulama geliştirme
- Takım projelerinde çok başarılı
- Kapsamı dar, belirsizliği fazla olan projeler için kullanışlı,
- İhtiyaçların tam olarak belirlenemediği projelerde daha uygun,
- İş birimleri ve proje ekibinin esnek ilişkide olması ön şart
- Proje yönetimi ile yürütme bir arada

Çevik Yazılım Geliştirme Manifestosu

2001 yılında, dünyanın önde gelen çevik yazılım geliştiricileri (XP, Scrum gibi metodolojilerin yaratıcıları) ortak bir zeminde buluşabilmek adına bir araya gelerek **çevik yazılım geliştirme manifestosunu** ve çevik yazılımın prensiplerini yayınlamışlardır. Böylelikle çevik metotların projelere genel bakış açıları ifade edilmiştir.

Bu manifestoda;

Bireyler ve aralarındaki etkileşimlerin, kullanılan araç ve süreçlerden;

Çalışan yazılımın, detaylı dokümantasyondan;

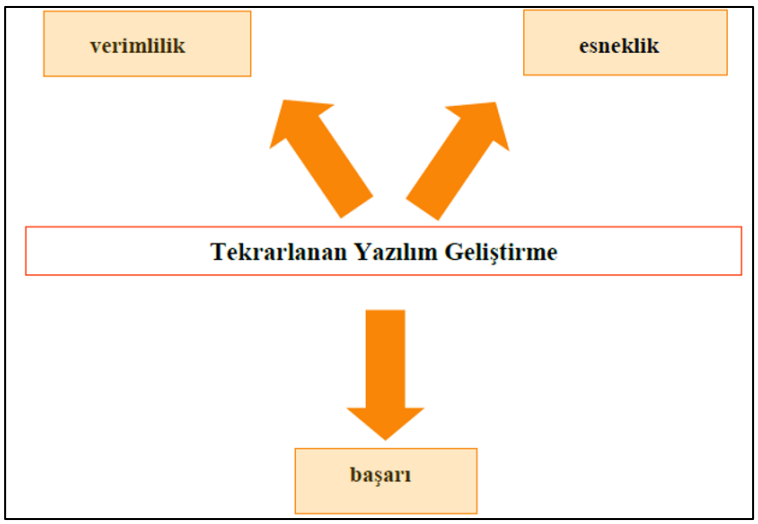
Müşteri ile işbirliğinin, sözleşmedeki kesin kurallardan;

Değişikliklere uyum sağlayabilmenin, mevcut planı takip etmekten; daha önemli ve öncelikli olduğu belirtilmektedir.

Avantajlar

- Sabit aşamalar veya gereksinimlere odaklanma olmadığından, kaynaklarınıza deneme yapma ve aşamalı değişiklikler yapma konusunda çok daha fazla özgürlük verir. Bu da özellikle yaratıcı projeler için uygun olmasını sağlar.
- Agile yönetim ile paydaşlardan düzenli geri bildirim alır ve buna göre değişiklikler yaparsınız. Bu, paydaşların her adımda yer alması nedeniyle proje başarısızlığı riskini büyük ölçüde azaltır.

- **Müşteri servislerinin hızlandırılmış teslimatı** - Her artış müşteriye en öncelikli işlevselliği gönderir.
- **Sistemle kullanıcı bağlantısı** - Kullanıcılar geliştirme sürecinin içine dahil edilmelidir, zira sistem daha büyük bir olasılıkla onların ihtiyaçlarını karşılar ve kullanıcılar daha fazla sisteme adanır.



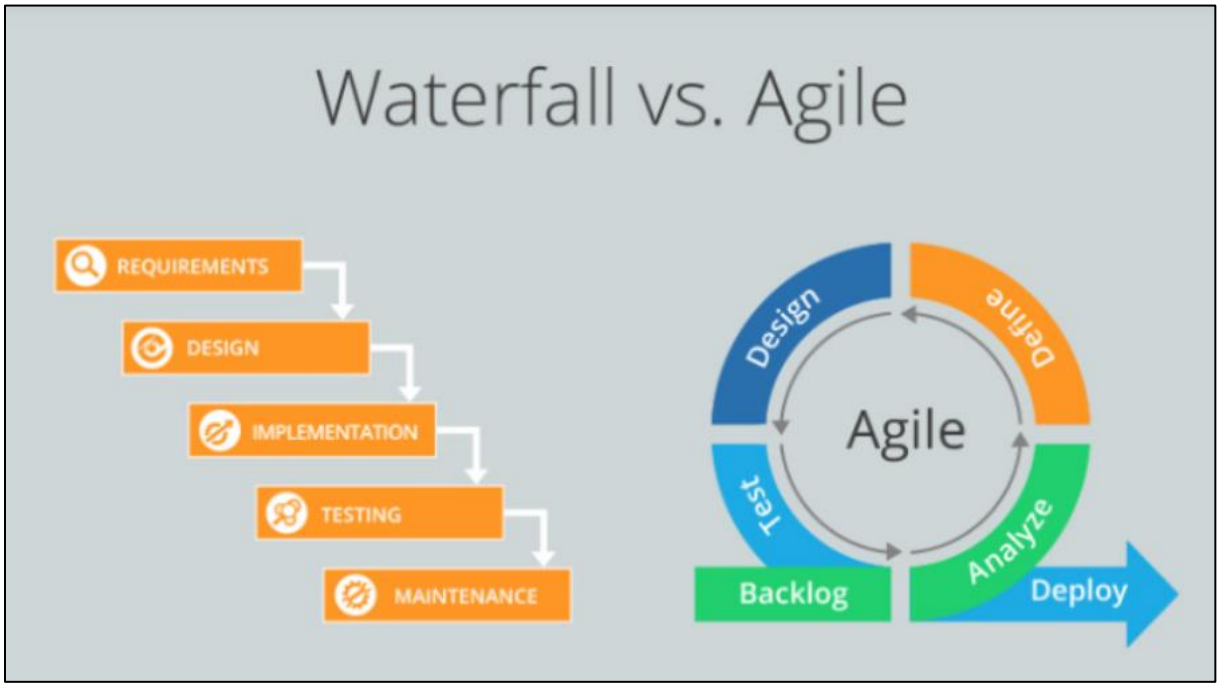
Dezavantajlar

- Sürece dahil edilen müşterilerin ilgisini sürekli kılmak zor olabilir.
- Takım üyeleri çevik metotları tanımlayan yoğun karışmaya uygun olmayabilirler.
- Önceliklerin değişimi birden fazla stakeholder(ortak) olması durumunda zordur
- Sadeliği koruma fazladan iş gerektirir.
- Tekrarlanan geliştirmeye farklı yaklaşımlar olduğunda sözleşme bir problem olabilir.

Şelale (Waterfall) ve Çevik (Agile) Karşılaştırma

Şelale: Her aşama birbirini takip ediyor ve geri dönüş yok. Müşteri ile el sıkışıldıktan sonra proje bitene kadar müşteri ile görüşme olmuyor. Kapalı bir kutu gibidir.

Agile: Aşamalar iteratiftir, gerektiğinde geri dönüş var. En yüksek iş değeri en kısa sürede. Takım projelerinde çok başarılı. Daha küçük parçalar ve otonom takımlar mevcut.



Şelale modeli ile çevik yöntemler aynı amaç doğrultusunda ortaya çıkmış yaklaşımlardır. Her iki yaklaşımda da planlama önemli bir yer tutar. Şelale modelinde bütün süreci kapsayan uzun vadeli ve detaylı bir planlama yapılırken, çevik yöntemlerde değişen şartlara cevap verebilmek için uzun vadeli planlamadan ziyade kısa vadeli ve daha detaylı planlamalar yapılır. Bütün süreç sürüm planlarına, sürüm planları yineleme planlarına, yineleme planları ise günlük planlara bölünür. Böylece daha detaylı hedefler belirlenebilir. Her iki yaklaşımda da süreç içinde detaylı incelemelere önem verilir. Şelale modelinde bu incelemeler safhalar arası geçişte yapılan detaylı kontroller ve dokümantasyon vasıtasıyla yapılırken, çevik yöntemlerde yinelemeler sonucu üretilen çalışan ürünler ve müşterinin bütün sürecin içinde bulunması ile detaylı bir inceleme sağlanmış olur. İki metodolojide de yapılan faaliyetler ortaktır. Bu faaliyetlerin icra şekilleri farklıdır. Şelale modelinde bu faaliyetler her gereksinime uygulanan safhalar olarak karşımıza çıkar ve ana sürecin alt bileşenleridir. Çevik yöntemlerde ise her yinelemenin uygulandığı faaliyetler olarak karşımıza çıkarlar ve ana sürecin alt bileşeni yinelemeleri oluşturan faaliyetlerdir.

3.Kanban Metodolojisi

Kanban, proje yönetimine görsel bir yaklaşımdır.

İş akışının ve ilerlemenin tüm katılımcılar için net olduğu bir Kanban panosuna görevler yerleştirerek iş akışını yönetmeye yardımcı olur. Kanban, verimsizliklerin iyileştirilmesine yardımcı olur ve Agile projelerinde yalın üretimi programlamak için kullanılmıştır.

Çağımızda Trello gibi yazılımda görsel planlama panolarının ortaya çıkmasıyla birlikte artık **Kanban araçları** ve **Kanban yöntemleri** için yeni kullanımlar vardır.

Kanban İlkeleri

Kanban, ekip yapınızda herhangi bir değişiklik yapmadan süreçleri ve iş akışı verimliliğini iyileştirmek için kullanılabilir.

Kanban metodunu uygulamadan önce, temel ilkelerini anlamak ve benimsemek önemlidir:

- **Ne yapıyorsanız oradan başlayın.** Kanban, belirli bir kurulum gerektirmez ve doğrudan mevcut iş akışınıza uygulanabilir. Bu, mevcut süreçlerinizi değiştirmenize gerek olmadığı için uygulamayı kolaylaştırır.
- **Değişimi kabul etmelisiniz.** Kapsamlı değişiklikler ekipleri rahatsız edebilir, akışı bozabilir ve performansa zarar verebilir. Kanban, sürekli, artan ve evrimsel değişiklikleri teşvik ederek minimum dirençle karşılaşacak şekilde tasarlanmıştır.

- **Mevcut sürece, rollere ve sorumluluklara saygı gösterin.** Başlangıçta herhangi bir organizasyon değişikliği olmamalıdır. Kanban, mevcut süreçlerin, rollerin ve sorumlulukların değeri olabileceğini ve korunmaya değer olduğunu kabul etmelisiniz.
- **Her düzeyde liderliği destekleyin.** Kanban, tüm üyeler arasında liderliği ve karar almayı destekler. En alt sıradaki ekip üyesinin parlak bir fikri varsa, bu kabul edilmeli ve benimsenmelidir. Çalışanlarınızın optimum performansa ulaşması için herkesin sürekli iyileştirme (**Kaizen**) zihniyetini teşvik etmesi gerekir.

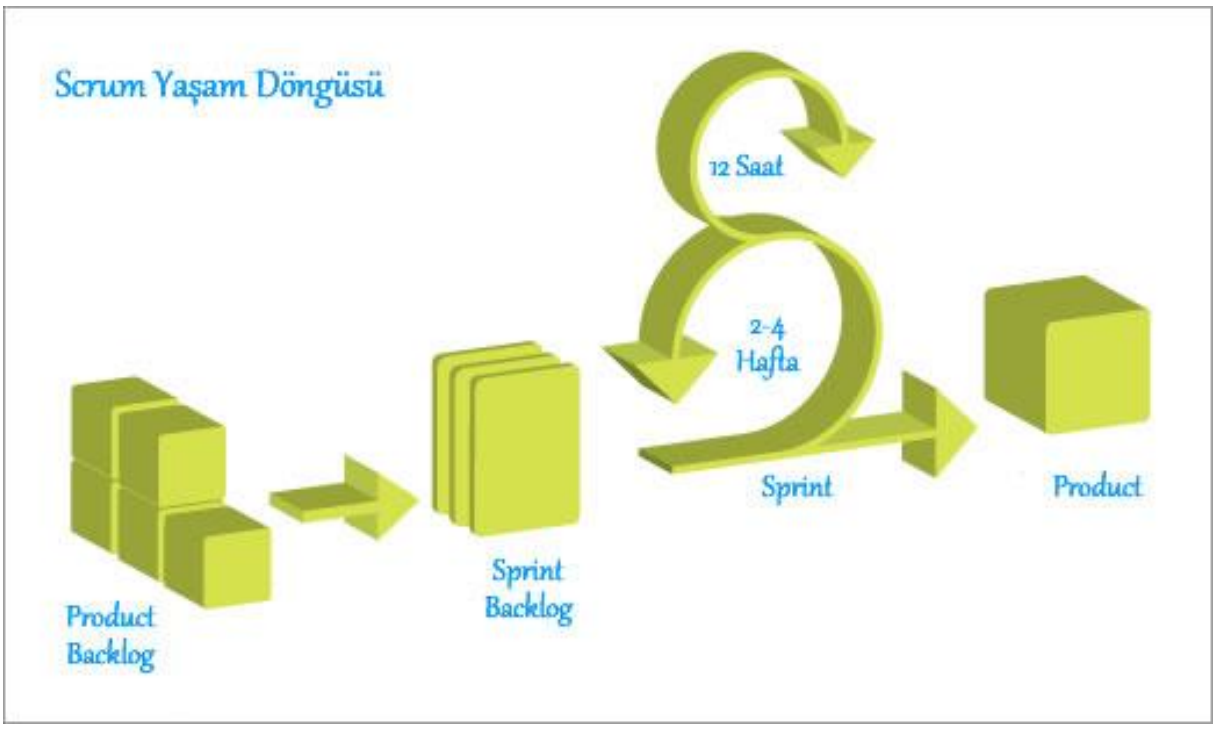
Avantajlar

- Kanban sistemi, şirketin üretim sistemlerinde sürekli ve sürdürülebilir iyileştirmeleri savunur. Kanban, yalnızca manuel yönergelerden veya kartlardan oluşmaz, aynı zamanda işin gözden geçirilmesini kolaylaştıran süreç çıktılarının görselleştirmelerinden oluşur. Bu ayrıca, daha fazla dikkat gerektiren diğer potansiyel sorunlu alanları da vurgulayabilir.
- Kanban sistemi çok duyarlı bir sistemdir ve herhangi bir gecikme veya gecikmeyi teşvik etmez. Görevler sürekli olarak kanban kartlarının sütunları arasında kaydırıldığı için, kaynakları diğerinden kaydırarak ve değiştirerek mümkün olan en kısa sürede yanıtlanabilecek genel çıktıyı tutabilecek sınırlayıcı faktörlerin ortaya çıktığı alanları otomatik olarak vurgular. görevler.

Dezavantajlar

- Görevler sürekli olarak Kanban panosunun sütunları arasında kaydırıldıkça, görevlerin veya etkinliklerin tamamlanması için belirli zaman çizelgelerinin tahmin edilmesi zorlaşır. Bunun nedeni, Kanban'ın yalnızca çekme üretim sisteminde bir sinyal bağlantı noktası olarak işlev görmesidir.
- Bir sistemde çok fazla faaliyet veya görev birbiriyle ilişkiliyse, Kanban'ı uygulamak çok zor hale gelecektir. Bunun nedeni, bu tür sistemlerin, farklı görevler arasında çok sık mal ve uzmanlık transferi olasılığını artırması ve tüm bu faaliyetlerin hızını korumadaki zorluğu artırmasıdır.

4. Scrum (Çevik Metodoloji Örneği)



Scrum, tam donanımlı bir proje yönetimi metodolojisi değildir. Bunun yerine proje ekiplerine, kısa sprintlere ve günlük stand-up toplantılarına odaklanarak Agile yönetim yaklaşımını açıklar.

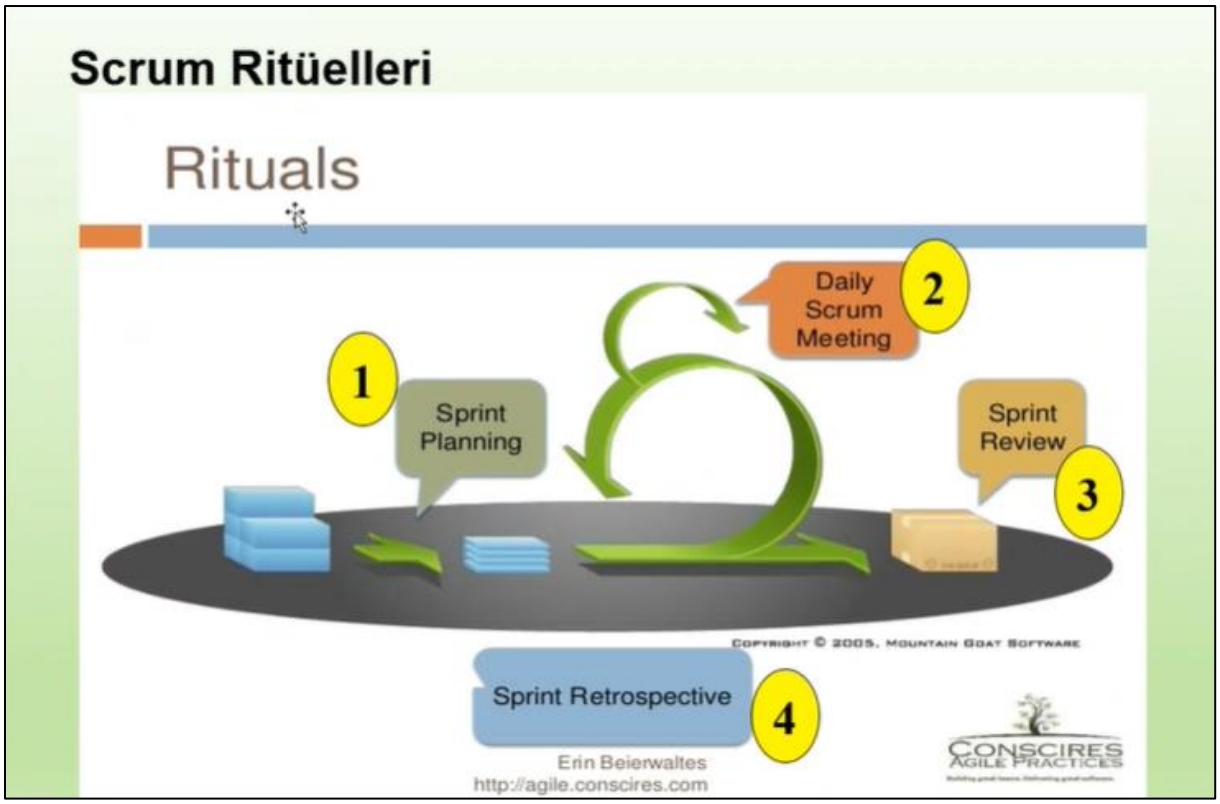
Agile'dan ilkeleri ve süreçleri ödünç alırken, Scrum'ın proje yönetimi ile uğraşmak için kendine özgü yöntemleri ve taktikleri vardır.

Modern yazılım projelerinin oldukça karmaşık olduğu ve en baştan tümünü planlamanın zor olacağı varsayımdan hareket eder. Bu karmaşıklığı 3 ilke ile azaltmaya çalışır:

- Şeffaflık
- Denetleme
- Uyarlama

Temel Özellikleri:

- Gözlemci
- Geliştirmeci
- Tekrara dayalı



Avantajlar

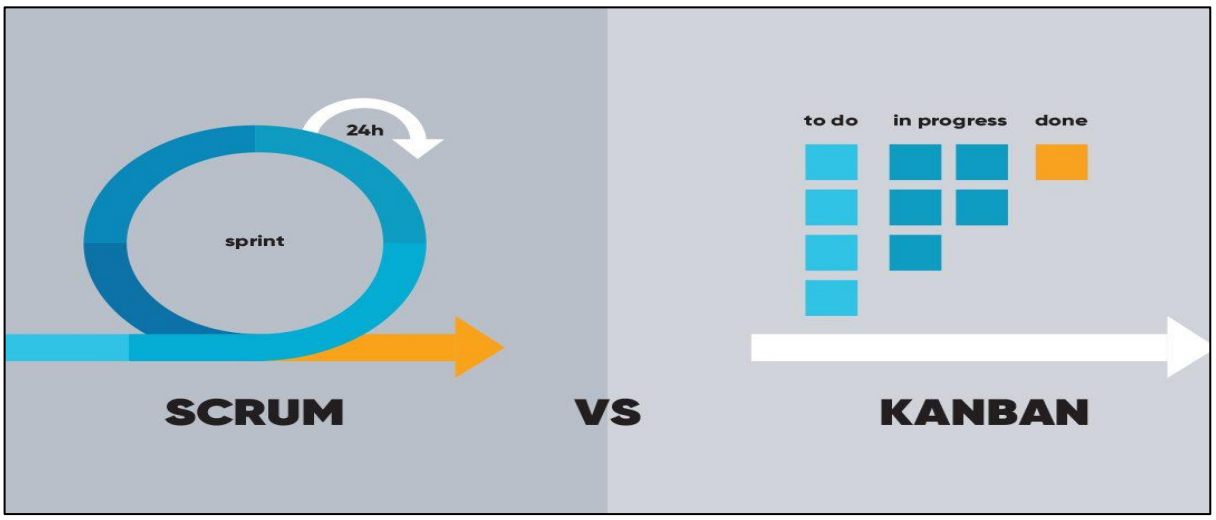
- **Scrum** yaklaşımı ağırlıklı olarak 30 günlük Sprint'lere odaklanır. Burası proje ekibinin nihai hedeflerin istek listesini küçük parçalara ayırdığı ve ardından günlük stand-up toplantılarıyla 30 günlük oturumlarda bunlar üzerinde çalıştığı yerdir. Bu, büyük ve karmaşık projeleri yönetmeyi kolaylaştırır.
- 30 günlük limiti ve günlük stand-up toplantılarıyla Sprint yaklaşımı, hızlı yinelemeyi ve gelişimi destekler.
- Proje ekibinin kendi kendini yönetmesi beklendiğinden, Scrum takımları projeyi net bir şekilde görebilir. Bu aynı zamanda proje liderlerinin yeteneklerine ilişkin kendi bilgilerine göre kendi önceliklerini belirleyebilecekleri anlamına gelir.

Bunlara ek olarak Scrum, Agile'ın tüm avantajlarına sahiptir: Hızlı yineleme ve düzenli paydaş geri bildirimi gibi.

Dezavantajlar

- Sabit bir bitiş tarihi veya zamanlama ve bütçeleme için bir proje yöneticisi olmadığından, Scrum kolayca kapsam sürünmesine neden olabilir.
- Proje ekibi kendi kendini yönettiği için, ekip yüksek oranda disiplinli ve motive olmazsa daha yüksek başarısızlık riski vardır. Takım yeterli deneyime sahip değilse, Scrum'ın başarısız olma ihtimali çok yüksektir.
- Proje ekibi odağı, ekibi arada bırakan herhangi bir kaynağın net sonuçları büyük ölçüde etkileyeceği anlamına gelir. Bu yaklaşım aynı zamanda büyük ekipler için yeterince esnek değildir.

Kanban ve Scrum Karşılaştırma

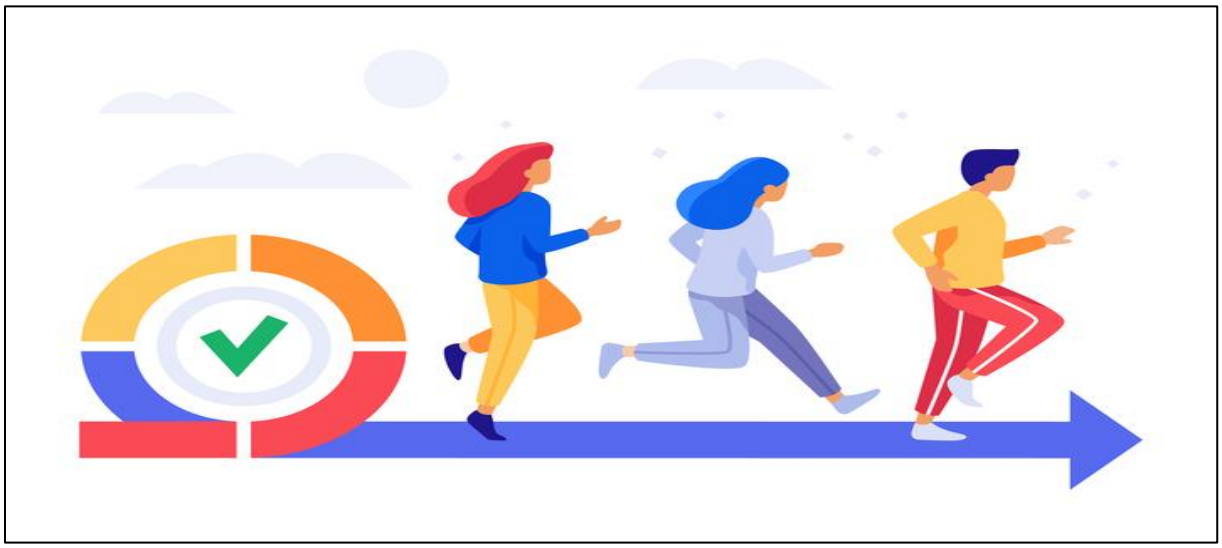


Hem **Scrum** hem de **Kanban**, sunumu iyileştirmek için hızlı tempolu, verimlilik temelli yaklaşımlar sunar. Her iki yöntem de büyük, karmaşık projeleri yönetilebilir parçalara ayırmanıza ve iş akışlarını tüm ekibi döngüde tutacak şekilde görselleştirmenize olanak tanır.

Bununla birlikte, uygulama açısından her bir yöntem temelde diğerinden farklıdır. Bu nedenle birçok takım, Kanban'ı mı yoksa Scrum'ı mı uygulayacağına karar vermeden önce dikkatlice analiz eder. Doğru ya da yanlış karar yoktur, hsize uygun olan hangisi ise onu kullanmanız gerekir.

Kanban Yöntemi, ani yapısal değişiklikler ve öngörülen törenler olmaksızın uygulanması en basit yöntemlerden biridir. İş akışınızı sürekli olarak analiz ettiğiniz ve yönettiğiniz sürece, Kanban olağanüstü sonuçlar sağlayabilir.

5.Lean Metodoloji(Yalınlık)



Lean metodolojisi, verimlilik temasına odaklanan bir proje yönetimi metodolojisidir.

Değeri tanımlayarak başlar ve ardından değer akışını optimize ederek ve israfı ortadan kaldırarak sürekli iyileştirme yoluyla verimliliği maksimuma çıkarır.

Lean metodolojisi, proje teslim sürecinizi gözden geçirirken benimsemek için yararlı bir zihniyet olabilir. Proje sürecinize değer sağlayan ve tüyler ürpertici olan şeyleri ortadan kaldıran çıplak temellere nasıl geri çekebileceğinizi Lean metodolojisi ile kolayca bulabilirsiniz.

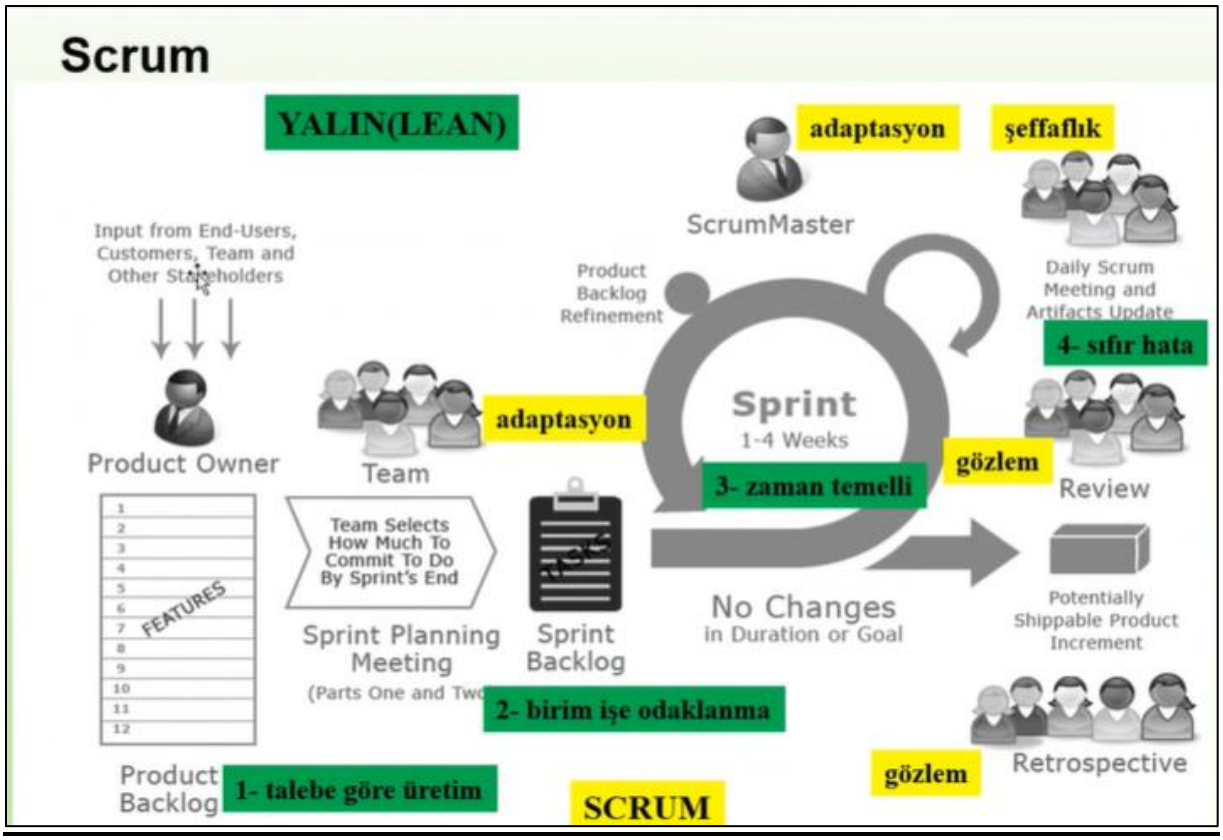
Avantajlar

- **Lean yönetim**, kârı en üst düzeye çıkarmakla ilgilidir. Satış fiyatı, ürün kalitelerine veya pazarlara bağlı olabilecek çeşitli faktörlerden etkilenirken, genellikle şirketler maliyetlerini kontrol etmek için daha fazlasını yapabilir ve Lean uygulamalar, tüm tasarrufların kara eklenebilmesi için maliyetleri düşürmeye yardımcı olur.
- Lean yönetim, müşterinin bakış açısını dikkate alır. Personelle iletişim kurma biçimleri, endişelerine yanıt verme ve ürünle ilgili deneyimleri, savurgan uygulamaları azaltmanın önde gelen nedenlerinden bazılarıdır.

Dezavantajlar

- Geleneksel olarak, Lean yönetim uygulamasında, taşıma maliyetlerini düşürmek için düşük miktarlarda stok tutulur.
- Çalışanlar her zaman Lean yönetim uygulamalarına sıcak bakmayabilir. Bu tarzın uygulanması, çok fazla sabır ve iş süreçlerinde tam bir revizyon gerektirecektir. Bu da uzun süreli çalışanların rahat edeceği bir şey olmayabilir.

- Çoğu zaman daha önce hiç Lean yönetim kullanmamış bir şirkette Lean yönetim uygulandığında, tüm sistemlerin ve üretim süreçlerinin mevcut durumunda sona ermesi muhtemeldir.



6.Critical Path Method (CPM)

Yukarıdaki dört proje yönetimi yöntemi, yazılım geliştirmeden ortaya çıktı. Bunları kesinlikle yazılım dışı projeler için kullanabilseniz de, seçenekleriniz arasında daha iyi alternatifler vardır.

Critical Path Method'da, projeyi bir iş kırılım yapısı içinde tamamlamak için gereken tüm etkinlikleri kategorilere ayırırsınız. Ardından, her bir faaliyetin tahmini süresini ve bunlar arasındaki bağımlılıkları eşlersiniz.

Bu, eşzamanlı olarak tamamlanabilecek etkinlikleri ve diğerlerinin başlayabilmesi için hangi etkinliklerin tamamlanması gerektiğini belirlemenize yardımcı olur.

Avantajlar

- Faaliyetlerin süresinin ve karşılıklı bağımlılıklarının haritalanmasına yapılan vurgu, görevleri daha iyi planlamanıza yardımcı olur. X görevi, önce bitirilecek Y görevine bağlıysa, CPM bunu belirlemenize ve planlamanıza yardımcı olacaktır.
- CPM metodolojisinin başarısı, kritik ve kritik olmayan faaliyetlerin tanımlanmasına ve haritalanmasına bağlıdır. Bu etkinlikleri haritalandırdıktan sonra, kaynaklara daha iyi öncelik verebilirsiniz.

Dezavantajlar

- Deneyimli bir proje yöneticisinin size söyleyeceği gibi, işler her zaman beklediğinizden daha fazla zaman alır. Planlama konusunda gerçek dünya deneyiminiz yoksa, her aktivite için zamanı yanlış hesaplamak zorunda kalırsınız.
- Waterfall yöntemi gibi, CPM de başlangıçta zordur. Her şeyi en baştan planlamanız gerekir. Herhangi bir değişiklik varsa, tüm programı geçersiz kılar. Bu, bu yöntemi değişen gereksinimleri olan projeler için uygunsuz hale getirir

7.Critical Chain Project Management (CCPM)

Critical Chain Proje Yönetimi, piyasadaki en yeni proje yönetimi metodolojilerinden biridir. Kaynak yönetimi odaklı Critical Path Method'a alternatif olarak geliştirilmiştir.

CCPM ile, nihai hedeften geriye doğru çalışırsınız. Teslim edilebilirleri tanırsınız, ardından projeyi tamamlamak için gereken görevleri haritalamak için geçmiş deneyimi kullanırsınız. Ayrıca kaynaklar arasındaki karşılıklı bağımlılıkları da planlar ve bunları her göreve uygun şekilde tahsis edersiniz.

CCPM, kaynak kullanımını vurgular ve üretkenlik kaybını en aza indirir. Büyük ölçüde "tek göreve", yani eldeki göreve odaklanmaya ve çoklu görevden kaçınmaya dayanır.

Kaynak sıkıntısı çeken proje ekipleri için CCPM güçlü bir metodoloji olabilir.

Avantajlar

- Uygun kaynak yönetimine tüm odaklanma, CCPM'yi kaynak açısından en verimli proje yönetimi metodolojilerinden biri yapar. Tek göreve yapılan vurgu, aynı zamanda çoklu görevin zararlı etkilerine dair modern anlayışımızla da uyumludur.
- CCPM, bir soruna "optimum" çözüm üzerinde takıntılı değildir. Bunun yerine, nihai hedefe ulaşılmasına yardımcı olabilecek "yeterince iyi" çözümlere öncelik verir. Son hedeften geriye doğru da çalıştığınız için, CCPM genellikle karmaşık projeler için daha iyi sonuçlar verir.

Dezavantajlar

- CCPM'nin kaynak odaklı yaklaşımı yalnızca tek proje ortamlarında çalışabilir. Çok projeli ortamlarda, projeler kaynakları paylaşabilir. CCPM, böyle bir senaryoda kaynak dağıtımını planlayamaz.

- CCPM, bir görev süresi uzunluğunu türetmek için görevler arasında bir boşluk veya dolgu sağlar. Teoride, bunun kendi verimliliklerini fazla hesaplayan kaynakları telafi etmesi gerekiyor. Gerçekte, Parkinson Yasası uyarınca kaynaklar, dolguyu aşırı gecikmelerle doldurur.

Sonuç Olarak;

Bir proje yöneticisi için aralarından seçim yapabileceği birkaç proje yönetimi metodolojisi mevcuttur. Bu metodolojilerin her birinin kendi güçlü ve zayıf yönleri vardır. Doğru olanı seçmek, projenin daha hızlı, daha sorunsuz ve daha verimli yürütülmesini sağlayacaktır.

Kaynakça:

https://acodez.in/12-best-software-development-methodologies-pros-cons/

https://members.tripod.com/war_project/projeler/proje1.html