

YAZILIM MÜHENDİSLİĞİ TEMELLERİ

HAZIRLAYAN: UMUT MADANOĞLU 220601002

ÖĞRETİM ÜYESİ: Dr. Öğr. Üyesi Zekeriya Anıl GÜVEN

İÇİNDEKİLER

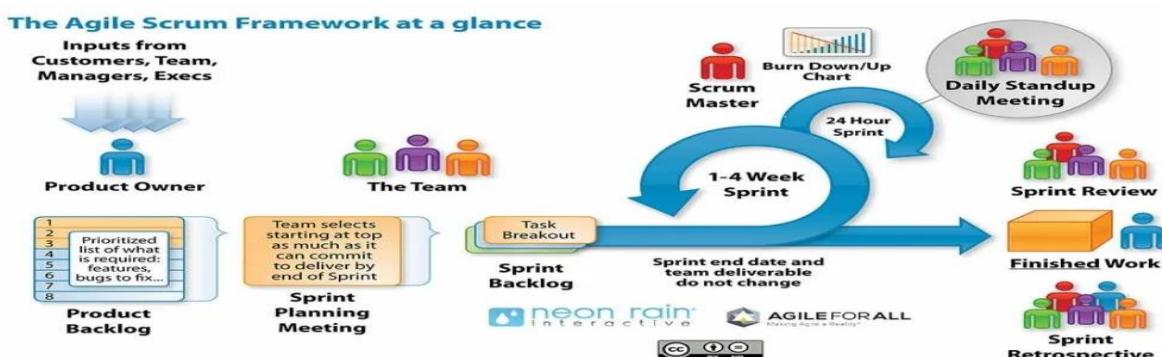
- SAYFA 1 Kapak ve İçindekiler
- SAYFA 2 Yazılım Yaşam Döngüsü
- SAYFA 2 SCRUM
- SAYFA 3 SCRUM'UN DİĞER MODELLERE GÖRE FARKI
- SAYFA 4 SCRUM NEDEN POPÜLER
- SAYFA 4 HANGİ PROJELERDE SCRUM KULLANILIR
- SAYFA 4 V SÜREÇ MODELİ
- SAYFA 5 V MODELİ DİĞER MODELLERE GÖRE FARKI
- SAYFA 5 HANGİ PROJELERDE V MODELİ KULLANILIR
- SAYFA 5 ÇAĞLAYAN YAŞAM DÖNGÜ MODELİ
- SAYFA 6 ÇAĞLAYAN YAŞAM DÖNGÜ MODELİ DİĞER MODELLERE GÖRE FARKI
- SAYFA 6 HANGİ PROJELERDE ÇAĞLAYAN YAŞAM DÖNGÜ MODELİ KULLANILIR
- SAYFA 6 SPİRAL MODEL
- SAYFA 7 SPİRAL MODELİN DİĞER MODELLERDEN FARKI
- SAYFA 7 HANGİ PROJELERDE SPİRAL MODEL KULLANILIR
- SAYFA 7 ARTIRIMSAL GELİŞTİRME SÜREÇ MODELİ
- SAYFA 7 ARTIRIMSAL MODELİN DİĞER MODELLERE GÖRE FARKI
- SAYFA 7 HANGİ PROJELERDE ARTIRIMSAL MODEL KULLANILIR
- SAYFA 8 KAYNAKÇA

YAZILIM YAŞAM DÖNGÜSÜ

Yazılım yaşam döngüsü yazılım geliştirme sürecinin ortak bir şekilde anlaşılması sağlayıcı bir dizi aşamadır. Bu iş fikir ve gereksinimlerini işlev ve özelliklere dönüştürmek için iş anlayışı ve gereksinimlerin ortaya çıkarılması aşamasından, tamamlanan uygulamanın bakımına kadar bir bilgi sistemi geliştirme projesinde yer alan aşamaları tanımlayan, proje yönetiminde kullanılan manevi bir modeldir. Yazılım geliştirme aşamasında takip edilen, belirlenen ve tasarlanan farklı yazılım geliştirme yaşam döngüsü modelleri vardır. Her süreç modeli, yazılım geliştirme adımda başarısı sağlamak için kendi türüne özgü bir dizi aşamayı takip eder.

SCRUM

Scrum, ekiplerin çalışmalarını bir dizi değer, ilke ve uygulama aracılığıyla yapılandırmasına ve yönetmesine yardımcı olan çevik bir proje yönetimi çerçevesidir. Scrum takımları deneyimler yoluyla öğrenmeye, bir problem üzerinde çalışırken kendi kendini organize etmeye ve sürekli gelişmek için kazandıkları ve kaybettikleri üzerine düşünmeye teşvik eder. Yani Scrum sadece yazılım geliştirmede değil her alanda uygulanabilir. Ekiplerin ürünleri ve hizmetleri yinelemeli olarak piyasaya sürmesine olanak tanırken aynı anda bunları sürekli ortaya çıkan sorunları çözmek için uyarlar. Ürün geliştirmeye yönelik Scrum yaklaşımı, sonuçları asla hafife almaz ve ürün ekiplerini müşterilere hızlı ve sürekli olarak yüksek kaliteli ürünler sunma arayışlarında destekler. Scrum da büyük projeler parçalara bölünerek ve her bir parçaya da "sprint" adı verilerek sonrasında her bir sprintin tek bir geliştirildiği bir proje yönetimidir. Bu metodoloji, karmaşık ortamlarda adım adım yazılım geliştiren ekipler için uygundur. Bu karmaşıklığı üç ilke ile azaltmaya çalışır bunlar: Şefaflık, denetleme ve uyarlamadır. Tüm unsurları birbirine bağlayan kritik bir Scrum Takımı özelliği güvendir. Bir Scrum Takımında Güven yoksa, işlerin yapılması yolunda büyük olasılıkla gerilm olacaktır. Scrum Değerleri, çalışma şeklinize rehberlik etmeye ve güveni artırmaya yardımcı olduklarıdan, Scrum Takımlarının bağlı kalması için de kritik öneme sahiptir. Scrum'un çalışma mantığına bakacak olursak, müşteri tarafından istenilen tanımlanan işlevler dört haftalık "Sprint" adı verilen dönemler içerisinde geliştirilir. Her sprint sonunda yazılımın foksiyonel parçası bitmiş ve müşteriye teslim edilebilir durumda olur ve günlük toplantılar yapılır. Bu iterasyonun tamamlanması 30 günden fazla sürmez.



Scrum'da üç rol vardır. Bunlar Scrum master, ürün sahibi ve scrum takımıdır.

- Scrum Master: Scrum, master ekibi kurabilir, toplantıyı ayarlayabilir ve süreç için engelleri kaldırabilir.
- Ürün Sahibi: Ürün sahibi, ürün biriktirme listesini oluşturur, gecikmeye öncelik verir ve her tekrarda işlevselligin dağıtilmasından sorumludur.
- Scrum Takımı: Takım işini yönetir ve sprinti veya döngüyü tamamlamak için işi organize eder.

Scrum'da üç temel kavram vardır. Bunlar roller, toplantılar ve bileşenlerdir.

- Roller: Müşteri tarafından görevlendirilen projede geri dönüş açısından çok önemlidir. Scrumda proje yöneticisi rolü bulunmaktadır, sebebi deneyimsiz bir scrum ekibinde sorunlar çıkabileceğidir. Takım kendi içinde üç parçaya ayrılır. Yazılımcı, testçi ve tasarımcıdır.
- Toplantılar: Scrum'un en önemli olan kısmı toplantılardır. Geliştiriciler ve yöneticiler arasında toplantı yapılır ve bu toplantılarında geliştiricilerin önceki günlerde neler yaptıkları, geliştirme süreci esnasında karşılaşlıklarını sorunlar ve bugün yazılım üzerinde neler yapmaları gerektiği hakkında bir planlama yaparlar. Sprintlere bakılarak ne kadar ilerleme kaydedildiğine de bakılır.
- Bileşenler: Ürünün Gereksinimler dokümanı oluşturulur. Dokümana proje süresince neler yapılması gerektiği basit bir şekilde kaydedilir. Sprint için de doküman oluşturulur. Sprint dokümanı ve ürün dokümanının ayrı ayrı oluşturulmasının amacı her sprintin ürün gereksinimler dokümanına uygun ayarlanmasıdır. Sprint kalan zaman grafiği bulunmaktadır ve bu grafik yapılmış olan işlemlerin ne seviyede bulunduğu ve planlanan zamanın neresinde olunduğunu belirlemek için hazırlanır.

SCRUM'UN DİĞER MODELLERE GÖRE AVANTAJ VE DEZAVANTAJLARI

Scrum diğer yazılım modellerine göre daha esnektir, bunun sebebi sürekli olarak müşteri geri bildirimlerine dayalı olarak değişen bir ortamda çalışır ve bu nedenle projelerin gereksinimlerine ve ihtiyaçlarına hızla yanıt verir. Scrum daha az belge kullanma avantajı sunar, Scrum ekip üyeleri arasındaki işbirliği ve iletişim teşvik eder, daha küçük parçalarda kodlayıp test ederek, sürekli geri bildirim vererek ve hataları düzeltmek için çok pahalı hale gelmeden önce düzeltilebilir. Günlük toplantıları sayesinde tüm takım kimin ne yaptığıını bilir, birçok yanlış anlaşılma ve karışıklığı ortadan kaldırır. Bunlar Scrum modelinin diğer modellere göre avantajlarıdır.

Scrum diğer modellere göre belirli bir bitiş tarihi olmaması nedeniyle paydaşlara ek işlevsellik talep etmek için cazip gelebilir. Scrum Master ekibi kontrol etmeye çalışırsa, proje başarısız olur. Görevler iyi tanımlanmamışsa proje maliyetleri ve zaman çizelgeleri kesinlikle doğru olmaz. Başlangıçtaki hedefler net değilse, planlama zorlaşıır ve sprintler başlangıçta tahmin edilenden daha fazla zaman alabilir. Scrum Takımı'nda tanımlanmış bir rol olmadığından (herkes her şeyi yapar), teknik deneyime sahip takım üyelerine ihtiyaç duyar. Ekip ayrıca günlük Scrum toplantılarına da bağlı kalmak ve proje süresince takımada kalmak zorundadırlar. Bunlar Scrum modelinin diğer modellerle karşılaştırdığımızda dezavantajları olarak sayılabilir.

SCRUM GÜNÜMÜZDE NEDEN POPÜLER

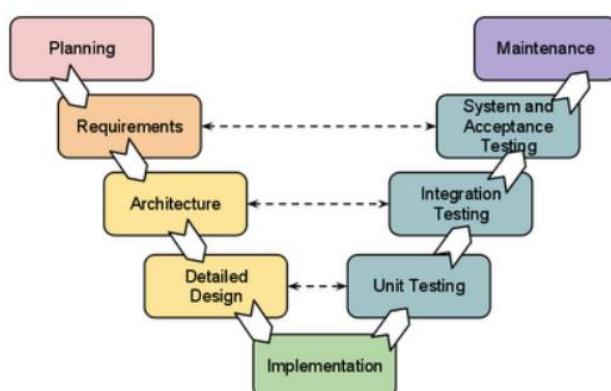
Ekiplerin değişen gereksinimlere yanıt vermesi ve müşterilere hızlı bir şekilde değer sunmasını sağladığı için yazılım geliştirmede popüler hale geldi. Scrum, en popüler Çevik metodolojilerden biridir çevikliğe yönelik artan talepte Scrum'un popüler olmasını sağlamıştır. Ekipleri çalışmalarını sık sık test etmeye ve gözden geçirmeye teşvik ederek kaliteyi artırır. Ekipler, geliştirme süreci boyunca kaliteye odaklanarak sorunları erkenden tespit edip düzelterek kusur riskini azaltabilir ve ürünün genel kalitesini iyileştirebilir. Genel olarak, SCRUM'un popülaritesi, ekiplerin daha verimli ve etkili çalışmasına, işbirliğini ve iletişimini teşvik etmesine ve müşterilerin ve paydaşların ihtiyaçlarını karşılayan yüksek kaliteli ürünler sunmasına yardımcı olma becerisinden kaynaklanmaktadır.

HANGİ PROJELERDE SCRUM KULLANILABİLİR

Scrum, özellikle yazılım geliştirme sürecinde sıkça kullanılan bir çevik proje yönetimi metodolojisidir. Scrum, takım odaklı, tekrarlanabilir, ölçeklenebilir ve esnek bir yapıya sahip olduğu için, birçok yazılım geliştirme projesinde, özellikle kompleks ve değişken projelerde başarılı sonuçlar almak için kullanılabilir. Bir uygulamanın veya web sitesinin tasarılanması ve geliştirilmesi, bir yazılım altyapısının oluşturulması gibi projeler için ve IT projelerinde Scrum kullanılabilir.

V SÜREÇ MODELİ

Model V harfini andırırken sol tarafı üretimi, sağ tarafı ise sınama işlemleri ile ilgili aşamalardan oluşmaktadır. Lineer bir şekilde aşağı doğru hareket etmek yerine uygulama ve kodlama aşamasından sonra süreç adımları yukarı doğru büükülerek tipik V şeklini oluşturur. V şeklindeki model ile şelale modeli arasındaki en büyük fark, V şeklindeki modelde erken test planlamasıdır. Model testinde ve geliştirmesinde, adım paralel olarak planlanır. Dolayısıyla, doğrulama aşamaları vardır. V-Model, Kodlama aşamasında birleşir. Her bir üretim aşamasının karşısında bulunan sınama aşamaları sayesinde hata kaynaklarına dönüş daha kolaydır. V- modelinde aşamalarda tekrar bulunmaz ve risk çözümleme için ayrılan bir yer yoktur. V modeli döngüsü esas olarak 3 modelden oluşmaktadır. Bunlar kullanıcı modeli, mimari model ve gerçekleştirim modelidir. Belirsizliklerin az, iş tanımlarının belirgin olduğu projeler için uygun bir modeldir.



V modelin kullanılabilmesi için müşterinin ne istedğini bilmek ve projenin gelişimini size bırakması gerekmektedir, projenin kodlama aşamasına kadar prototip üretilmeyeceğinden bu oldukça önemlidir. Projede bir değişiklik söz konusu olduğunda test dökümanlarının güncellenmesi gerekmektedir. Bakım, onarım ve projeden vazgeçmek V modelinin içeriğinde değildir. V modelde son ürüne ulaşmak doğrusal bir yolla ifade edilemez. Her aşama bir sonraki aşama başlamadan önce tamamlanmalıdır

V MODELİN DİĞER MODELLERE GÖRE AVANTAJ VE DEZAVANTAJLARI

Kullanımı ve takibi kolay olması, onaylama ve doğrulama işlemlerinin planın erken aşamasında uygulanabilir olması, hatalar erken aşamada bulunur ve bir sonraki aşamaya hataların geçmesi önlenir, planlama ve test tasarımı gibi test faaliyetleri kodlamadan önce gerçekleştirildiği için proje içerisinde çok zaman kazandırır bu nedenle diğer modellere göre daha avantajlıdır.

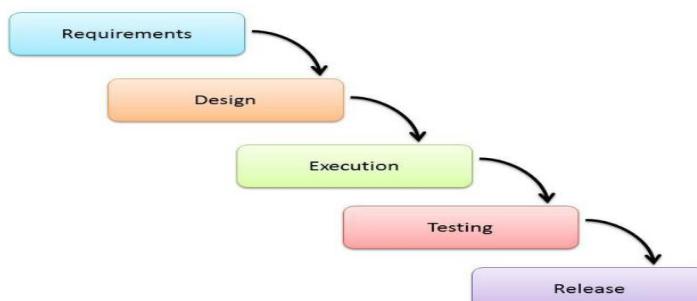
Yazılım geliştirme aşamasında geliştirilir bu nedenle yazılımın erken prototipleri üretilemez, herhangi bir aşamada gereksinimler üzerinde değişiklik olursa test belgelerinin de diğer belgeler ile güncellenmesi gerekir, uygulama şekli oldukça katı ve kesin kurallara bağlı olması diğer modellere göre dezavantajlarından sayılabilir.

HANGİ PROJELERDE V MODELİ KULLANILABİLİR

Gereksinimleri açık ve belli olan küçük, orta ve büyük boyut projelerde, belirsiz veya tanimsız bir gereklilik yok ise, kullanılan teknoloji dinamik değilse, ürün tanımı sabit projelerde kullanılabilir. e-devlet uygulamaları ve finansal yazılımlarda kullanılabilir

ÇAĞLAYAN YAŞAM DÖNGÜ MODELİ

Çağlayan modeli doğrusal sıralı bir akıştır. Geliştirmenin gereksinim analizi, tasarım, uygulama, test, entegrasyon ve bakım adımları aracılığıyla sürekli olarak aşağı doğru aktığının görüldüğü sürekli bir yazılım geliştirme modelidir. Faaliyetlerin doğrusal sıralamasının bazı önemli sonuçları vardır. Birincisi, bir aşamanın sonunu ve bir sonrakini başlatıcını belirlemek için, her adımın sonunda bazı sertifikasyon tekniklerinin kullanılması gerekir. Bazı doğrulama ve geçerli kılma genellikle bunu, aşamanın çıkışının girdisiyle (önceki adımın çıkışısı olan) tutarlı olmasını ve aşamanın çıkışının sistemin genel gereksinimleriyle tutarlı olmasını sağlayacak anlamına gelir. Gereksinimdeki değişiklikleri işlemek için önceki aşamaya geri dönme süreci tanımlanmaz.



Bu, geliştirme sürecindeki herhangi bir aşamanın yalnızca önceki aşama tamamlandığında başlayacağı anlamına gelir. Çağlayan modelinde yazılım aşamalarının en az birer kez tekrarlanması gereklidir.

ÇAĞLAYAN MODELİN DİĞER MODELLERE GÖRE AVANTAJ VE DEZAVANTAJLARI

Her aşamada doğrulama yapılması hataların erken tespit edilmesini sağlar. Projeyi planlamaya ve programlamaya yardımcı olur. Müşteriler için anlaşılır adımlardan oluşur, bu yüzden müşterilere açıklaması kolaydır. Proje yöneticileri için iş dağılımı yapmak kolaydır. Bu sayılanlar diğer yazılım modellerine göre çağlayan modelinin avantajlarıdır.

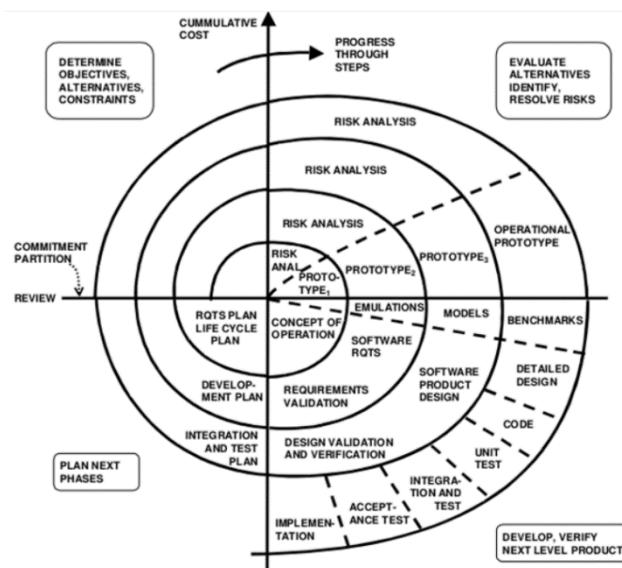
Yazılımın müşteriye ulaşma zamanı uzundur. Yanlışların ve eksiklerin düzeltilme maliyeti yüksektir. Tasarım ve test aşamalarına önem verilmemektedir kod yazma kısmı daha ön planda çıkmaktadır. Proje bittikten sonra herhangi bir aşamaya geri dönmek oldukça zor çağlayan modelinin diğer yazılım modellerine göre dezavantajlarıdır.

HANGİ PROJELERDE ÇAĞLAYAN MODEL KULLANILABİLİR

Çağlayan yazılım modeli, dil anlama ve doğal dil işleme konularında oldukça başarılı bir modeldir ve birçok farklı projede kullanılabilir. Projelere örnek verecek olursak: Dil Çeviri uygulaması çağlayan modeli, birden fazla dilde çeviri yapmak için kullanılabilir. Özellikle çeviri kalitesi açısından oldukça başarılıdır. Otomatik metin düzenleme uygulaması bu uygulama, uzun metinleri özetleyen ve anahtar noktaları belirleyen bir yazılımdır. Çağlayan modeli, otomatik metin özetleme konusunda oldukça başarılıdır.

SPİRAL MODEL

Spiral model risk odaklı bir süreç modelidir. Bu geliştirme modeli, prototipleme modelinin ve şelale modelinin özelliklerini birleştirir. Spiral modeli diğer modellerden ayırmakta olan bir özellik risk analizinin ön planda olması ve prototip oluşturulmasıdır. Risk analizi ön planda olması ortaya çıkan hatalar için erken giderilebilme imkânı sağlayabilir. Spiral teknik, tasarım ve geliştirme faaliyetlerinde hızlı prototipleme ve eşzamanlılığın bir kombinasyonudur. Planlama, risk analizi, üretim ve kullanıcı değerlendirme adında 4 ana aşamadan oluşup tekrarlanır. Tekrarlanırken yaptığı her bir tekrara faz denir. Bu fazlar esnasında risk analizlerinin ardından kullanıcıdan olumlu ya da olumsuz yanıt almak için, ürünün prototipi üretilip kullanıcıya sunulur. Her sunumun ardından alınan değerlendirmeler sonucu yeni geliştirilmiş planlarla yeni bir faza başlanır.



SPİRAL MODELİN DİĞER MODELLERE GÖRE AVANTAJ VE DEZAVANTAJLARI

Riskleri yönetmesi ve sistemi aşamalar halinde geliştirmesi, pek çok yazılım modelini içinde bulundurması, hataları erken giderilmeye çalışılması diğer yazılım modellerine göre avantajlarıdır. Riskleri değerlendirmek için özel ihtiyaçlara duyması, çok fazla dokümantasyona ihtiyaç duyması bu modelin diğer modellere göre dezavantajlarından sayılabilir.

HANGİ PROJELERDE SPİRAL MODEL KULLANILABİLİR

Spiral model, büyük, pahalı ve karmaşık projeler için tercih edilir. Spiral modeli, yazılım geliştirme projelerinde risk yönetimi ve kalite kontrolünün önemli olduğu durumlarda kullanılabilir. Havacılık yazılımı, sağlık yazılımı, bankacılık yazılımı, Telekomünikasyon yazılımı gibi karmaşık yazılım projelerinde Spiral model tercih edilebilir.

ARTIRIMSAL GELİŞTİRME SÜREÇ MODELİ

Bir yazılım geliştirme yaşam döngüsünün, başlangıçtaki basitleştirilmiş bir uygulamaya odaklanan ve daha sonra son sistem tamamlanana kadar aşamalı olarak daha fazla karmaşıklık ve daha geniş bir özellik seti kazanan özel bir uygulamasıdır. Sistemi tek seferde teslim etmek yerine parçalara böler. Bu yöntemin arkasındaki temel fikir, yazılım geliştiricilerin sistemin önceki bölümlerinin veya sürümlerinin geliştirilmesi sırasında öğrenilenlerden yararlanması izin vererek, tekrarlanan döngüler ve her seferinde daha küçük parçalar halinde bir sistem geliştirmektir.

ARTIRIMSAL MODELİN DİĞER MODELLERE GÖRE AVANTAJ VE DEZAVANTAJLARI

Sınırlı kaynakları daha iyi kullanmayı amaçlar, doğrusal yaklaşımlardan daha çok müşteriye odaklı olması, projede yapılacak değişiklikleri daha erken tespit edebilmesi, tüm projenin başarısız olma riskini azaltması diğer modellere göre avantajlarıdır. Ağır belgeler gerektirmesi işleri bölmek sorunlu olabilir, proje planlanması sırasında yinelemeler dikkate alınmazsa yinelemeler arasında entegrasyon sorunu olabilmesi diğer modellere göre dezavantajlarıdır.

HANGİ PROJELERDE ARTIRIMSAL MODEL KULLANILABİLİR

Artırımsal model, özellikle büyük ve karmaşık projelerde kullanılabilir. Örnek olarak: E-ticaret web siteleri müşteri deneyimini artırmak için birçok farklı özellik ekleyebilirsiniz. Online oyunlar, oyun mekanlığını geliştirmek ve yeni özellikler eklemek için sürekli olarak geliştirmeler yapabilirsiniz. Finansal yazılımlar, özellikle büyük ölçekli finansal kurumlar için karmaşık projelerdir. Artırımsal model, finansal yazılımlarda kullanarak, müşteri deneyimini iyileştirmek ve işlem verimliliğini artırmak için yeni özellikler ekleyebilirsiniz.

HESAPLARIM

<https://github.com/UmutMdn>

https://tr.linkedin.com/in/umut-madano%C4%9Flu-7454811b3?original_referer=https%3A%2F%2Fwww.google.com%2F

<https://medium.com/@madanoglumut>

KAYNAKÇA

<https://www.atlassian.com/agile/scrum>

<https://www.scrum.org/learning-series/what-is-scrum>

<https://www.yazilimtestmerkezi.com/post/yazilim-gelistirme-modeli-nedir-cesitleri-nelerdir>

<https://www.javatpoint.com/software-engineering-sdlc-models>

İzmir Bakırçay Üniversitesi Yazılım Mühendisliği Temelleri Dersi Dr. Öğr. Üyesi Zekeriya Anıl GÜVEN Slaytları