

Proje Yönetim Süreçleri: Kullanılan Metodolojiler

Onur Osman GÜLE¹

¹Bilgisayar Mühendisliği Bölümü, Sakarya Üniversitesi Bilgisayar ve Bilişim Bilimleri Fakültesi, Sakarya, Türkiye
G171210021
onur.gule@ogr.sakarya.edu.tr

Özet – Günümüzde yazılım projelerinde farklı farklı metodolojiler kullanılmaktadır. Bu metodolojiler öncelik olarak projeyi zaman ve bütçe konusunda rahatlatmak amacıyla da olsa metodolojilerin takımlar üzerindeki etkisi kaçınılmazdır. Farklı metodolojilerin ve bu metodolojilerin takımlar üzerindeki etkisi ve buna bağlı olarak takımın yazılım geliştirme sürecindeki etkileri değişebilmektedir. Bu etkiler olumlu ya da olumsuz olabilmektedir. Herhangi bir problem yaşamayan bir takımda tamamen olumlu etkiler göstermekte ve bu olumlu etkiler maddi ve manevi olarak zaman ve bütçeyi olumlu yönde ilerletmektedir. Temel etkenlerden birinde problem yaşayan projelerde ise takım ve proje yöneticisine bağlı olarak olumsuz etkiler göstermekte ve bu olumsuz etkiler ışığında maddi ve manevi olarak zaman ve bütçe olumsuz yönde ilerletmektedir. Bu çalışmada öncelikle temel metodolojiler incelenecek olup bu metodolojiler ile ilgili geçmişteki araştırmalar da incelenip bu metodolojilerin takım çalışmalarına etkileri temel alınarak projede oluşabilecek riskler ve projeyi olumlu veya olumsuz yönde ilerletebilecek etkenler ortaya konulmaktadır.

Anahtar Kelimeler — Şelale Modeli, Yinelemeli Şelale Modeli, Çevik Yöntemler, Kanban, Yalın Model, Scrum, Metodolojilerin Etkileri

Project Management: Methodologies

Summary – Today, different methodologies are used in software projects. Although these methodologies are primarily intended to ease the project in terms of time and budget, the impact of the methodologies on the teams is inevitable. The impact of different methodologies and these methodologies on the teams and accordingly the effects of the team on the software development process may change. These effects can be positive or negative. It shows completely positive effects in a team that does not have any problems, and these positive effects materially and spiritually advance the time and budget. In projects that have problems in one of the main factors, it shows negative effects depending on the team and the project manager, and in the light of these negative effects, it advances time and budget negatively materially and spiritually. In this study, first of all, basic methodologies will be examined, and the previous researches related to these methodologies are examined and the risks that may occur in the project based on the effects of these methodologies on team studies and the factors that can advance the project in a positive or negative way are revealed.

Keywords — Waterfall Model, Recursive Waterfall Model, Agile Methods, Kanban, Lean Model, Scrum, Effects of Methodologies



SAKARYA
ÜNİVERSİTESİ

Giriş

Günümüzde yazılım projelerinde kullanılan PMI standartlarına uygun metodolojilere bakıldığında işgücünün yoğun olduğu gözlemlenebilir. Yaygın olarak kullanılan metodolojiler arasında Waterfall(Şelale), Agile(Çevik), Kanban ve Scrum modelleri bulunmaktadır. Her ne kadar Waterfall modelinde yönetilen bir projenin kişisel olarak yürütülmesi mümkün olsa da çevik yöntemlerle yönetilen projelerin kişisel olarak yürütülmesi, projeyi teslim etme açısından bir hayli zor olacaktır. Scrum gibi çevik yöntemler, tamamlanması beklenen işler birbirinden ayrıldığı için takım çalışmasına daha uygun, yerine göre düşük maliyetli, düşük riskli ve yüksek verimli yöntemler olarak sınıflandırılır. Günümüzdeki girişim projelerinde ilk arananlardan biri nitelikli ve uyumlu takımlar olmaktadır. Her geçen gün zorlaşan rekabet ile tüm firmalarda verimliliğin artması ile projelerin performansının yükselmesi ihtiyacı doğmuştur. Bu durumun yarattığı arayışlar, birey odaklı projelerin takım odaklı projelere evrilmesine yol açmıştır. Her ne kadar grup çalışmalarının bazen düşük üretkenlik, zayıf karar mekanizmaları gibi olumsuz sonuçları ortaya çıkabilse de birçok güncel çalışmada takım çalışmasının eş zamanlı olarak üretkenliği ve çalışan memnuniyetini artırdığını ortaya koymuştur. Araştırmacılar takım çalışmasının daha yüksek verimlilik, daha düşük emek maliyeti getirdiğini ve bireysel performansı kesin bir şekilde artırdığını gözlemlemişlerdir.

Başarılı Projelerdeki Çalışmalar

Başarılı projelere Google, Facebook, Microsoft gibi örnekler verilebilir. Tüm bu şirketler bir ekip ruhu çerçevesi altında doğmuşlardır ve bu çerçevede hızla büyümeye devam etmektedir. Şu anda en başarılı, büyük şirketlerden biri olan Google'ı örnek verebiliriz takım çalışmasının başarılarına. Google, 2019 yılının son çeyreğinde çalışan sayısını 118.899 olarak açıklamıştır[1]. Google, yalnızca ana görevi olan arama motoru işlevi dahilinde birçok farklı projeye imza atmıştır. Çalışan sayıları da birçok projede çalışan, yönetici, personel olarak, takım olarak bu şekilde dallanmıştır. Google, 1996'da Amerika Birleşik Devletleri'ndeki Kaliforniya eyaletindeki Stanford Üniversitesi'nde doktora öğrencileri olan Larry Page ve Sergey Brin'in araştırma projesi olarak hayata geçirildi. Google şu anda kocaman bir şirket olduğundan projeleri yönetmen bir hayli zor olmakta. Proje yönetiminde tüm projelerde olmasa da bazılarında Çevik, Scrum metodolojilerini kullandıklarını açıklamışlardır.

Şirketlerin Projelerinde Agile Metodolojisi Kullanma Oranları[2]	
Paypal	56%
Spotify	52%
Adobe	39%
Apple	26%
Microsoft	25%
Cisco	25%
Twitter	24%
Oracle	23%
Coca Cola	13%
Google	10%
IBM	9%
Tesla	9%
Facebook	8%
Amazon	8%
SpaceX	5%
McDonald's	0%

Paypal, Spotify gibi daha yenilikçi şirketler Çevik metodolojisini kullanmakta çok iyi yerlerde olurken Apple, Microsoft, Twitter, Oracle gibi şirketler daha az kullanmakta. Google, Cisco, IBM gibi ürünleri eskiden beri var olan şirketler ise bu tür metodolojilere ayak uydurmakta zorlanmakta.

Proje Yönetim Metodolojilerinin Amacı

Proje belirli bir başlangıç ve bitiş tarihi olan, belirli bir amaç için (şirket hedefleri ve stratejileri doğrultusunda) gerçekleştirilen çalışmadır. Proje yönetiminin en önemli faydaları, şirket stratejilerine uyumlu çıktıların elde edilmesini sağlamak, projede kullanılacak kaynak (insan, yazılım, hammadde vs.) kullanımını optimize etmek, maliyet ve bütçe kontrolünü iyileştirmektir. Buna ek olarak proje yönetim teknikleri müşteri memnuniyeti, organizasyonel öğrenme ve etkin takım çalışmasını da geliştirir.

Proje yönetiminin en önemli aşaması başlatma evresidir, bu evre öncesinde şirket üst yönetiminin proje ile elde edilecek kazançlar hakkında bir fizibilite çalışması yapmış olması ve projenin paydaşları, bütçesi, zaman planı ve amacı konusunda ortak bir karar almış olması gerekir. Bu fizibilite çalışması ve ortak karar olmadan başlayan her proje sahipsiz kalmaya ve başarısız olmaya mahkumdur diyebiliriz. Projenin başlatılma kararının ardından “proje yaşam döngüsü” başlar. Öğrencilik yıllarımda birçok projede yer almakla birlikte bu proje yaşam döngüsünün gerçekten önemli bir konu olduğunu tecrübe etmiş bulunmaktayım. Takım büyüklüğü, proje kapsamı gibi konuların çerçevesinde uygun bir metodoloji seçilmesi gerekmektedir.

Şelale Modeli

Şelale yönteminde (yaygın kullanılan adı WaterFall Model) [yazılım](#) geliştirme süreci [analiz](#), tasarım, kodlama, test, sürüm ve bakım gibi safhalardan oluşur. Geleneksel yazılım metodlarında bu safhalar şelale modelinde olduğu gibi doğrusal olarak işler. Her safha, başlangıç noktasında bir önceki safhanın ürettiklerini bulur. Kendi bünyesindeki değişiklikler doğrultusunda teslim aldıklarını bir sonraki safhanın kullanabileceği şekilde değiştirir.

Şelalenin her basamağında yer alan aktiviteler eksiksiz olarak yerine getirilir. Bu bir sonraki basamağa geçmenin şartıdır. Her safhanın sonunda bir doküman oluşturulur. Bu yüzden şelale modeli doküman güdümlüdür. Yazılım süreci doğrusaldır, yani bir sonraki safhaya geçebilmek için bir önceki safhada yer alan aktivitelerin tamamlanmış olması gerekir. Kullanıcı katılımı başlangıç safhasında mümkündür. Kullanıcı gereksinimleri bu safhada tespit edilir ve detaylandırılır. Daha sonra gelen tasarım ve kodlama safhalarında müşteri ve kullanıcılar ile diyaloga girilmez.

Müşteri ne istediğini tam olarak bilmeyebilir. Bu yüzden proje öncesi detaylı analiz yapılması, müşterinin her gereksimini dile getirdiğinin garantisi değildir. Müşterinin bazı gereksinimlerini projenin ilerleyen safhalarında keşfetmesi durumunda, oluşan değişikliklerin projeye dahil edilmesi hemen hemen imkansız olacaktır. Bunun en büyük sebeplerinden birisi de yazılım için oluşturulan tasarımın projesi öncesi belirlenmesi ve bu yüzden ilerde meydana gelebilecek değişikliklerin göz önünde bulundurulmamış olmasıdır. Projenin ilerleyen safhalarında meydana gelen her değişiklik tasarımı zorlar. Tasarım çevik olmadığı için değişiklikleri taşıyamaz.

Şelale yöntemi ile müşterinin istediği yazılım sistemi proje sonunda tamamlanır. Ancak bu safhada müşteri yazılım sistemini test edebilir. Müşteri tamamlanan yazılım sistemini tüm artı ve eksileriyle kabullenmek ve kullanmak zorundadır[?1].

Agile(Çevik) Model

Agile Proje Yönetimi, sürekli test ve değişime açık, yinelemeli bir proje yönetimi yaklaşımıdır. Agile proje yönetimi yöntemleri yazılım geliştirmeye odaklanmıştır ve Agile Manifesto yazılım geliştiricileri tarafından oluşturulmuştur.

Agile manifestosunda yer alan 4 temel değer aşağıdaki şekildedir[?2].

- Kişiler ve etkileşimler, işlemler ve araçlardan daha önceliklidir.
- Yazılım üzerinde çalışmak, döküman üzerinde çalışmaktan önceliklidir.
- Müşteri katkısı, müşteri ile yapılan sözleşmelerden önceliklidir.
- Değişime verilen cevap, planlamadan daha önceliklidir.

Agile yöntemlerin en büyük avantajlarından biri, değişen öncelikleri yönetme yeteneğidir. Agile'nin yinelemeli yaklaşımı ve sürekli geri bildirimle önem vererek, ekibin yalnızca öngörülen koşulları değil, gerçek koşullara göre daha etkili seçimler yapmasına izin verir. Agile belirlenmiş kısa döngüleri, daha net proje görünürlüğü ve düzenli raporlama güncellemeleri ile, ekipler proje öngörülebilirliğini artırabilir ve riski azaltabilir. Müşteri memnuniyeti Agile Proje Yönetimi'nin 4 değerinden birisidir. Agile'nin en büyük değerlerinden birisi, daha fazla müşteri işbirliği ile, daha fazla müşteri memnuniyeti sağlamasıdır. Agile Proje Yönetimi, müşteriyi ön plana çıkarır ve onlarla beraber çalışmalar yapmanızı ister. Böylece gerçekten onların sorunlarını çözen şeyler üretmeye odaklanırsınız. Agile takımları diğer takımlara göre daha özertir. Buda takımlara diğer proje yönetim methodojilerinde olmayan yeni fikirler oluşturmaya, yenilikler geliştirmeye ve problem çözme becerisi verir. Fikrini özgürce ifade edebilen, projenin gelişmesine katkı sağlayan takımlar kendilerini daha değerli hissederler. Ekip ruhu bu şekilde pekişir.

Kanban Modeli

Kanban, tam zamanında Üretim ortamında malzeme hareketlerinin kontrolü amacıyla kullanılan bir çizelgeleme yaklaşımıdır. Toyota'nın üretim verimliliğini artırmak amacıyla aynı zamanda Lean Production (Yalın Üretim)'un da mucidi olan Taiichi Ohno tarafından geliştirilmiştir. Yöntem 1953'ten bu yana kullanılmaktadır[?3].

- **İş akışınızı görselleştirin;** İş küçük parçalara bölün, her küçük parçayı bir karta yazın ve bu kartları bir duvara yapıştırın. Kendinize özgü kolonlar yapın ve işlerinizi bu kolonlara taşıyın.
- **WIP(Work In Progress) Limiti koyun;** Aynı anda kaç iş alıp işleyebileceğinizi belirleyin ve bu sınırın dışına çıkmayın. Bu iş bitmeden başka iş almayın. WIP belirleme Kanban metodunda çok önemli yere sahiptir. Ancak olgunluğa erişene kadar sürekli değişerek geliştirilmelidir.
- **Lead Time (Teslim Süresi Ölçümü);** Bir parça işin ne kadar sürede teslim edilebileceğini tahminleyin. Bu sürenin ölçümü deneyseldir. Sürekli iyileştirilebilir. Ve bu süreye göre WIP limitinizi de iyileştirmiş olacaksınız.
- **Sürekli İyileştirme;** Özünü KAIZEN(kai:değişim,zen:daha iyi)'den alan bu yöntem ile üretim ve geliştirme hattınızdaki tıkanıklık yaratan noktaları ortaya çıkararak sürekli iyileştirmeye gidin.

Yalın Model

Lean'de temel fikir müşteriye sunulan değeri arttırırken israfı (waste) azaltmaktır. Bir başka ifadeyle, daha az kaynak ile daha çok değer üretmektir. Lean bir organizasyon, müşterinin değerini anlayan ve proseslerine bu değeri arttırmak için odaklanan bir yapıya sahip olmalıdır.

Lean bir maliyet azaltma programı ya da tekniği değil, tüm organizasyon için israfı azaltmaya, müşteri için değer üretmeye ve sürekli gelişmeye odaklanan bir yaklaşım biçimidir. Lean yaklaşımın temelinde insana saygı ve sürekli gelişme (kaizen) vardır.

Lean prensiplerinin kökeni lean (yalın) üretimden gelmektedir ve üretim ('just-in-time production') olarak da bilinen lean üretimin öncüsü Toyota'dır.

Lean üretim bir proses yönetimi felsefesidir ve Toyoto tarafından araba üretimi yaklaşımlarında kullanılmıştır. Kısa kullanımıyla lean, "lean üretim" felsefesinden gelmektedir ve orijini Toyota Üretim Sistemi'dir[?4].

En özet anlatımıyla israf, müşteri için değer kazandırmayan herhangi bir şeydir. Müşteri için değer üretmeyen her şey organizasyon için maliyet üretmeye devam ettiği için Lean'de israf olarak nitelendirilmiştir.

İsraf 3 farklı şekilde ortaya çıkacak şekilde ayrılmıştır:

- **Muda:** Aynı zamanda 7 israf (waste) tipi olarak da bilinir. Müşteri için değer yaratmayan bilakis kaynakları harcayan herhangi bir eylemin adıdır. Bu israfların ana kategorileri; bekleme, hata/onarım, atıl fazla üretim, gerekenden/ihitiyaçtan fazla işlem yapma, fazla/az taşıma, fazla stok ve gereksiz harekettir.
- **Muri:** Makina, ekipman ve insana kapasitesinin üzerinde iş yüklemek olarak tanımlanır.
- **Mura:** Üretim planlaması ve üretim miktarının düzenli olmadığı, geçici olarak yükselip alçalarak dalgalanmaların oluşması durumuna denir.

Scrum

Scrum, kelime olarak rugby oyununda oluşturulan küçük ekiplere verilen isimdir. Günümüzde çokça tercih edilen bir metottur. Eski yöntemlerdeki gibi projede izlenmesi gereken adımlar belirtilmemekte, bunun yerine basit ama önemli birkaç kuralı sayesinde esnek ama üretken bir işleyiş sunmaktadır. Scrum'ın sağladığı faydalardan biri de, işleyiş sürecini şeffaf hale getirerek aksaklıkları ortaya çıkarır. Böylelikle proje ekibinin hataları fark ederek sürekli olarak iyileştirme yapmalarına olanak sağlar.

Scrum'ın genel olarak çalışma mantığına gelecek olursak, yapılacak olan iş (Product Backlog) parçalara bölünür. Daha sonra işin belirli bir parçası (Sprint Backlog) ele alınır ve 2-4 hafta süreyle bu işin tamamlanması planlanır. Genelde 4 hafta olan bu süreye "Sprint" denir.

Klasik metodojilerde (Waterfall, ...) kullandığımız sıralı yaklaşımda: (Sequential Approach) gereksinim, analiz, tasarım gerçekleştirme gibi aşamalar vardır. Bu metodolojiler bir aşama bitmeden diğerine geçişe izin vermez. Scrum'ın bize getirdiği overlapping (örtüşme) yaklaşımıyla ise tüm sprinte bakıldığında bu 4 aşama örtüşen bir şekilde beraber ilerler. Aşamalar bir bütünlük gibi hareket edildiği için bir aşamada oyalanarak sıkılma olmaz ve sürekli bir ürün ortaya konulduğu için de müşteri bundan memnun olur.

Scrum' da 3 temel kavram vardır. Bunlar "**Roller**", "**Toplantılar**" ve "**Araçlar**" olmaktadır.

Rollere gelirsek ilk olarak bir **Ürün Sahibi (Product Owner)** olmalıdır. Bu kişi illa müşterinin kendisi olmak zorunda değildir. Bir müşteri temsilcisi veya Scrum ekibinin bir parçası da olabilir. Önemli olan bu kişinin müşteriyle iletişim kurabilmesi, müşterinin ne istediğini bilmesidir. Bu kişi ekiple birlikte hareket eder ve projede bizzat yer alır.

Scrum Yöneticisi' nin (Scrum Master) görevi takıma emirler verip yönetmek değildir.

Scrum Master: takıma yardımcı olan, bir problem olduğunda onu çözen, takımlar ve ürün sahibiyle olan iletişime yardımcı olan, takımın ve organizasyonun Scrum'a adapte olmasını sağlayarak üretkenliğin artmasına katkıda bulunan kişidir.

Scrum Takımı (Scrum Team) , birbirleriyle iletişim halinde olan ve programlama işini yapan gruptur. Kendi içlerinde farklı görevleri olabilir ama hedefleri ortaktır. Genellikle 5 ila 10 kişiden oluşur.

Toplantılar:

Sprint Planning de sprintlerde neler yapılacağı, gereksinimlerin listesi, takımların belirlenmesi, risk analizleri ve maliyet hesaplamaları yapılır.

Tüm sprint boyunca her gün takım üyeleri bir araya gelerek **15 dakika** boyunca ayaküstü toplantılar yaparlar. Bu toplantılara **“Daily Scrum Meeting”** denir. Bu toplantılarda önceki gün neler yapıldığı, ne gibi sorunlarla karşılaşıldığı, o gün neler yapılacağı ve nasıl yapılacağı hakkında özet şeklinde konuşmalar yapılır. Ayaküstü toplantı denmesinin sebebi, toplantı denilince akılda şekillenen karşılıklı oturup slaytlar üzerine konuşmalar değil, herhangi bir grup üyesinin masasının başında bile olsa toplanarak takım içi iletişimi kaybetmemek ve birbirine yardımcı olmak için görüş alışverişi yapılmasıdır.

Her sprint ardından bir **Sprint Gözden Geçirme (Sprint Review)** yapılır. Buraya herkes davet edilebilir ve yapılanlar gözden geçirilir. Bu toplantılar uzun sürebilir. Ardından bir sonraki sprinte hazırlık yapılır.

Üçüncü temel değer olan Araçlar 'ın (**Artifacts**) başında **Ürün Gereksinim Dokümanı(Product Backlog)** gelir. Proje boyunca tamamlanması gereken işlerin listesidir. Listeyi **product owner** yönetir. Bu liste genelde kullanıcı hikayelerinden (**User Story**) oluşur. Yani kullanıcının isteğine göre yeni gereksinimler eklenir veya çıkarılır.

Sprint Listesi (Sprint Backlog), bir sprint turunda takımın yapması gereken işlerin listesidir. Product Backlog 'ın bir alt kümesi olarak düşünülebilir. Bir sprint döngü için seçilen gereksinimler, görevlere bölünerek takımda bulunan üyelere dağıtılır.

Sprint Kalan Zaman Grafiği (Burndown Chart), bir **sprint turunda** görevler yerine getirildikçe yapılan iş ile kalan iş arasındaki bağıntıyı ortaya çıkaran grafikdir. Belirlen sürenin yeterli olup olmadığını, işin gidişatında bir sorun olup olmadığını ve işin zamanında bitip bitmeyeceği hakkında takıma ve yöneticilere bilgi sağlar.

Scrum, uzmanlık gerektiren ve maliyeti yüksek bir modeldir. Ancak kısa sürmesi, kolay uygulanması ve başarı garantisinin yüksek olması nedeniyle günümüzde kullanımı oldukça artmaktadır. Google, Microsoft, IBM ve Yahoo gibi büyük şirketler tarafından tercih edilmektedir.

Yazılım süreç modelleri birbirlerinden tamamen ayrı şeylerdir ve çoğu zaman beraber kullanılmaktadırlar. Bir proje için hangi modellerin ve hangi adımların uygulanacağını “süreç mimarı” seçer. Bütün modellerdeki güçlü ve zayıf yanları kıyaslayarak projeye en uygun olanları seçer. Bu seçim yapılırken; projenin büyüklüğü, karmaşıklığı, şirketin yapısı, zaman ve maliyet gibi kriterler göz önüne alınır. Günümüzde en çok tercih edilen ve geçiş yapılmak istenen model, Agile Modellerdir. Kısa zamanda, esnek biçimde ve yüksek başarı sağlaması tercih sebeplerinin başında gelir.

Karşılaştırma

Geleneksel modeller ve yeni çevik yöntemler arasında takım çalışmasına yönelik birçok farklılık bulunmaktadır. Bu farklılıklar bazen çalışma şekilleriyle, bazen de takımdaki uyuma göre değişiklik göstermektedir.

Yazılım geliştirme süreç değişimi etkileyen faktörler, aşağıdaki tabloda belirtildiği gibidir:

Kategori	Tema	Kod
Organizasyon	Üretkenlik	<ul style="list-style-type: none">BireyselTakımÖrgütsel
	Sosyal Yapı	<ul style="list-style-type: none">KişilikUyumYenilikçi
	Roller	<ul style="list-style-type: none">GeliştiriciTestçiLider
Gereksinim Analizi	İşlevsel	<ul style="list-style-type: none">MaliyetMemnuniyetYeniden kullanma
	İşlevsel olmayan	<ul style="list-style-type: none">KullanılabilirlikGüvenilirlikBirlikte çalışılabilirlikŞirketin büyüklüğüEntegrasyonMüdahale
İyileştirme aktiviteleri	Model	<ul style="list-style-type: none">KabiliyetlerTecrübeEsneklik
Çevik Metotlar	Lean Scrum Kanban	<ul style="list-style-type: none">Scrumban uygulamasıÇeviklik DeğerleriSüreç Disiplini
	Çevik	<ul style="list-style-type: none">Eşli programlamaGünlük Scrum Toplantıları
Etkileşim	Klasik	<ul style="list-style-type: none">Kod gözden geçirmeToplantı süreleri

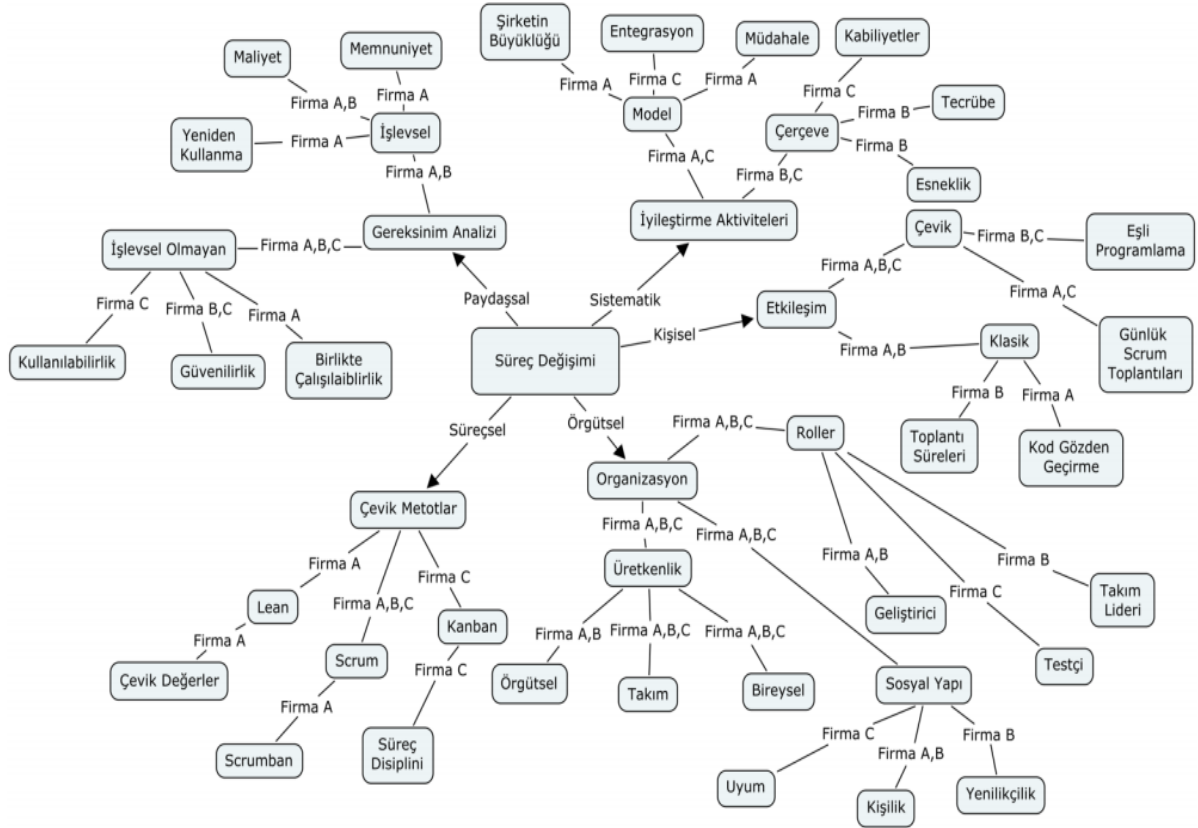
Organizasyonlarda üretkenliği direkt olarak etkileyen etkenler bireysel, takım ve örgütsel olarak tanımlanmıştır. Sosyal yapıyı etkileyenler kişilik, uyum, yenilikçi olup rolleri etkileyenler ise geliştirici, testçi ve lider olmaktadır.

Gereksinim analizinde ise projenin işlevselliğini maliyet, memnuniyet ve yeniden kullanılabilirlik gibi ölçütler etkilemektedir.

İyileştirme aktivitelerinde çerçeve olarak kabiliyetler, tecrübe ve esneklik kilit anahtar kelimelerdir.

Projelerde etkileşimler ise çevik ve klasik olarak ikiye ayrılmakla beraber çevik olarak eşli programlama ve scrum örnek verilirken klasik olarak kod gözden geçirme ve toplantı süreleri ele alınmaktadır.

Yazılım süreç değişimlerine yönelik görüşlerin, daha önceki araştırmalara dayanılarak firma bazlı düzenlenmiş kategori ve temaları aşağıdaki gibidir:



Firmaların kullandığı metodolojilere ve performanslarına bakıldığında, hacmi çok büyük olan firmaların çevik yöntemlere adapte olmasının daha zor olduğu gözlemlenmektedir. Ancak büyümekte olan firmalarda ise çevik yöntemler daha avantajlı olduğu tespit edilmektedir.

Yukarıda araştırılan firmaların içindeki bir şirket ilk yıllarında klasik plan-tabanlı yazılım geliştirme yöntemlerini benimsemiştir. Ancak, müşteri taleplerinin yıllar içinde değişmesi ve piyasa şartlarının hızlı ve teknoloji odaklı üretim istekleri şirketi çevik yöntemleri kullanmaya yöneltmiştir. Şirket, Türkiye'deki diğer birçok yazılım firması gibi Scrum metodolojisini kullanmayı tercih etmiştir. Bu durumun sebepleri şu şekilde sıralanabilir:

1. Üretim için 5-8 kişilik yazılım takımı ihtiyacı,
2. Müşteri temsilcisinin takımla sürekli bir arada bulunması gerekliliği,
3. 20 ile 26 günlük koşullar üzerinde ürün parçalarının üretim gerekliliği.

Şirket özellikle bazı dönemlerde her bireyin sabit bir rol ile iş yapması yerine herkesin her işi yapabilmesi ilkesine uygun bir kültür geliştirmiştir. Üretim planlaması çevik yöntemler sayesinde dört veya altı aylık teslimat dönemlerinden on veya on iki haftalık dönemlere çekilerek müşteri memnuniyeti sağlanmıştır.

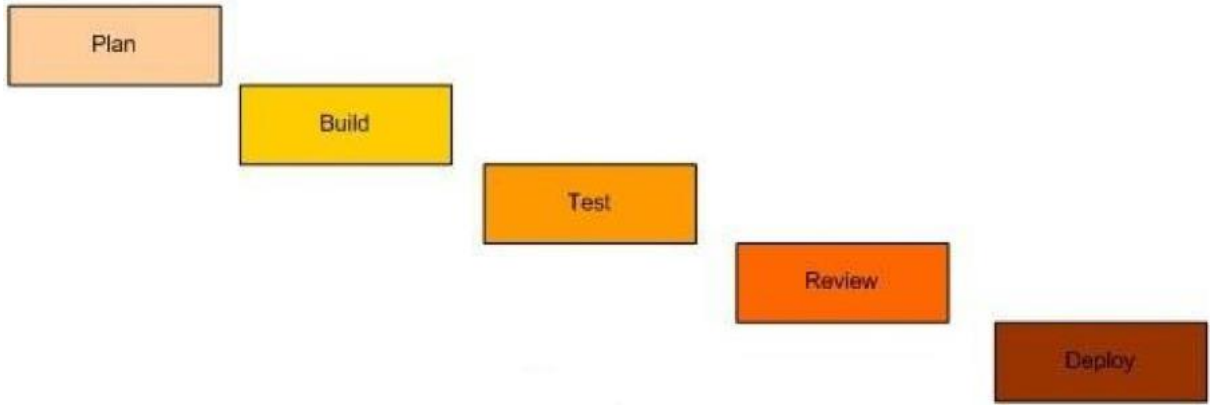
Yazılım Geliştirme Metodolojilerinin Karşılaştırması

Yazılım geliştirirken en çok kullanılan genel metodolojiler aşağıdaki gibidir:

1. Waterfall (Şelale) Modeli
2. Iterative Waterfall (Yinelemeli Şelale) Modeli
3. Agile (Çevik) Model
4. Scrum Modeli
5. Kanban Modeli
6. Lean (Yalın) Model

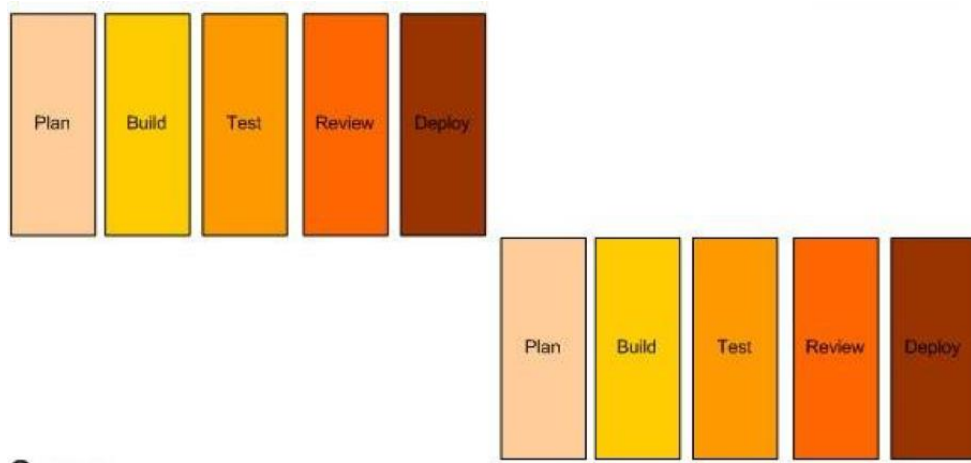
Bu metodolojilerin arasındaki farklar, şirketlerin kullanmak istediği, en işe yaradığını düşündüğü metodolojiyi seçme kararını verme aşamasında en belirleyici özelliklerdir. Bu farklar, takımlar arasındaki ilişkileri de değiştirirler.

Şelale modeli, klasik yazılım geliştirme yaklaşımı olarak da bilinmektedir. Bir faz bitmeden, diğer faza geçilmemesi bu metodolojinin en önemli kuralıdır. Önceki fazlar mükemmel olmadan diğer faza geçilemeyeceğinden maliyeti yüksek, verimi ise düşüktür. Önceki fazlarda değişiklik minimum olduğundan takımlarda anlaşmazlık ortaya çıkabilir.



Şelale Modeli

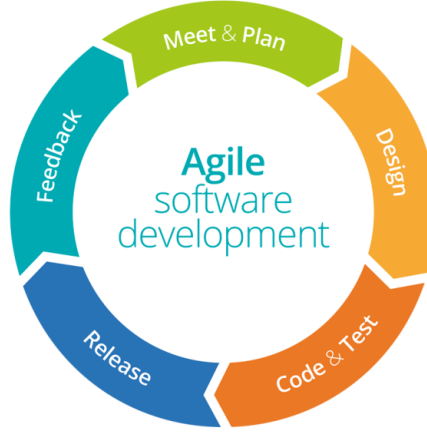
Yinelemeli şelale modeli, şelale modeline göre daha az riskli olmasına rağmen çevik yöntemlere göre yine daha riskli ve verimliliği düşüktür. Herhangi bir fazda problem olduğunda genellikle ilk fazdan itibaren sorun kontrol edilir. Tüm takım hedefe uyma konusunda kalan işlere odaklanmalıdır. Ancak takımın hızı ve kalan işler konusunda bir bilgi yoktur. Sprintin sonuna kadar süreci izlemek konusunda herhangi bir yol yoktur.



Yinelemeli Şelale Modeli

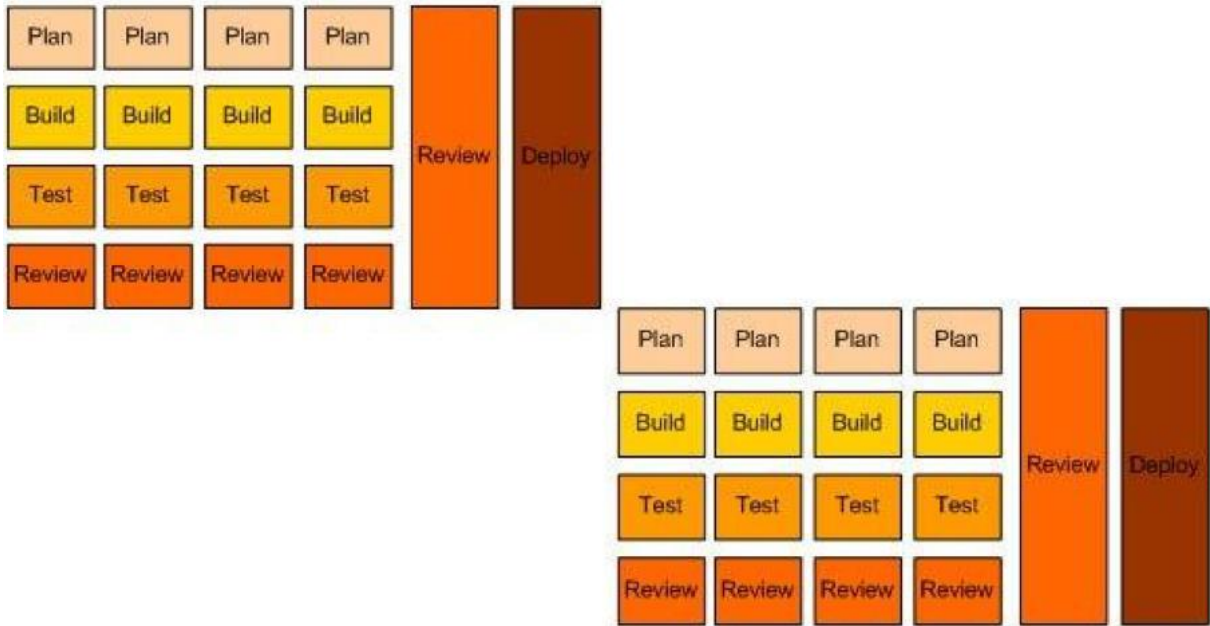
Çevik model, şelale modeli yöntemlerine göre daha az riskli olmakla beraber maliyeti biraz daha yüksek ancak verimi daha yüksek olacaktır. Bir hata olduğunda şelale modellerine göre daha kolay düzenleme yapılacaktır, en baştan farkedilecektir.

Proje için harcanan bütçe baştan anlaşılacak ve maliyetler şelale modeline göre daha da azalacaktır. Şelale modelinde proje tamamlanana kadar yatırımcıdan feedback alamazken çevik modelde sürekli feedback alınacağından yatırımcının isteklerini daha hızlı gerçekleştirebileceğiz, yanlış anlaşılmalardan uzak durulacaktır.



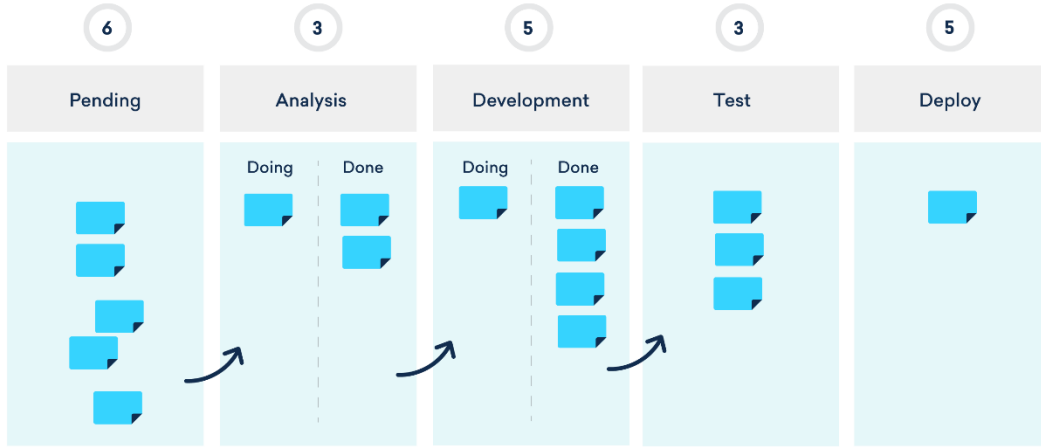
Çevik Model

Scrum modeli ise şelale modeline göre çok daha az risklidir ancak maliyeti biraz daha yüksektir. Bu modelde takım tam olarak test edilmiş, bağımsız, değer oluşturan küçük özelliklere odaklanır. Böylece risk dağıtılarak, eğer bir özellik geliştirilirken sorun çıkarsa bunun diğer parçaları geliştirmeye engel oluşturma riski azaltılmış olur. Bu modelde yine sprint adı verilen iterasyonlar oluşturulur ve her iterasyonun sonucunda istenirse uygulama yayına alınabilir.



Scrum Modeli

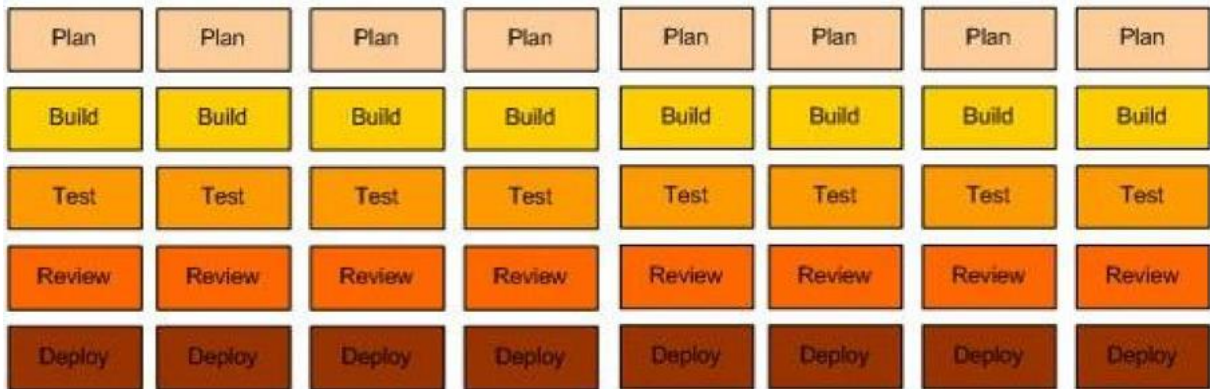
Oturmuş bir yapımız var ve işler yürüyor, fakat kimseyi ürkütmeden biraz gelişme kaydetmek ve hızlanmak istiyorsak doğru tercih Kanban olacaktır diye düşünüyorum. İşlerin yürümediği ve değişimin kaçınılmaz olduğu bir setup mevcutsa, daha büyük değişimlere açıksak, isterlerin henüz net olmadığı karmaşık bir proje ile uğraşıyorsak, Scrum doğru tercih olacaktır.



Kanban Modeli

Yalın metodolojisinde ise küçük parçalara bölünmüş özelliklere odaklanma konusunda Scrum modeline benzese de bu özelliği daha öteye taşımaktadır. Yalın modelde bir sonraki özelliğin planlama, geliştirme, test ve devreye alım süreçleri belirlenmeden önce yapılacak olan özelliğin planlama, geliştirme, test ve devreye alım aşamaları belirlenir. Böylece risk seviyesi özellik kırımına kadar inerek mümkün oldukça azaltılmış olur.

Yalın modelin temel yaklaşımı israfı engellemektir. Bu sebeple gerekli olmayan hiçbir özellik geliştirilmez.



Yalın Model

Sonuç – Yorum

Şahsen ben freelance olarak çalıştığım ufak çaplı projelerde ve ufak çaplı kişisel projelerimde waterfall metodolojisini kullanıyorum. Ancak orta çaplı projelerimde ve girişim projelerinde Kanban metodolojisini kullanmayı tercih ediyorum.

Waterfall metodolojisini kullanma sebebim ufak çaplı projeler olduğu için çok detaylı olmayışı ve daha hızlı geliştirme yapılabildiğinden olup proje tamamlandığında çok fazla istek çıkmayışındandır. Elbette uzun süren ve orta çaplı projelerim çok detaylı olduğundan ve genellikle takım çalışması gerektirdiğinden Kanban metodolojisi daha kullanışlı oluyor. Bir değişiklik gerektiğinde zorlanmadan takım içerisinde sorunu gidermek için gerekenler takım içerisinde bölüştürülebiliyor.

Scrum kullandığımızda takımın genellikle öğrenci olduğundan zorunlu toplantılara katılması biraz daha zor olduğundan pek verimli geçmemişti ancak Kanban'a döndüğümüzde Scrum'a göre çok daha verim almıştık.

Şu an hala da Toyota Üretim Sistemi'nin dünyaya katkılarından biri olan Kanban metodolojisini kullanıyoruz ve takım olarak verimimiz tamamen olumlu bir şekilde.

Kaynakça

- [1] Baltaş, A., Ekip Çalışması ve Liderlik, Remzi, İstanbul, 2000
- [2] Dengiz, G. M., Takım Çalışması Teknikleri, Academyplus Ankara, 2000
- [3] PEM360, Waterfall, Iterative Waterfall, Scrum ve Lean Yazılım Geliştirme Metodolojileri Arasındaki Farklılıklar, <https://www.pem360.com/blog/Agile/Waterfall-Iterative-Waterfall-Scrum-ve-Lean-Yazilim-Gelistirme-Metodolojileri-Arasindaki-Farkliliklar/167>
- [4] PEM360, Agile ve Waterfall Yaklaşımları Karşılaştırılması, <https://www.pem360.com/blog/Agile/Agile-ve-Waterfall-Yaklasimlari-Karsilastirilmesi/179>
- [5] PlanView, Top 6 Software Development Methodologies, <https://blog.planview.com/top-6-software-development-methodologies/>
- [6] ASPMVCNET, Şelale Yöntemi, <http://www.aspmvcnet.com/tr/m/yazilim-muhendisligi/selale-yontemi-waterfall-modeli.html>
- [7] Argenova, Agile Proje Yönetimi, <https://www.argenova.com.tr/agile-proje-yonetimi-nedir>
- [8] Muhammed Şimşek, Medium, Waterfall, Scrum, Kanban ya da hiçbiri, <https://muhammedsimsek.medium.com/waterfall-scrum-kanban-ya-da-hi%C3%A7biri-1e93c3b48663>
- [9] T4A, Lean Nedir?, <https://t4a.consulting/lean-nedir/>