

Mobil Uygulama Sistemlerinde Çevik Yazılım Geliştirme Metodunun İncelenmesi

Investigation of Agile Software Development Method On Mobile Application Systems

İsmail DURU, Erdem KÖSE, Oya KALIPSIZ

Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul
iduru@yildiz.edu.tr, erdemkose@gmail.com, kalipsiz@yildiz.edu.tr

Özet

Son yıllarda mobil bilgisayarlardaki gelişmeler ve cep telefonlarından internet kullanımının yaygınlaşması ile mobil uygulama geliştirme sayısı çok arttı. Var olan yazılım geliştirme metodolojileri mobil platformlar için bu yöntemlerin nasıl uygulanabileceğine özel olarak değinmemektedir.

Makalemizde kendine özgü koşulları dikkate alınarak mobil uygulama geliştirme platformlarında çevik uygulama geliştirme sürecinin nasıl uygulanabileceği çevik yazılım geliştirme prensiplerinden faydalanılarak açıklanmaktadır.

Çalışma kapsamında çevik yazılım geliştirme metodolojilerini kolaylaştırmak için geliştirilen Trello aracından faydalanılmış ve bu aracın özelliklerinden bahsedilmiştir. Aynı zamanda mobil uygulama geliştirmede çevik yazılım geliştirme kullanımı sonucu görülen farklar ortaya konulmuştur.

Çalışma sonucunda mobil uygulama sistemlerinde çevik yazılım geliştirme yönteminin uygulanması ile elde edilen sonuçlar hakkında bilgilendirmede bulunulmuştur.

Anahtar Kelimeler: Mobil Uygulama, Android, Proje Yönetimi, Çevik Metodoloji

Abstract

In the last few years, with the development on mobile computers and extension of internet usage on mobile phones, mobile application development counts increased so much. Current software development methodologies arent mentioning from usage of these methods for mobile platforms, specially.

In our article, with the special conditions how to implement agile application development on mobile application development platforms as utilizing agile software development principle is explained.

In content of our studying, Trello tool which is developed to facilitate agile software development methodologies is utilized and its priorities are also mentioned. At the same time, differences which are seen as a result of using agile

software development on mobile application development also displayed.

At the end of this paper, informations about results of using agile software development methodology on mobile application systems are displayed

Key Words: Mobile Application, Android, Project Management, Agile Methodology

1. Giriş

Yazılım proje yönetimi; yazılım, proje ve yönetim kavramlarının yazılım projelerine uygun yazılım geliştirme süreçlerinin seçilmesini ve proje süreci içerisinde etkin bir şekilde yönetimin gerçekleştirilmesini sağlayacak şekilde bir araya getirilmesi ile oluşturulmuştur.

Yazılım proje yönetimi; risk yönetimi, kapsam yönetimi, iletişim yönetimi, maliyet yönetimi, tedarik ve sözleşme yönetimi, zaman yönetimi, İK yönetimi, kalite yönetimi kavramları ile doğrudan ilişkilidir.

Yazılım proje yönetim modelleri yazılım ile ilgilenen ve yazılım projelerinde yer alan veya yer alacak olan herkesin bilmesi gereken modellerdir. Yazılım projelerinde şelale modeli, artırımlı model, çevrimsel model ve çevik yazılım geliştirme modeli gibi modeller uygulanmaktadır.

Şelale modeli, yazılım yaşam döngüsü adımlarının baştan sona en az bir kez izlendiği, iyi tanımlanmış projelerde ve yapımı için az süre gerektiren projelerde daha fazla tercih edilen bir yöntemdir.

Çevrimsel model, risk analizinin ön plana çıktığı yinelemeli ve artırımsal bir yaklaşıma sahiptir. Bu modeldeki her döngü bir fazı ifade eder ve model içerisinde doğrudan adım tanımlama gibi bir faz yoktur. Model ayrıca prototip yaklaşımına sahiptir ve içerdiği süreçler 4 gruba ayrılır. Bu süreçler; planlama, risk analizi, üretim, kullanıcı değerlendirmesidir.

Artırımsal model, üretilen her yazılım sürümü birbirini kapsayacak ve giderek artan sayıda işlev içerecek şekilde geliştirildiği yazılım geliştirme süreç modelidir. Uzun zaman alabilecek ve sistemin eksik işlevle çalışabileceği türden projeler bu modele daha uygundur.

Çevik yazılım geliştirme, projeyi aşamalı olarak geliştirmeyi öngörerek, bu aşamalar sırasında mümkün oldukça esnek davranmaya imkan tanıyan ve kullanıcılarla iletişim halinde olunmasını öneren bir proje yönetim metodudur.

Yazılım projelerinde yazılım geliştirme süreç modelinin doğru seçilememesi veya yeterince etkin kullanılmaması; projelerin zamanında bitirilememesi, müşteri isteklerine uymayan yazılımların geliştirilmesi ve projelerin başarısızlıkları gibi sonuçlar doğurmaktadır. Karşılaşılan bu problemlere karşı bir çözüm üretmek için dünyanın farklı yerlerinden 17 profesyonel yazılımcı projelerin başarısızlık oranını düşürme, müşteri memnuniyetini artırma ve sorunlara karşı çözüm üretmek için Amerika'nın Utah eyaletinde ve 2001 yılı Şubat ayında toplanarak Çevik Yazılım Geliştirme Manifestosu'nun ortaya koymuşlardır[1].

Yazılım geliştirme süreçlerinden kaynaklanan sorunlara karşı bir çözüm olarak üretilen çevik yazılım geliştirme metodolojisinin çözüm getirdiği sorunlar;

- Teknolojinin çok hızlı gelişmesi ve yeniliklerin projeye uygulanamaması
- Müşterilerin proje başlangıcında gereksinimlerini tam olarak ortaya koyamamaları
- Müşterilerin gereksinimlerinin çok sık değişmesi nedeni ile ortaya çıkan yazılımın müşterilerin güncel ihtiyaçlarına cevap verememesi
- Her geçen gün projelerin yönetiminin gittikçe daha zor ve karmaşık hale gelmesi

Çevik yazılım geliştirme sürecinin amacı; planlama, dökümantasyon, süreç ve araçlardan öte müşteri memnuniyeti, çalışan yazılım, uyumlu yazılım geliştirme takımı ve müşteri isteklerine uygun olarak kısa zamanda geliştirilebilecek yazılımları üretmektir.

Çevik yazılım geliştirme; süreçlerin, dökümanların ve tasarımların proje başlangıç aşamasında tanımlanmasını değil, yazılım geliştirme aşamasında karşılaşılan sorunlara ve değişen koşullara göre uygun ve gerekli kararların verilmesi gerektiğini savunur.

Çevik yazılım geliştirme metodolojisinde Çevik Yazılım Geliştirme Manifestosu'nun ilkelerine uymak esastır. Bu manifestoda birey ve etkileşimler, çalışan yazılım, müşteri işbirliği ve değişime karşılık vermek önemli ilkeleri oluşturmaktadır[1].

2. Mobil Uygulamalarda Çevik Yazılım Geliştirme Modelleri

Mobil uygulama geliştirmede diğer uygulamalardan farklı olarak çevik proje geliştirme metodolojisini kullanmayı gerekli kılan bazı kısıtlar ve gereksinimler vardır. Bunlar:

- Mobil uygulama geliştirmede istekler ve platformlar çok sık değişebilmesi [2]
- Mobil uygulamalarda geliştirilen ürünün pazara hızlı çıkma zorunluluğunun olması [2]
- Mobil yazılım geliştirmede genelde küçük uygulama geliştirme ekiplerinin yer alması [3]
- Mobil uygulamaların boyutları genelde 10.000 satırdan daha az olması [3]

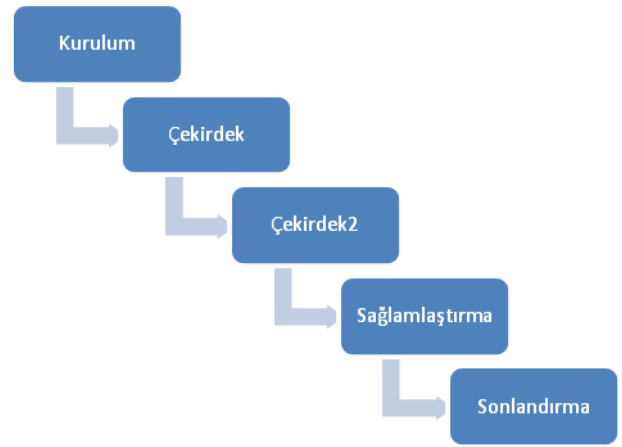
Bu kısıtlar ışığında mobil uygulama geliştirmede kullanılabilecek çevik yöntemler aşağıdaki gibi sıralanabilir:

- Mobile-D
- Hibrid Metod Mühendisliği Yaklaşımı
- Scrum
- MASAM

2.1. Mobile-D Yöntemi

Mobil uygulama sistemlerinde Java ME, Symbian, iOS ve Android gibi işletim sistemleri ile kapalı sistemden açık platforma geçiş ile birlikte ortaya çıkan güncel durum ve gelecek öngörüsü incelenmiş ve çevik süreçler ile mobil uygulama geliştirme sorunlarına çözüm olarak önerilmiştir.

Mobile-D yaklaşımını takip eden bir geliştirme projesi beş aşamaya bölünmektedir [4].



Şekil 1: Mobile-D yaklaşımındaki aşamalar

Bu yöntemdeki her aşama üç tür geliştirme gününe sahiptir.



Şekil 2: Mobile-D Geliştirme Günleri

Bu modelde aynı ürünün farklı parçaları üzerinde aynı anda birden fazla takım çalışıyorsa ayrıca bir bütünleştirme gününe ihtiyaç duyulmaktadır.

Bu aşamaların uygulamalarında dokuz temel bileşen yer almaktadır. Bu bileşenler aşağıdaki gibi sıralanabilir:

1. Aşamalama ve yürütme
2. Mimari Çizgi
3. Mobil Test Temelli Geliştirme
4. Devamlı Birleştirme
5. Eşli Programlama
6. Ölçütler
7. Çevik Yazılım Geliştirme Süreci
8. Şirket Dışı Müşteri
9. Kullanıcı Merkezli Odak

2.2. Hibrid Metod Mühendisliği Yöntemi

Hibrid Metod Mühendisliği yöntemi geliştirilirken mobil uygulama geliştirme metotları incelenip, mobil uygulama

karakteristiği gözden geçirilmiş ve yapılan bu işlemlerden sonra mobil uygulama geliştirmedeki zorluklar tespit edilmiştir [3].

Bu gözlem ve sonuçlara bağlı olarak çevik süreçlerin mobil uygulama geliştirme için uygun olduğu kabul etmektedir.

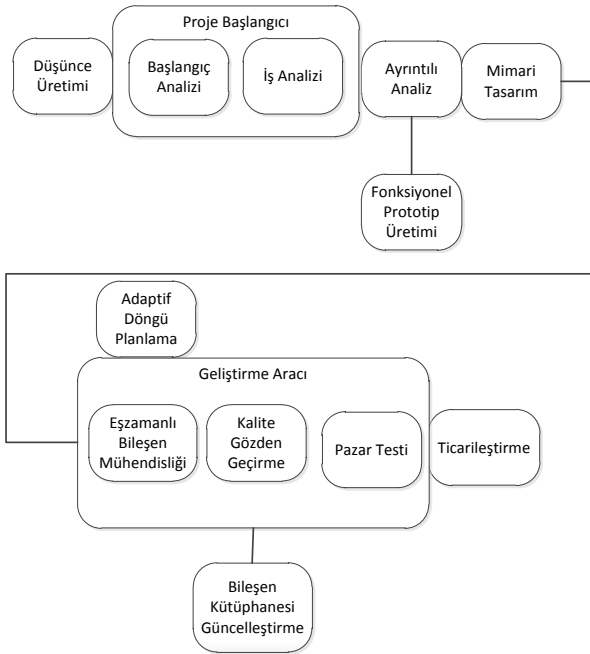
Hibrid Metod Mühendisliği yöntemi ile mobil uygulama geliştirmeye yönelik yeni bir çevik süreç önerilmektedir.

Hibrid Metod Mühendisliği genel yazılım geliştirme hayat döngüsüne ait modeli alıp, 3 farklı iterasyondan geçirmektedir.



Şekil 3: Genel Yazılım Geliştirme Hayat Döngüsü

Bu iterasyonlar sonucunda mobil uygulama geliştirme için yöntemin oluşturduğu final model aşağıdaki şekildeki gibi ortaya konmuştur:



Şekil 4: Hibrid Metod Mühendisliği İle Geliştirilen Final Yöntem

2.3. Scrum Yöntemi

Scrum metodunda mobil uygulama geliştirme projelerinde anında öğrenip geliştirmeyi sağlayacak şekilde Scrum metodunun mobil uygulama geliştirmede kullanımı incelenmektedir.

Bu yöntem ile geliştirilen uygulamanın testi gerçek bir proje sahibi ve gerçek bir Scrum uzmanı ile gerçekleştirilmiştir [2].

Bu yöntemde Scrum metodu kullanılarak kısa sürede başarılı bir ürün gerçekleştirilmesi hedeflenmiştir ve bu yöntem kullanılarak yapılan çalışmanın sonucunda bu hedefin gerçekleştirilebileceğine yönelik hipotez doğrulanmıştır.

Bu yöntemle göre Scrum başarısındaki anahtar etkenler aşağıdaki gibi verilmiştir [2]:

- Güçlü bağlılık ifadesi (göreve bağlılık)
- Planlanmış görevlere odaklanma

Bu yöntem ile gerçekleştirilen çalışmaların bir sonraki aşamasında farklı ülkelere dağılmış geliştiriciler kullanılarak nasıl bir sonuç elde edileceğini gözlemlemek hedeflenmektedir.

2.4. MASAM Yöntemi

MASAM Kore'deki bir mobil uygulama geliştirme şirketinde uygulanan bir yöntemdir. MASAM Mobile-D'den sonra geliştirilmiştir ve MASAM metodolojisi [5] Eclipse Süreç Çerçevesi [6] olarak tanımlanmış ve dağıtımına açılmıştır.

MASAM, Mobile-D ile benzer olarak klasik yaşam çevrimi veya şelale yerine çevik yaklaşımları önermektedir.

MASAM standart süreci 4 alt aşamadan oluşmaktadır. Bu adımlar aşağıdaki gibi listelenebilir:

- Geliştirme hazırlık aşaması
- Somutlaştırma aşaması
- Ürün geliştirme aşaması
- Ticarileştirme aşaması

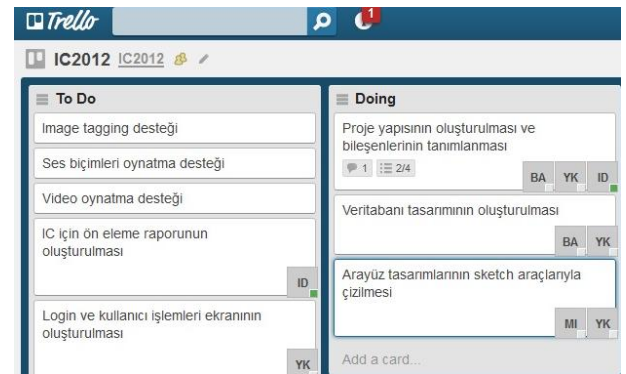
3. Uygulama

Çevik yazılım proje geliştirme metodolojilerini uygulamayı kolaylaştırmak için çeşitli araçlar geliştirilmiştir. Bu araçlardan bazıları aşağıdaki gibi listelenebilir:

- Trello [7]
- Telerik TeamPulse [8]
- Microsoft Team Foundation Server [9]
- Eclipse EPF [6]
- Version One [10]

Çalışma kapsamında bu araçlar incelenmiş ve inceleme sonucunda bazı araçların basit arayüz ve az bileşenle bu işlemleri yaparken bazı araçların ise çok kapsamlı işlevler içeren daha karmaşık bir ara yüze sahip olduğu görülmüştür.

3.1. Trello



Şekil 5: Trello Aracı İle Çevik Yazılım Proje Yönetimi

Trello, çevik yazılım geliştirme metodu ile geliştirilen tüm projelerin, tüm görevlerin, tüm takımın iyi organize edilebileceği ve her üyenin ayrı hesap ve yetkilere sahip olduğu kolay kullanıma sahip olduğu bir araçtır. Yeni kullanıcıların

kolayca anlayabileceği liste ve kart özelliklerine ve dakikalar içinde program üzerinde çalışmaya başlayabilecekleri basit arayüze sahiptir.

- Sahneler: Sahne listelerin tamamını içeren bölümdür. Genellikle her bir proje için ayrı sahneler kullanılmaktadır. Sahnelerin kullanımı çok kolaydır ve sahneler kullanılarak proje durumu kolayca yönetilebilir.
- Ortak Sahneler: Müşterilere yol haritası ve o anda sahip olunan özellikler hakkında bilgi vermek amacı ile kullanılır.
- Listeler: Listeler yalnızca basit listelerden oluşabileceği gibi süreçteki bir aşamayı gösterecek şekilde daha güçlü halde de kullanılabilirler. Listelerin basitçe sürüklemesi yöntemi ile iş akışına yeni bir görünüm kazandırılabilmesi gibi liste üzerindeki bir kart başka bir listeye taşınarak süreç içindeki ilerleme ortaya konulabilir.
- Kartlar: Kartlar görevlere karşılık gelmektedir. Takımın yapmaya ihtiyaç duyduğu şeyleri izlemek için kart oluşturabileceğiniz gibi kartlar dosya ekleme, video gömme, kullanıcı atama, tarih ekleme, kontrol listesi ekleme veya yalnızca yapılacak işleri gösterecek şekilde de kullanılabilir.
- Üyeler ve Organizasyonlar: Kullanıcılar sahneler üzerinde yetkisi kapsamında kontrole sahiptir. Tüm şirket ve takım göreceği şekilde sahneler oluşturulabileceği gibi sahneler yalnızca bazı kişilerin görebileceği şekilde de özelleştirilebilir. Tüm şirket kullanıcılarının tek davetle uygulamaya dahil edilebilmesine imkan vermektedir.
- Bildirimler: Trello, kullanıcılara yorum aldıklarında, atandıkları kartlar üzerinde herhangi bir değişiklik olduğunda veya başka bir sahne için davet gönderdiklerinde bilgi güncellemeler sütunu ile anlık olarak neler olduğunu bildirimlerle haber verir.
- Tüm Cihazlar Üzerinde Çalışma: Trello, akıllı telefonlar, tabletler ve dizüstü bilgisayarlar gibi farklı araçlar üzerinde çalıştırılabilir.

4. Mobil Uygulama Geliştirmede Çevik Yöntemlerin Farkları

Çevik yazılım geliştirme yöntemi uygulandığında ortaya çıkan süreçleri ve bu süreçlerin sonuçlarını inceleyebilmek için örnek çalışmalar gerçekleştirilmiş ve bu örnek olayların sonuçları geleneksel plan tabanlı yazılım geliştirme yöntemleri ile kıyaslanmıştır [3].

Mobil uygulama geliştirmede geleneksel plan tabanlı yazılım geliştirme yöntemleri yerine çevik yazılım geliştirme yöntemleri uygulandığında görünen olumlu etkiler aşağıdaki gibi listelenebilir:

1. Süreçteki görünürlüğün artması
2. Teknik sorunlara erken tanı
3. Sorumluluk paylaşımı
4. Verimli bilgi paylaşımı
5. Düşük hata yoğunluğu
6. Sürekli geliştirme temposu

Çevik yazılım geliştirme yöntemleri uygulanırken karşılaşılan ve gelecek araştırmalara yön verecek zorluklar [4]:

- Platforma özgü geliştirme yeteneği gereksinimi

- Mobil cihazlardaki çeşitliliğin getirdiği taşınabilirlik sorunu
- Mobil uygulamalar için test tabanlı geliştirme araçlarının olmaması
- Sık yayın çevrimine alışma
- Çevik metodların değiştirilmeden kullanılmasının yarattığı sorunlar

5. Sonuçlar

Bu çalışmada mobil uygulama geliştirmede çevik yazılım geliştirme metodolojisinin uygulanabilmesi için önerilen modeller ve mobil uygulama geliştirme platformlarında kapalı platformlardan açık platformlara geçiş ile başlayan yüksek uygulama kullanımı ve yüksek uygulama üretimi için en uygun metod olan çevik yazılım geliştirme 4 farklı modelde incelenmiş ve bu modellerin içeriği ve izledikleri adımlar ortaya konmuştur. Bu adımlara bağlı kalarak çevik yazılım geliştirme yöntemini uygulamayı kolaylaştıracak programlar test edilmiş ve bu programların da yardımı ile diğer yöntemlere göre daha verimli sonuçlar elde edildiği görülmüştür.

Mobile-D diğerlerinden önce geliştirilen bir model olmakla birlikte diğer modeller için referans olmuş ve bu modellerin Mobil-D'nin ayrıntılı analizi yapıp belli özellikleri geliştirilerek üretildiği ve bu modeller için test ve erişim ortamlarının oluşturulmuş olduğu gözlemlenmiştir.

6. Kaynaklar

- [1] Çevik Yazılım Geliştirme Manifestosu, <http://agilemanifesto.org/iso/tr/>
- [2] Christelle Scharff and Ravi Verma. 2010. Scrum to support mobile application development projects in a just-in-time learning context. In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering* (CHASE '10). ACM, New York, NY, USA, 25-31. DOI=10.1145/1833310.1833315
- [3] Rahimian, V.; Ramsin, R.; , "Designing an Agile methodology for mobile software development: A hybrid method engineering approach," *Research Challenges in Information Science*, 2008. RCIS 2008. *Second International Conference on* , vol., no., pp.337-342, 3-6 June 2008doi: 10.1109/RCIS.2008.4632123
- [4] Pekka Abrahamsson, Antti Hanhineva, Hanna Hulkko, Tuomas Ihme, Juho Jääfinoja, Mikko Korkala, Juha Koskela, Pekka Kyllönen, and Outi Salo. 2004. Mobile-D: an Agile approach for mobile application development. In *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications* (OOPSLA '04). ACM, New York, NY, USA, 174-175. DOI=10.1145/1028664.1028736
- [5] Yang-Jae Jeong; Ji-Hyeon Lee; Gyu-Sang Shin; , "Development Process of Mobile Application SW Based on Agile Methodology," *Advanced Communication Technology*, 2008. ICACT 2008. *10th International Conference on* , vol.1, no., pp.362-366, 17-20 Feb. 2008doi: 10.1109/ICACT.2008.4493779.
- [6] Eclipse Process Framework Project (EPF), <http://www.eclipse.org/epf/>
- [7] Trello, <https://trello.com/>
- [8] Telerik TeamPulse, <http://www.telerik.com/agile-project-management-tools/>

- [9] Microsoft Team Foundation Server,
<http://www.microsoft.com/visualstudio/tur/products/visual-studio-team-foundation-server-2012>
- [10] Version One, <http://www.versionone.com/>