

Uygulama Yaşam Döngüsü Yönetimi Karşılaştırmalı Süreç İncelemesi

Yagup Macit, Eray Tüzün
HAVELSAN, 06510 Ankara, Türkiye
{ymacit, etuzun}@havelsan.com.tr

Özet. Uygulama Yaşam Döngüsü Yönetimi (UYY), yaklaşık 40 yıldır süreç ve üretkenlik açısından araştırmalar yapılan bir alandır. Günümüzde, her şirketin faaliyet alanından bağımsız olarak bir şekilde yazılım üretim süreçlerine dâhil olması, UYY kavramını reel sektör konusu haline dönüştürmüştür. UYY, başlangıçta tüm kurumlar için sadece şelale (waterfall) süreç modeli anlamına gelmiş olsa da son dönemlerde özellikle proje başarımları açısından sorgulanmış ve çevik süreç modelleri gündeme gelmiştir. Çevik süreç modellerinin özel bir uygulaması olan Kenetlenme (Scrum) çerçevesi etkinlik kazanmıştır. Bu incelemede, geleneksel Şelale süreç modeli ile Kenetlenme süreç modelinin, proje yönetimi bakışı ile karşılaştırmaları yapılmıştır.

Anahtar Kelimeler: Uygulama yaşam döngüsü yönetimi, uygulama yaşam döngüsü süreç modelleri, şelale modeli, kenetlenme modeli, proje yönetimi

Abstract: Application Lifecycle Management (ALM) has a history of research related to software development processes and productivity over the last 40 years. Recently software has penetrated into almost every industry, thus every company is a software company now. Parallel to the history of ALM process models, at first it was considered synonymous to waterfall. However with the questioning of the success of waterfall methodology in software project management, recently agile processes come into play. Among the agile processes, especially Scrum methodology is the most popular and became prominent. In this study, we provide a comparison of Waterfall Model and Scrum model from project management perspective

Keywords: Application lifecycle management, application lifecycle management process models, waterfall model, scrum model, Project management

1 Giriş

Uygulama yazılımlarının fikir aşamasından itibaren geliştirme, dağıtım ve bakım süreçlerinin tamamı, Uygulama Yaşam Döngüsü Yönetimi (UYY) çerçevesinde incelenmektedir[1].

Konrad Zuse ile 1945 yılında başlayan bilgisayar yazılımları, 1956 yılında IBM ile başlayan endüstriyel amaçlı işletim sistemlerin ortaya çıkması sonucunda, uygulama yaşam döngüsü evrimine girmiştir.

Endüstriyel yazılımların gündeme gelmesi ile birlikte, bir ihtiyacın ele alınıp çözümlenmesi ve çalışır bir ürüne dönüştürülerek teslim edilmesi pratik bir sorun olarak ortaya çıkmıştır. Hem akademik çevreler hem de endüstriyel çevreler ortaya çıkan bu pratik soruna dönük yanıtlar bulmaya çalışmışlar ve bu çalışmalar sonucunda farklı süreç modelleri geliştirilmiştir. Bu çalışmada, endüstriyel olarak öne çıkan Şelale ve Kenetlenme süreç modelleri incelenmiştir.

Çalışmanın ikinci bölümünde, Şelale ve Kenetlenme süreç modellerin ortaya çıkışları ve kullanımları ele alınmıştır. Üçüncü bölümde ise, süreç modellerinin proje başarımı açısından önem taşıyan; proje yönetim üçgeni, ihtiyaca yaklaşma, görünürlük, değişime açıklık, risk yönetimi, değer üretimi, benzer proje üretkenliği ve karmaşıklık başlıklarında ele alınmıştır. Çalışmanın dördüncü bölümünde, üçüncü bölümdeki kriterlere dayanılarak özet bir süreç seçim yaklaşımı ortaya konmuştur. Beşinci bölümünde ise sonuç ortaya konulmuştur.

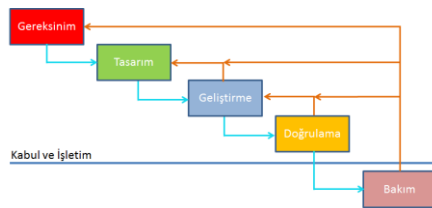
2 Süreç Modelleri

Yazılım geliştirme süreci, bir yazılım ürününün geliştirilmesi için organize edilmiş, kendi içerisinde sıralı adımları barındıran ayrıştırılmış fonksiyonlar kümesidir. Yazılım geliştirme süreci aynı zamanda bir yaşam döngüsü modelidir.

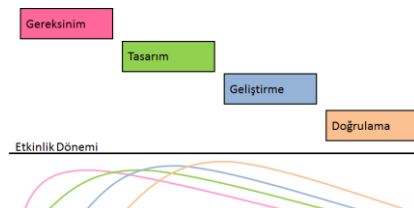
2.1 Şelale Süreç Modeli

Bilişim sektörünün gelişimi ile birlikte, 1970'li yıllarda yazılım üretimleri için standardize edilmiş ve iyi işleyebilecek bir üretim modeli isteklerine Royce tarafından Şelale Süreç Modeli ile [2] yanıt verilmiştir.

Royce tarafından verilen yanıt detaylı olsa da özet olarak Şekil 1'de görülen ana süreç ayrıştırmalarına (decomposition) sahiptir. Bu yanıt sonrasında, yazılım endüstrisinde çalışanların rolleri, yapılan sözleşmeler ve hakedişlerde hep bu süreç kırılımları esas alınmıştır. Bu kırılımlar sonucunda karmaşıklık ayrıştırma yöntemi ile çözümlenmiş ve projelerin takvimleri belirgin hale gelmiştir.



Şekil 1 Şelale Süreç Modeli



Şekil 2 Şelale Fonksiyon Etkinlikleri

Şelale modeli, tüm endüstri ve özellikle savunma sanayi için bir standart halinde geldikten sonra, Software Engineering Institute tarafından geliştirilen CMMI çalışmaları [3] dahi, aynı kırılımlar üzerinden yüklenici değerlendirmeleri yapmıştır.

Şelale modeli yaygın ve etkin olarak kullanıldıkça, eksiklikleri ve zayıf noktaları da ortaya çıkmıştır. Gereksinimlerin, projenin ilerleyen aşamalarda değişmesinin çok daha zor olması engelini aşabilmek için, gereksinim safhasına Prototip eklenmiştir.

Tüm ürünün doğrulamasının en sona bırakılması telafi edilemeyecek maliyet ve gecikmelere neden olduğundan, modelin doğrulama tarafı için V-Model[4] ile iyileştirmeler yapılmış ve her aşama da müşteri geri bildirimine devreye alınmıştır.

Bilişim sektöründeki projelerin gereksinim, tasarım, geliştirme ve doğrulama aşamalarının doğal akışında net olarak ayrıştırılmaması, şelale modelindeki yapay ayrıştırma ile çözülememiştir. Şekil 2’de gereksinim, tasarım, geliştirme ve doğrulama etkinlikleri, şelale basamakları ve dönemsel etkinlik yoğunlukları ile birlikte gösterilmiştir.

Şelale modelindeki evrelerin isimleri, o evre içerisinde en yoğun gerçekleşen fonksiyonel etkinliklerden alınmıştır. Evrelerin isimleri yoğunlukların göre farklı olsa da neredeyse her evrede tüm fonksiyonel alanlarda etkinlikler devam etmektedir. Örneğin, geliştirme evresinde, yoğun etkinlik geliştirme için yaşanırken, tasarıma ilişkin iyileştirici ve düzeltici etkinlikler, geliştirme için altyapı ve öncül üretim etkinlikleri yaşanabilmektedir. Bu durumda, evreler ve o evrelere ilişkin çıktılar izole şekilde değerlendirmemize karşın, neredeyse her evrede her fonksiyonel etkinlik türü farklı yoğunluklarda devam etmektedir. Bir evrenin kendi ana etkinlik türü dışındaki diğer etkinlikleri yok sayması, o etkinlikler ile iletişime kapalı olmasına neden olmaktadır. İletişime kapalılık beraberinde değişime kapalılığı da getirmektedir.

2.2 Kenetlenme Süreç Modeli

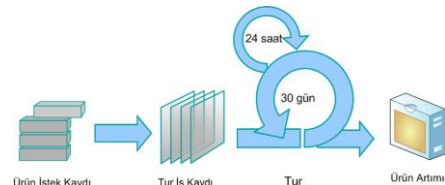
Endüstriyel üretim hatlarında, üst-üste binen üretim fonksiyonlarının geleneksel sıralı yaklaşımla ayrışık ve ardışık olarak ele alınması eş zamanlı etkinlikler için etkileşim sorunu oluşturmuştur. Bu sorun, 1986 yılında Hirotaka Takeuchi ve Ikujiro Nonaka tarafından Fuji-Xerox ve Honda örneğinde, Amerikan futbolu (Rugby) oyunundan esinlenilerek yeni geliştirme oyunu[5] ile çözülmüştür. Bu çözüm, 1990’lı yıllarda Ken Schwaber ve Jeff Sutherland tarafından bilişim uygulamaları için yapılan çalışmalarda kullanılmıştır[6].

Bilişim sektöründe yinelenmeli ve artımlı [7] geliştirme konusundaki deneyimler ile Scrum ve diğer deneyimlerin birikimi sonucunda 2001 yılında Çevik Bildiri (Agile Manifesto)[8] yayınlanmıştır. Çevik bildiri ile bireyler ve etkileşim, çalışan yazılım, müşteri ile işbirliği, değişime açıklık öne çıkan ana ilkeler olarak kabul edilmiştir.

Kenetlenme süreç modeli temel olarak deneyciliği (Empiricism) öne çıkartmaktadır. Deneycilik; kenetlenme açısından, görünürlük, denetim ve değişim kaldıraçları üzerinde şekillenmektedir.



Şekil 3 Kenetlenme Deneycilik Modeli[6]



Şekil 4 Kenetlenme Çerçevesi

Şekil 3’de görülen deneycilik modeli ile geleneksel sıralı yaklaşımın üretimin bütününe bir kez uygulanması yerine, küçük zaman dilimlerinde n defa tur olarak

uygulanması sağlanmıştır. Her bir tur sonucunda yapılan gözlemler ile sonraki tur için iyileştirmeler yapılmış ve mükemmel öngörü yerine deneyimleyerek öğrenme ve uygulama süreci tanımlanmıştır. Deneycilik öğretisi bilişim sektöründe termostat örneğine göre şekillenmiştir. Hiç bir mükemmel öngörü ve algoritmanın, termostatın şu anki sıcaklık derecesini ölçerek, bir sonraki eylemine karar vermesinin yerine geçemeyeceği öngörülmüştür.

Kenetlenme Çerçevesi, Şekil 4’de görüldüğü gibi, projenin ihtiyaçların belirli zaman dilimleri içerisinde tekrar eden turlar ile eritilmesini ve her tur sonunda çalışabilir çıktı elde edilmesini amaçlamaktadır.

Kenetlenme çerçevesi ile ilk tur sonunda bile belirli bir işi yapan bir ürün elde edilmesi hedeflenmektedir. Bütün çaba, her tur sonunda elde edilen çıktının çalışabilecek ürün olması için harcanmaktadır.

Tur başlangıcında, o tur için Ürün İstek Listesinden önceliği olan ve takım kapasitesi ile tur içerisinde üretilebilecek olan istekler, seçilerek detaylandırılmakta ve Tur İş Listesi oluşturulmaktadır. Tur içerisinde, üretim takımı, her gün aynı zaman ve aynı yerde günlük bilgi paylaşımı ve taahhüt toplantıları yaparak, Tur İş Listesini ürüne dönüştürmektedir. Tur sonunda, yapılan çalışma ürün sahibi tarafından değerlendirilerek, uygun olanlar Ürün Artımına dâhil edilmektedir.[9]

2.3 Süreç karşılaştırması ile ilgili Çalışmalar

Bu bölümde bahsedilen Şelale süreç modeli ve Kenetlenme süreç modeli literatürde farklı açılardan karşılaştırılmıştır. Süreç modellerinde belki de iki karşıt uç olarak tabir edebileceğimiz bu iki modelin, çevik metodolojilerin popüler olduğu 2000’li yıllardan sonra karşılaştırılmaya başlandığını görülmektedir. [10][11]

Vohra ve Singh [10] süreç karşılaştırmaya özellikle proje yönetimi açısından yaklaşmış ve süreç modellerini geleneksel plan tabanlı ve değişime açıklık tabanlı (çeviklik) olarak ikiye ayırmıştır. Temel olarak bu iki yaklaşımı değişime açıklık, ürün teslimat tarihi, kalite, müşteri katılımı, risk faktörü gibi açılardan karşılaştırmalı olarak incelemiştir.

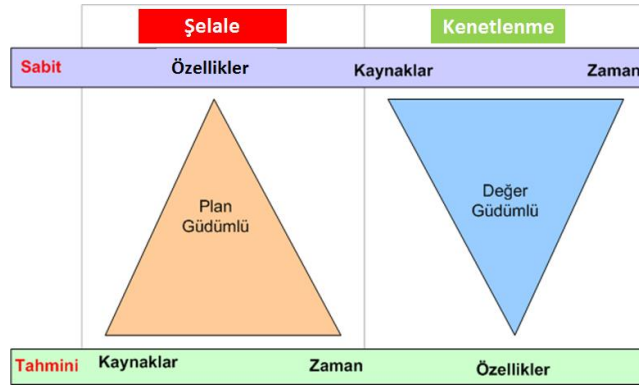
Karşılaştırma çalışmalarından günümüze en yakın olanı 2013 yılında Carnegie Mellon Yazılım Mühendisliği Enstitüsünün yayınladığı “Parallel Worlds: Agile and Waterfall Differences and Similarities” [11] adlı teknik rapordur. Bu çalışmada Çevik ve Şelale dünyasındaki terim ve kavramlar açıklanmış ve bunların arasındaki benzerlikler ve farklılıklar özetlenmiştir. Çevik ve şelale dünyasında kullanılan terminoloji özetlenmiş ve karşılaştırılarak anlatılmıştır.

3 Karşılaştırma

UYY süreç modelleri mühendislik ve proje yönetimi yaklaşımları ile karşılaştırılmıştır[12]. Ayrıca savunma sanayii açısından şelale süreci ve çevik sürecin farklılıkları ve benzerlikleri üzerinde çalışmalar yapılmıştır[13]. Bu çalışmada, şelale sürecini ve kenetlenme sürecini proje yönetim desteği açısından proje yönetim üçgeni, yabancılaşma, benzerlik, görünürlük, değişime açıklık, risk, değer üretimi ve karmaşa, başlıklarında karşılaştıracamız.

3.1 Proje Yönetim Üçgeni

Proje yönetimi açısından kısıtlı kaynakların ödünleşimi olarak bilinen proje yönetim üçgeni (iron triangle), projenin tercihleri ve değişkenlik boyutları açısından belirleyici olmaktadır. Bir projenin, hangi özellikler kapsamında, hangi zaman aralığında ve ne kadar bir kaynak ile yapılacak kararı, o projenin ihalesini, sözleşmesi ve kabulünü etkileyen ana etmenlerdir. Üçgenin bir boyutu üzerindeki değişikliğin kaybı veya kazanımı için diğer boyutlardan en az birinde de değişim gerektirmektedir. Örneğin, özellik boyutunda genişlemeye gidiyorsanız ya aynı kaynaklar ile zamanı arttırmak ya da aynı zaman diliminde kaynakları arttırmak gerekecektir. Şelale ve Kenetlenme için proje yönetim üçgenindeki değişken ve sabit boyutlar Şekil 5’de verilmiştir.



Şekil 5 Proje Yönetim Üçgeni

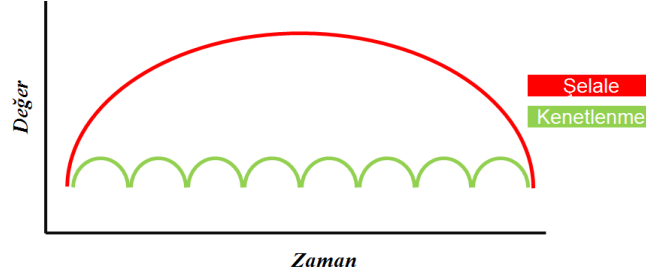
Şelale sürecini kullandığımızda, projenin kapsamını oluşturan özellikler kümesi sabit olarak karşımıza çıkmaktadır. Proje yönetimlerinin faaliyet planlaması açısından kullanabileceği temel seçenekler, zaman ve kaynak kullanımının eniyileştirilmesi şeklinde olacaktır. Sözleşmelerde zaman ve bütçe unsurları, kısıt olarak belirtilmiş olsa da belirtilecek şartlar içerisinde değişiklik sağlanabilmektedir. Teknik şartname olarak da bilinen özellik kümesi ise değişime daha kapalı bir özellik sergilemektedir.

Kenetlenme sürecini kullandığımızda, projenin bütçe ve hedeflenen sürüm tarihleri temel sabitler olarak karşımıza çıkmaktadır. Bu süreçte asıl hedef, belirlenmiş sürüm tarihleri için en öncelikli özellik kümesinin teslim edilmesidir. Bu nedenle özellikler arasında önceliklendirme yapılmaktadır. Eniyileme çalışması, bütçe ve sürüm planına sadık kalınarak en fazla değeri oluşturacak özelliklerin belirlenmesi şeklinde olacaktır. Kullanıcılar açısından ürün özelliklerinin istenilen düzeye gelmesi durumunda proje sonlandırılabilir.

3.2 İhtiyaca Yabancılaşma

Bilişim projelerinde, kullanıcı ihtiyacı ile bu ihtiyacı karşılamak için yapılacak ara üretim çalışmaları ve çıktılar, özgün ihtiyaçtan kopuş yaşamaya yani yabancılaşmaya neden olmaktadır. Bir projedeki kullanıcı ihtiyacı, önce formal gereksinime, sonra bir tasarım modeline, sonra bir yazılım koduna, sonra test prosedürüne ve sonunda

ihtiyacı karşıladığı düşünülen ürüne dönüşmektedir. Bu aşamalı ve ara teslimatlı üretim yaklaşımı, özgün ihtiyaçtan uzaklaşma ve yabancılaşma potansiyeli barındırmaktadır. Şelale ve Kenetlenme için yabancılaşma grafiği Şekil 6'da verilmiştir.



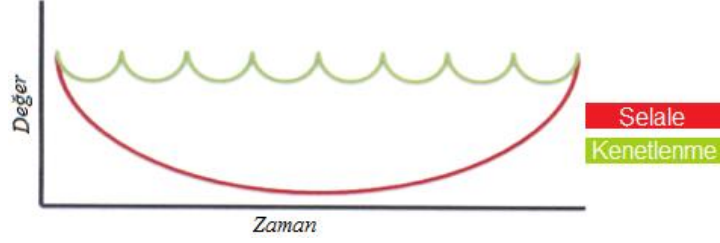
Şekil 6 Yabancılaşma Karşılaştırması

Şelale sürecini kullandığımızda, sadece projenin başındaki gereksinim analizi asıl ihtiyaç ile etkileşim halinde iken sonraki aşamalarda bir öncekinden devralınan yorumlanmış çıktı ile işlem yapılmaktadır. Bu şartlarda, bir yazılımı geliştirici için yazılması gereken kod, kullanıcı ihtiyacı değil, tasarım modelinin belirttiği şey olacaktır. Bu durumu projenin tüm ihtiyaçları ve üretim takvim ile birlikte değerlendirdiğimizde, kabul dönemine kadar yabancılaşma söz konusu olmaktadır.

Kenetlenme sürecini kullandığımızda, bütün proje süresi yerine, her tur başında kullanıcı ihtiyacı üretime alınmaktadır. Üretim aşamasında da kullanıcıyı doğrudan temsil eden ürün sahibi sürekli olarak çıktılar üzerinde doğrudan söz sahibi olmaktadır. Kullanıcının kendisi ise, her tur sonunda ihtiyacını karşılayan ürünü doğrudan görüp değerlendirebilmektedir. Bu durumda yabancılaşma etkisi, bir tur boyunca kullanıcıyı temsil eden ürün sahibinin yabancılaşması ile sınırlı kalmaktadır.

3.3 Görünürlük

Bilişim projelerindeki görünürlük tüm paydaşlar açısından karar girdisi olarak önem taşımaktadır. İşveren, yüklenici, alt yüklenici veya geliştiren personel açısından bakıldığında, beklentiler ve alacağımız kararlar açısından sistemin görünür olması gerekli görülmektedir. Şelale ve Kenetlenme için görünürlük grafiği Şekil 7'de verilmiştir.



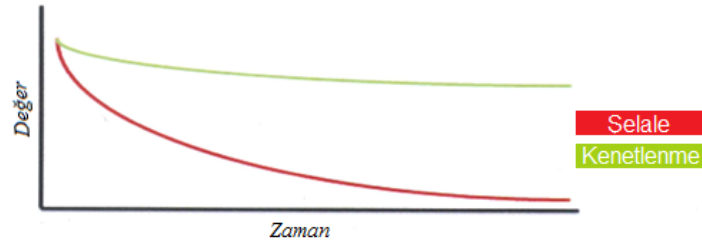
Şekil 7 Görünürlük Karşılaştırması

Şelale sürecini kullandığımızda projenin başında teknik şartname bilgisi ve analiz çalışmaları herkes için açık durumdadır. Sonraki tasarım ve geliştirme dönemleri faaliyeti gösterenler hariç herkes için kapalılığa ve belirsizliğe işaret etmektedir. İşlerin tekrar görünür hale gelmesi ancak doğrulama döneminde olanaklı hale gelmektedir.

Kenetlenme sürecini kullandığımızda, her tur sonunda her şey herkes tarafından açıkça görülebilir hale gelmektedir. Tur içerisindeki durumda ise belirli oranda şelale sürecinin proje süresindeki handikaplarını taşıdığı söylenebilir. Bilgi ihtiyacı duyan paydaşlar açısından bakıldığında en fazla bir aylık döngülerle tam görünürlük sağlanmaktadır.

3.4 Değişime Açıklık

Bilişim projelerindeki değişime açıklık, projenin başarı şansını ve ortaya çıkacak ürünün kullanılma şansını doğrudan etkilemektedir. Değişime açıklık konusu temel olarak ihtiyaçlardaki veya gereksinimlerdeki değişikliğe olan açıklığı ifade etmektedir. Şelale ve Kenetlenme için değişime açıklık grafiği Şekil 8'da verilmiştir.



Şekil 8 Değişime Açıklık Karşılaştırması

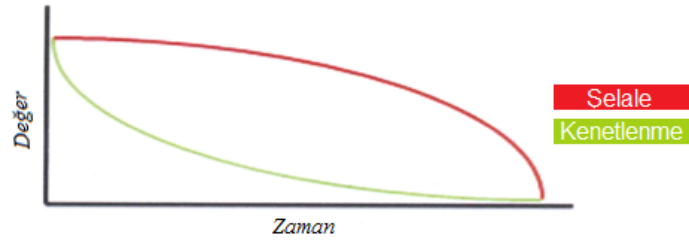
Şelale süreci, projenin başarıya ulaşmasının temel şartı olarak başlangıçta iyi tanımlanmış gereksinimlerin olması gerektiğini ifade eder. Gereksinimler üzerinde sonraki aşamalarda yapılacak değişiklik, tersine şelale zorlukları nedeniyle, dramatik sonuçlara ve büyük ek maliyetlere yol açmaktadır. Şelale süreci için gereksinimler, teknik şartname adıyla sözleşmenin ana eki olarak yer alırlar. Bu durum sözleşmenin kapsamını da belirler.

Kenetlenme süreci, her tur başında yeni bir projeye başlıyormuş gibi bir planlama sürecine sahiptir. Ayrıca her tur sonucunda ortaya çıkan ürünü kullanıma verme potansiyeline sahiptir. Kullanıcı, istemiş olduğu bir özelliğin pratik faydasının

olmadığını gördüğü zaman bir sonraki turda bu özelliğin iptalini veya değiştirilmesini isteyebilmektedir. Bu değişim olanağı ile proje sonunda kullanıcı ihtiyacı ile uyumlu bir ürün elde edilebilmektedir.

3.5 Risk Yönetimi

Bilişim projelerindeki risk yönetimi, temel olarak üretim ve tedarik zorlukları ile elde edilen ürünün ihtiyacı karşılayamama potansiyelinden kaynaklanmaktadır. Şelale ve Kenetlenme için risk grafiği Şekil 9’da verilmiştir.



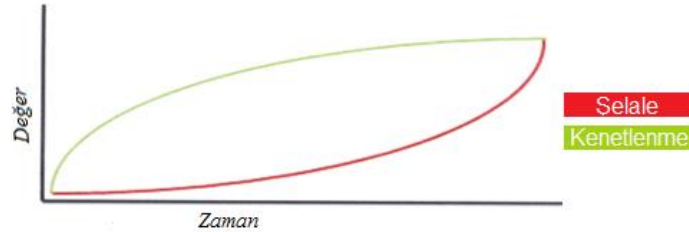
Şekil 9 Risk Karşılaştırması

Şelale süreci, proje başlangıcından itibaren hem üretim hem de kabul risklerini barındırmaktadır. Üretim riskleri planlanan takvim ve sözleşmeler ile azaltılabilmektedir. Ancak, V-Model ile azaltılmaya çalışılsa da kabul riskleri, entegrasyon testleri ve kabul aşamasına kadar risk envanterini doldurmaktadır.

Kenetlenme sürecini, grafik üzerinden risk eritme süreci gibi değerlendirilebilir. Gerçekleştirilen her tur ile hem üretim riskleri ortadan kaldırılmakta hem de tur sonunda teslim edilen çıktı ile kabul riskleri eritilmektedir.

3.6 Değer Üretimi

Değer üretimi bilişim projelerinin başından itibaren özel olarak takip edilen bir başlıktır. Kazanılmış değer analizleri tüm proje yöneticileri için önemli bir yönetim alanı olarak kendisini göstermektedir. Bir proje, bir zaman diliminde incelendiğinde, geçmiş olan süreye bağlı olarak beklenen üretim değeri değişecektir. Şelale ve Kenetlenme için değer üretimi grafiği Şekil 10’de verilmiştir.



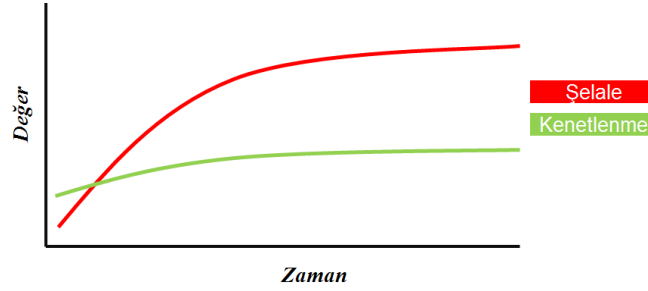
Şekil 10 Değer Üretimi Karşılaştırması

Şelale süreci, proje başlangıcında gereksinim analizi ve tasarım üzerinde yoğunlaşmak zorunda kaldığı için bu aşamalarda ürünsel bir değer üretimi söz konusu olmamaktadır. Bu dönemlerde elde edilen varlıklar sadece dokümanlardan oluşmaktadır. Geliştirme evresi ile birlikte müşteri için değer ifade edebilecek çıktılar üretilebilmektedir. Doğrulama evresi ile beklenen değerleri elde edebilmektedir.

Kenetlenme süreci, ilk turdan itibaren değer üretme üzerine odaklanmış durumdadır. Her tur ile değer üretimini arttırılmaktadır. Proje hangi zaman diliminde incelenirse incelenir her aşamada geçen süreye göre anlam ifade eden üretilmiş değer elde edilebilmektedir.

3.7 Benzer Proje Üretkenliği

Endüstriyel bakışla, belirli bir dönme içerisinde gerçekleşen projelerin arasında; alan, kapsam ve teknolojiler açısından benzerlikler bulunabilir. Benzer projelerin yakın zaman dilimleri içerisinde gerçekleştirilmesi, proje organizasyonu ve diğer paydaşlar için öğrenilmiş patika ve roller oluşturabilmektedir. Her organizasyonun tekrarlı projelere vereceği öğrenme ve hatırlama tepkisi farklı seviyelerde olabilmektedir. Şelale ve Kenetlenme için değer üretimi grafiği Şekil 11'de verilmiştir.



Şekil 11 Benzer Projeler için Üretkenlik Karşılaştırması

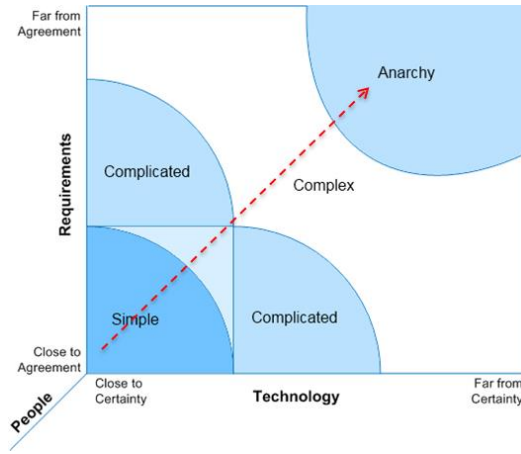
Şelale süreci, yeni bir organizasyonla yeni bir projeye başladığında, rollerin ve iletişimin oturmaması nedeniyle süreci yavaş işletecektir. Aynı proje organizasyonu ile benzer ikinci proje üretimine geçildiğinde, ilk projede yaşanan öğrenme sürecinin sonuçları doğrudan kullanılarak doğrudan üretime geçilebilecektir. Tekrarlı proje sayısı arttıkça organizasyonun olgunluk ve üretkenlik düzeyi yükselecektir. Bu düzey, tüm değişkenlerin en iyi şekilde organize edilebileceği noktada sabitlenecektir.

Kenetlenme sürecinin, proje başlangıç toplantısı ile doğrudan ilk tur çıktısını üretmek için organize olma çabası, farklılık yaratacaktır. Proje kapsamındaki bir özellik kümesinin her bir tur yeniden üretilmesi ve teslim edilmesi, şelale deneyiminin bütün proje boyunca yaşadığı öğrenme ve optimize olma deneyimini kenetlenme sürecinin aylık olarak yaşamasına olanak sağlamaktadır. Kenetlenme sürecine ilk projede dahi hız kazandıran planlama toplantıları, takım modeli, ürün özelliklerini işleme ve gözden geçirme mekanizması, benzer projeleri tekrar gerçekleştirirken hızlanmasını engelleyen ritüellere dönüşmektedir. Bu nedenle üretkenlik artışı sınırlı seviyede kalmaktadır.

3.8 Karmaşıklık

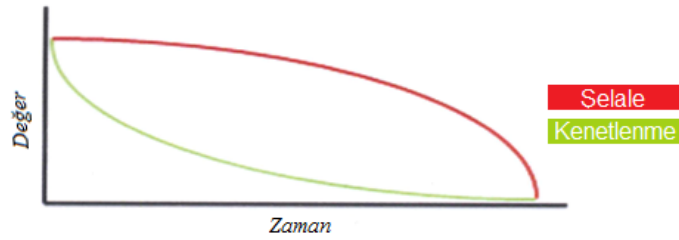
Karmaşıklık, bilişim projelerinin maliyetini, süresini ve başarımını etkileyen faktörlerden biridir. Projeler; sözleşme kapsamı, kullanılacak teknoloji ve organizasyon nedeniyle belirsiz taşıyabilmektedir. Sözleşme kapsamı belirgin olarak nitelendirilse de oradaki ihtiyaçların etkileşimli bir bütünlük içerisinde anlaşılması teklif açısından bile zorluklar barındırmaktadır.

Koşulların belirgin olmadığı ve öngörülebilirliğin düşük olduğu durumlar için Stacey tarafından Şekil 12’de görülen grafiksel çözümlene geliştirmiştir[14].



Şekil 12 Belirsizlik ve Karmaşa Haritası

Grafik araç üzerinde üç boyut bulunmaktadır. Bu boyutlar, organizasyon, gereksinim ve teknolojiden oluşmaktadır. Organizasyondaki insan sayısı arttıkça yönetim karmaşıklaşır hatta kaosa dönüşür. Gereksinim ve teknolojiye belirsizlik arttıkça karmaşıklık artar ve hatta karmaşaya dönüşür.



Şekil 13 Karmaşıklık Karşılaştırması

Şelale süreci, projenin akılcı ve nedensel bir akış ile ilerleyerek tamamlanacağını öngörür. Ayrıca, işin en başında iken tasarım, geliştirme ve test planlarının keskin bir şekilde yapılabileceğini öngörür. Şelale evrelerinden her biri öncekinin sonucu, sonrakinin nedenidir. Her bir evre, öncekinden devraldığı karmaşaya, yeni notasyonlar ve roller ile birlikte yeni karmaşalar ekleyecektir. Karmaşanın azalması ancak kabul döneminde, olayların öngörülebilecek uzama girmesi ile azalacaktır.

Kenetlenme süreci, karmaşayı en baştan kabul ederek, yönetilebilecek ve tepki verilebilecek bir miktarda tutmaya amaçlamaktadır. Tur içerisine alınan işlerin azlığı teknoloji, ihtiyaçlardaki belirsizlik, organizasyon ve kullanıcıların ürüne vereceği tepkilerden oluşan karmaşayı hep belirli bir dozda tutabilmektedir. Tur sonlarında yaşanan gözden geçirme ve değişim ile karmaşadaki değişikliğe göre tepki verebilme olanaklı hale gelmektedir.

4 Süreç Seçim Yaklaşımı

Yeni bir projeye başlanılacağı zaman hangi kısıtlar, varsayımlar ve yaklaşım ile hangi proje yönetimi sürecinin seçilebileceği önemlidir. Süreç karşılaştırma bölümünde yer alan alt başlıklar, süreç seçiminde kullanılacak kriter listesi olarak kabul edilebilir. Karşılaştırma başlıklarına göre süreç seçim yaklaşımı Tablo 1’de görülmektedir.

Kriter	Şelale	Kenetlenme
Proje Yönetim Üçgeni	Gereksinim sabit, bütçe ve zaman değişebilir.	Bütçe ve zaman sabit, gereksinimler değişebilir.
İhtiyaca Yabancılaşma	Son kullanıcı ürünü uzun süre sonra değerlendirebilir.	Son kullanıcı ürünü kısa süreli teslimatlar ile erkenden değerlendirebilir.
Görünürlük	Sözleşme doküman tabanlı iletişimi öngörmektedir.	Sözleşme ve işveren iletişime açıktır.
Risk	Proje önemli riskler içermemektedir.	Projenin önemli riskler içermektedir.
Değer Üretimi	Kabul ve kullanım proje sonunda öngörülmektedir.	Kısa aralıklar ile kazanılmış değer üretimi ve kullanımı öngörülmektedir.
Değişime Açıklık	Gereksinimler değiştirilemez.	Gereksinimler kullanıcı ihtiyacına göre değişebilir.
Benzer Proje Üretkenliği	Yakın zamanda benzer proje deneyimi var.	Benzer proje deneyimi yok.
Karmaşıklık	Teknoloji ve gereksinimler belirgin.	Teknoloji ve gereksinimler belirsiz.

Tablo 1 Kriterler ve Süreç Seçimi

Kriter ve süreç seçim tablosu, ilgili karşılaştırma satırında, her bir süreç için öngörülen değerlendirmeyi barındırmaktadır. Karşılaştırma başlıkları için, yönetim tercihinin göre istenilen ağırlık ile çekim merkezi nitelemesi yapılabilmektedir.

5 Sonuç

1970 ve sonrasında elde edilen şelale süreç modeli ile özellikle öngörü sahibi olunan alanlarda başarılı sonuçlar alınmış ve yazılım geliştirme endüstrisi gelişmiştir. Ancak, bilinirliğin azaldığı ve sürenin uzadığı projelerde şelale modeli zorlanmıştır.

Kenetlenme modeli, bu durumda, bilinmezin ışığını bulmamıza yardımcı olmuştur. Süreç seçimi için Stacey grafiğine benzer yaklaşımlar işimizi kolaylaştırmıştır.

Eğer gereksinimlerini ve teknolojisini bildiğiniz bir işi yapıyorsanız ve dolayısıyla basit bölgeye düşüyorsanız kenetlenme sürecinin deneycilikle öğrenme maliyetinden kaçınmın ve şelale süreci ile projenizi geliştirin.

Eğer, gereksinimlerin belirsiz olduğu ve teknolojisi bilinmeyen bir işi yapıyorsanız ve dolayısıyla karmaşık bölgeye düşüyorsanız deneycilikten faydalanın ve kenetlenme süreç modeli ile projeniz geliştirin.

Teşekkür. Yazarlar, HAVELSAN yönetimine çalışmaya verdiği destek için teşekkürler ederler

Kaynaklar

- [1] D. Chappell, "What is Application Lifecycle Management?", Chappell & Associates 2008.
- [2] W.W.Royce, "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON 26 (August): 1-9 . pp.328-338, 1970]
- [3] CMMI Product Team, "CMMI for Development, Version 1.3," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2010-TR-033,2010.
- [4] K.Forsberg, H. Mooz, "The Relationship of System Engineering to the Project Cycle", Proceedings of the First Annual Symposium of National Council on System Engineering, p: 57-65, October 1991
- [5] H. Takeuchi and I. Nonaka, "The new new product development game," *Harvard business review*, no. January-February, pp. 137-147, 1986.
- [6] J.Sutherland, "Agile Development: Lessons learned from the first Scrum", 10-2004
- [7] Gerald M. Weinberg, as quoted in Larman, Craig; Basili, Victor R. (June 2003). "Iterative and Incremental Development: A Brief History". *Computer* 36 (6): 47-56.
- [8] M. Fowler and J. Highsmith, "The agile manifesto," *Software Development*, vol. 9, no. August, pp. 28-35, 2001.
- [9] K. Schwaber and J. Sutherland, "The scrum guide," *Scrum. org, October*, vol. 2, no. July, p. 17, 2011.
- [10] Pankaj Vohra and Ashima Singh. Article: A Contrast and Comparison of Modern Software Process Models. IJCA Proceedings on International Conference on Advances in Management and Technology 2013 iCAMT:23-27, February 2013
- [11] Palmquist. Steven, Lapham. Mary Ann, Garcia-Miller. Suzanne, Chick. Timothy, and Ozkaya. Ipek, "Parallel Worlds: Agile and Waterfall Differences and Similarities," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Note CMU/SEI-2013-TN-021, 2013.
- [12] N. Mohammed Ali Munassar and A. Govardhan, "A Comparison Between Five Models Of Software Engineering", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, pp.94-101, September 2010.
- [13] R. Stacey, "Complexity and Emergence in Organizations" (London: Routledge, 2001).
- [14] J.Sutherland, K.Schwaber, "Software in 30 Days", John Wiley & Sons, Inc., Hoboken, New Jersey, 2012