

Uygulama Yaşam Döngüsü Yönetim Araçlarının Verimini Arttırmak İçin Ciddi Oyun Uygulaması

Çağdaş ÜSFEKES¹, Murat YILMAZ², Eray TÜZÜN³

^{1,3} HAVELSAN A.Ş., Ankara, Türkiye
{cusfekes, etuzun}@havelsan.com.tr

² Çankaya Üniversitesi, Ankara Türkiye
myilmaz@cankaya.edu.tr

Özet. Uygulama yaşam döngüsü yönetimi (UYU), bir araç takımı yardımıyla yazılım ekosisteminde, gereksinim analizinden ürünün dağıtım aşamasına kadar olan tüm aktivitelerin daha fazla üretkenlik sağlayarak (hızlı ve güvenilir bir şekilde) yürütülmesini hedefler. Bu üretkenlik, yazılım geliştiricilerin kendilerine verilen görevleri motivasyon kaybetmeden yapabilmesi ile mümkün olabilmektedir. Dolayısıyla yazılım geliştiricilerin üretkenliklerinin artırımı kendilerine en uygun olduğunu düşündükleri işlerin seçim sorumluluğunu alarak sağlanabilmektedir. Makale kapsamında, bu problemin çözümü için algoritmik oyun kuramı kullanılarak bir market ve ödül mekanizması tasarlanması ve ciddi oyun kavramı yardımı ile modelin yazılım uygulamasının gerçekleştirilmesi ile yapılan ön çalışmalar anlatılmıştır.

Anahtar Kelimeler: Uygulama yaşam döngüsü yönetimi, ciddi oyun uygulamaları, ödüllendirme mekanizmaları, yazılımda verimlilik, oyunlaştırma

Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools

Çağdaş ÜSFEKES¹, Murat YILMAZ², Eray TÜZÜN³

^{1,3} HAVELSAN A.Ş., Ankara, Turkey
{cusfekes, etuzun}@havelsan.com.tr

² Çankaya University, Ankara Turkey
myilmaz@cankaya.edu.tr

Abstract. Application lifecycle management aims to provide higher productivity using a set of tools for software development processes. It supports all processes from requirement analysis to deployment. This productivity is possible with software developers when they perform their tasks without losing motivation. If software developers can choose the optimum tasks, their productivity can increase. In this article an auction market and reward mechanism is developed by using algorithmic game theory and the studies on development software application.

Keywords: Application Lifecycle Management, serious game applications, reward mechanisms, software engineering productivity, gamification

1 Giriş

Günümüzde yazılım sektöründen çıkan ürün veya fikirler farklı sektörlere de fayda sağlamaktadır. Bu sebeple uygulama yazılım geliştirme süreci sadece yazılım geliştirme temelli firmaların değil tüm firmaların başlıca işlevlerinden biri olmuştur.

Uygulama yaşam döngüsü yönetimi (UYU) kuramı günümüze kadar iki versiyon üzerinden evrilmiştir. Bu versiyonlardan ve versiyonlar arası farklardan makalenin ileri aşamalarında detaylı bahsedilecektir. UYU araçlarındaki gelişmeler ile birlikte bu araçların proje geliştirme süreçlerindeki önemi artmaktadır. Bir projenin fikir aşamasından, teslimat ve bakım aşamasına kadar geçen tüm süreçlerde UYU araçlarının rolü büyüktür. Bu sebeple UYU araçlarından alınan verimin artması projeden elde edilecek olan ürünün kalitesinin artmasını sağlayacak ve proje maliyetleri ile birlikte zamandan tasarruf edilmesini sağlayacaktır.

UYU araçlarından alınan verimi artırmanın en önemli yolu bu araçları kullanan farklı alandan çalışanların (yazılım mühendisleri, test mühendisleri, konfigürasyon yöneticileri, proje yöneticileri vb.) UYU süreçleri üzerindeki bilgi, deneyim ve tecrübelerini artırmaktır. Çalışanların UYU araçları üzerindeki bilgi ve tecrübelerini artırmak için çeşitli eğitimler, seminerler, kurslar vb. aktiviteler düzenlenebilir. Bu eğitimler için çalışanlar mesai zamanlarından vakit ayıracak ve eğitim sonunda ilgili araçlar üzerinde çalışırken tecrübe kazanacaktır. Bu noktada oyunlaştırma ve ciddi oyun uygulamaları kullanarak çalışanlar ayrı bir eğitim almadan bir yandan işlerini yaparken bir yandan da oyun uygulamaları ile sıkılmadan tecrübe ve yetkinliklerini artıracaklardır. Bu çalışmada ciddi oyun uygulaması kullanarak UYU araçlarından alınan verimin artırılmasından bahsedilecektir.

Bildirinin ikinci bölümünde UYU ve ilgili çalışmalar, üçüncü bölümde, oyunlaştırma konusunda literatürdeki çalışmalar özetlenmiştir. Dördüncü bölümde, Yazılım Yaşam Yönetimi için oyunlaştırma tabanlı kaynak dağıtım çerçevesi önerilmiş, beşinci bölümde, UYU Süreçleri ile Entegre Ciddi Oyun Uygulaması Geliştirimi anlatılmıştır. Bu uygulama bir UYU aracı olan TFS (Team Foundation Server) ile entegre çalışmaktadır ve buradan aldığı iş öğelerini açık artırma olarak kullanıcıların teklifine sunmaktadır. Kullanıcılardan alınan teklifler kredi, yapılabirlik gibi parametreler ile değerlendirilerek işlerin kullanıcılara dağıtılması sağlanır. Uygulamaya ilişkin teknik detaylar ilgili başlıkta anlatılmıştır. Son bölümde ise sonuçlar ve gelecek çalışmalar sunulmuştur.

2 Uygulama Yaşam Döngüsü

Uygulama Yaşam Döngüsü (UYU) bir yazılım projesi çıktısının fikir aşamasından geliştirme, dağıtım ve bakım süreçlerinin hepsini kapsayan bir yaklaşımdır [1]. Bu sayede yazılım geliştirme süresi boyunca güçlü bir raporlama yeteneği ile izlenebilirlik yeteneği sağlar [2]. UYU kavramı UYU 1.0 sürüm adı ile ortaya çıkmıştır. UYU 1.0 her disiplin için farklı araçların kullanımını ve tüm bu araçların kendi aralarında uyumlu

çalışması çözümünü önerir. Geliştirme, test, inşa, tasarım gibi farklı UYY süreçlerini kapsayan farklı araçlar diğer süreçleri yöneten araçlarla karşılıklı bilgi alışverişini yapabilir ve birlikte çalışabilir [3]. UYY 1.0 çözümünde farklı firmalar tarafından üretilen farklı araçların birbirleri ile entegrasyonlarında yaşanan problem ve eksiklikler sebebiyle UYY 2.0 çözümü doğmuştur [3,4]. UYY 2.0 yaklaşımında yazılım geliştirme süreçlerinin farklı araçlardan ve süreçlerden farklı alanların takip edilmesi yerine tüm bu süreçlerin tek bir araç altında toplanması benimsenmiştir [5,6]. UYY 2.0 çözümüne uygun araçlar tüm bu süreç boyunca rol ve yetki tabanlı bir yaklaşım izler. UYY 2.0 sürümünde süreçler arası veri eşleşmesi düzenli olarak otomatik olarak sağlandığından dolayı UYY 1.0 çözümündeki veri eşleme maliyetleri sıfırlanmıştır. Bu sayede ürün çıktısı üretimi, yazılım geliştirme yönetimi ve raporlama daha hızlı bir şekilde yapılmaktadır [7,8]. UYY 2.0 çözümünün sağladığı bu avantajlardan sonra büyük çaplı yazılım şirketinde de süreçlerin yönetimi gün geçtikçe UYY üzerinden yapılmaya başlanmıştır. Bu yazılım şirketlerinde bir projenin başlangıcından müşteriye teslimatına kadarki süreçte yazılım geliştirme mühendisleri, test mühendisleri, konfigürasyon yöneticileri, proje yöneticileri gibi pek çok farklı görevde bulunan çalışanlar çeşitli UYY (Uygulama Yaşam Döngüsü) araçları kullanır. Bu araçlar yazılım geliştirme süreçlerinde kullanılan modülleri tek bir uygulama altında kullanıcılara sunar [5]. UYY yaklaşımı gereksinim yönetimi, test yönetimi, inşa yönetimi, kaynak kod yönetimi, proje yönetimi ve risk yönetimi süreçlerinin tümünü kapsar.

Bütün bu farklı süreçleri birbiri ile entegre şekilde yürütebilmek bir yazılım projesinin başarılı şekilde sonuçlanmasında ve kalite seviyesinin belirlenmesinde büyük önem taşır.

3 Oyunlaştırma

Oyunlar, bireylerin, takımların ve kurumların stratejik etkileşimlerinin modellenmesi üzerine kuruludur. Tarihte araştırmalar, bireylerin etkileşimleriyle ilişkili olan oyunlara, analitik metotlara ve karar verme mekanizmalarına yardımcı olan araçlara odaklanmıştır [9]. 1960'lerde yapılan araştırmalar ve çalışmalar sosyal bilimler ile oyun kuramı arasındaki ilişkiyi başarılı bir şekilde ortaya çıkarmıştır. Özellikle son yıllarda oyunlaştırma ekonomik analiz alanında yaygınlaşmıştır [47]. Bu süreçte şirketler oyun kuramını çalışanlarının karakter analizini yapmak amacıyla da kullanmaktadır.

Oyun kavramı üzerine kurulan analizler, bireyler arası etkileşimleri inceleyen çeşitli analitik araçlar barındırır. Bu analizler etkileşen öğelerin (bireyler, şirketler, uluslar vb.) arasındaki etkileşimleri oyun düzeyinde inceler. Bununla birlikte, oyun kuramı zorunlu kararları, davranışları ve sonuç olarak katılımcılar arasındaki çelişki içeren etkileşimleri de keşfetmek için kullanılabilir. Oyun kuramının önemli kabülllerinden bir tanesi de tüm oyuncuların rasyonel olduğudur. Diğer bir deyişle, tüm oyuncular oyun içindeki kurallara uyarlar ve oyunu kazanmaya çalışırlar. Son on yılda oyun kuramı ekonomi dışında psikoloji, biyoloji, dil bilimi ve bilgisayar bilimi alanlarında da kullanılmaya başlanmıştır [10].

Oyun kuramı genel olarak iki bölüme ayrılabilir. Bunlar “Rekabetçi oyun kuramı” ve “İşbirlikçi oyun kuramı” olarak isimlendirilir [10]. Rekabetçi oyun kuramında seçimler oyun içindeki her bireysel oyuncunun çıkarlarını etkiler. Bu oyuncular kendi şahsi çıkarlarını düşünerek hareket ettikleri zaman daha kazançlı olurlar. 1950 yılında

John Nash “Nash Dengesi” adı verilen kavram geliştirmiştir. Nash dengesi oyuncuların stratejik seçimlerinin bir türüne verilen isimdir [11]. İşbirlikçi oyun kuramı rekabetçi oyun kuramından farklıdır. İşbirlikçi oyun kuramı bireylerin bir araya gelerek oynamalarına imkanı sağlar. Bu teoriye göre oyuncular kendi aralarında çıkarlarını göz önünde bulundurarak gruplara ayrılırlar. Bu teori yazılım geliştirme projelerinde kullanılabilir.

Son yıllarda oyun kuramının yazılım geliştirme mühendisliğinde kullanıldığı pek çok örnek bulunmaktadır [12, 13]. Bu çalışmalardan göreceğimiz üzere oyun kuramı yazılım geliştiriciler tarafından tanımlanmış veya potansiyel pek çok problemin çözümünde kullanılabilir. Lagesse [14] görev atama problemleri üzerine bir oyun geliştirmiştir. Bir diğer çalışmada Cockburn [15] yazılım geliştirmeyi bir çeşit oyun olarak kabul etmiş ve bu oyunun sınırlı proje kaynakları üzerinde iletişim ve koordinasyon yeteneklerinden ibaret olduğunu iddia etmiştir. Baskerville ise farklı bir çalışmada [16] hızlı kaynak kullanımına dayanan yüksek hızlı internet gelişimini incelemiştir. Sullivan [17] yazılım tasarım kararlarının değerlendirilmesi için ekonomik bir yaklaşım sunan çalışma sunmuştur. Bir diğer çalışmada Szawal ve Sudan [18] yazılım tasarım gelişimini şekillendirmek için karar vermeyi modelleme yapısına oyun kuramını dahil etmişlerdir. Bu çalışmada Szawal ve Sudan yazılım tasarım evrimi oyunu adlı, geliştirici ve müşteri tarafından oynanabilen bir yaklaşım kurgulamıştır. Ek olarak, yazılım geliştirme takımlarına durum analizi yapmak için hafif oyun teorik analiz tekniği kullanmalarını önermişlerdir.

Gao [19] yaptığı çalışmada karar hataları ile algılama yazılımlarının ilişkilendirmesine yönelik bir oyun geliştirmiştir. Gao-hui [20] tarafından yapılan bir başka çalışmada kurumsal şirketlerin yazılım proje geliştirmelerinin oyun kuramına dayandırılması ile ilgili analizden bahsedilmiştir. Soska [21] ve diğerleri tarafından akademik hayattaki öğrencileri baz alan bir çalışma yapılmıştır. Bu çalışmada öğrencilerin yazılım testi konusunda daha detaylı ve derinlemesine bilgi sahibi olmaları amacıyla bir kart oyunu geliştirilmiştir. Oscar Pedreira ve diğerleri tarafından yapılan çalışmada [22] yazılım mühendisliğinde oyunlaştırma kullanımı üzerine bir haritalandırma yapılmıştır. Bu sayede gelecekteki araştırmalara yönelik açık fırsatlar tanımlanmaya çalışılmıştır. Oyunlaştırmanın son yıllarda yazılım mühendisliği gibi akademik programlara, konferanslara ve akademik dergilere konu olması ile ilgili Sweedyk [23] bir çalışma yapmıştır. Kitagawa ve diğerleri tarafından bir çalışmada [24] ise yazılım geliştirme sürecinde yazılımın kalitesine etkisi büyük olan ve çıkabilecek hataları minimuma indirmeyi amaçlayan kod gözden geçirme üzerinde bir oyun kuramı modellenmiştir. Bir başka çalışmada Szabo [25] iş simülasyonu oyunu olarak bilinen “Game Dev Tycoon” adlı oyunu akademik alanda öğrencilerin yazılım mühendisliği öğrenimine uyarlamıştır.

Yazılım süreçlerinin iyileştirilmesi için kullanılan yöntemler, yazılım ürünü kalitesini artırmayı sağlayan tüm aktiviteleri kapsamalıdır [26, 27]. Bu aktiviteler yazılım geliştirmeye etkileri bakımından yeniden değerlendirilmelidir (yazılım geliştirmedeki insan faktörü, sosyal iletişim problemleri vb.). Yazılım geliştirme süreçleri genellikle bireysellikten ziyade takım olarak çalışıldığından, tüm bu çalışmalar birer sosyal faaliyet olarak kabul edilebilir [28]. Bu takımlardaki çalışanlar çalışma koşulları, karakterleri, rasyonellikleri ve dayanışma gibi özellikleri ile tanımlanırlar [29]. Yazılım süreç iyileştirme mekanizmalarının önemli amaçlarından bir tanesi de yazılım projelerinin kalitesini maksimum seviyede tutmak için proje

planına ve proje bütçesine uygun bir yol çizmektir. Buna göre yazılım geliştirme süreci boyunca bakım ve yönetim önemlidir. Örneğin projeler büyüdükçe sadece kaynak kodun okunabilirliği değil teknik dokümantasyonların da okunabilirliği azalmaktadır. Projelerin büyüklüğü arttıkça birbiri ile uyumlu çalışması gereken yazılım birimleri ve çalışan sayısı da artmaktadır. Yazılım geliştirme sürecinde bu uyum ile ortaya çıkan ürünün kalitesi doğru orantılıdır. Bu noktada yazılım geliştirme sırasında ortaya çıkan hatalar doğru kişilere atanabilir.

Yazılım geliştirmeyi sistematik hale getirmek için oyun benzeri aktiviteler tasarlamak gerektiği pek çok kanıt ile belirlenmiştir [30, 31, 32]. Özellikle, Yılmaz [32] yazılım geliştirmede takımların karakter özelliklerinin etkisini ortaya koyan oyun benzeri bir yaklaşımda bulunmuştur. Yılmaz ve diğerleri [31] yazılım geliştirme süreçlerini iyileştirmek için bir oyunlaştırma yaklaşımı üzerinde çalışmıştır. Yılmaz [30] tarafından tanımlanan ve yazılım geliştirme için ekonomik mekanizma tasarımı yaratma fikri ilgili alanda yapılan ilk çalışmalardan biri olmuştur. Bir çalışmada Yılmaz ve diğerleri [33] yazılım geliştirme süreçlerini iyileştirmek için bir ekonomik mekanizma geliştirilmesi üzerine çalışmışlardır. Yılmaz ve O'Connor oyunlaştırma kullanarak yazılım geliştirme süreçlerini iyileştirmeyi amaçlayan tamamlayıcı bir yaklaşımda bulunmuştur [34]. Bir diğer çalışmada Yılmaz ve O'Connor yazılım geliştirmeyi bütünüyle ekonomik bir aktivite olarak görmüş ve yazılım geliştirme sürecindeki görev atama problemlerine piyasa bazlı bir yaklaşım geliştirmiştir [35]. Bu çalışmalara genel olarak bakıldığında oyun benzeri yaklaşımların yazılım geliştirme süreçlerindeki verimliliğin artırılmasında önemli olduğu görülmektedir.

3.1 Ödüllendirme Mekanizmaları

Ödüllendirme mekanizmaları oyun kuramına bağlı bir alt araştırma alanıdır. Bu alan oyun tasarımında büyük rol oynar. Bir oyun içinde her oyuncu bireysel ve takım olarak başarılı olmak ve ödül kazanmak için çaba sarf eder. Ödüllendirme mekanizmaları, her türlü alan için geliştirilmiş oyun mekanizmaları olarak kullanılabilir. Houk [36] tıp alanında sinir ağları ile oyun kuramını birleştiren bir çalışma yapmıştır. Singh [37], insan bilgisayar etkileşimi üzerine, çevrim içi öğrenme sistemi ile ödüllendirme mekanizmalarını kullanmıştır. Lua [38] ise P2P sistemleri üzerine geliştirilmiş bir ödüllendirme mekanizması üzerinde çalışmıştır. Wang ve Sun [39] bilgisayar oyunları üzerindeki ödüllendirme mekanizmaları üzerine bir araştırma yapmışlardır.

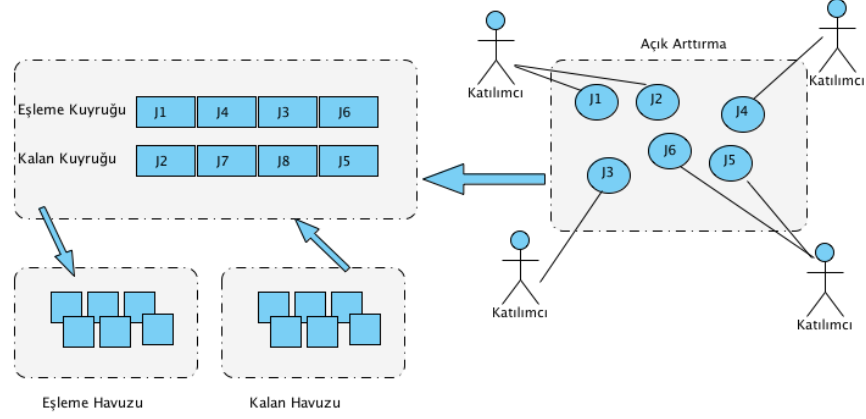
Ödüllendirme mekanizmaları, insan-bilgisayar etkileşimi ve insan öğrenmesi üzerinde etkili bir araçtır. Eğer ödüllendirme mekanizması doğru bir oyun kuramı üzerine inşa edilirse, herhangi bir alan için geliştirilmiş olan oyun kurgusu da o kadar verimli olur. Oyunlaştırma teorileri insanın bir sistemi öğrenirken ortaya çıkabilecek zorlukları daha kolay ve eğlenceli yoldan çözmeye yardımcı olur. Walz [40] çalışmasında oyunlaştırma teorilerini toplumların sosyal ve kültürel temellerine dayanan bir sistem olarak tanımlamıştır. Gonzales [41] bilgisayar mühendisliğinde öğretme ve öğrenme sürecinde oyunlaştırma teknikleri ve oyun mekanizmalarının kullanımından bahsetmiştir. Bunlara ek olarak, Qu [42] yazılım mühendisliğinin öğretimi ile ilgili araştırmalar yapmış ve bu süreçte oyunlaştırma mekanizmalarının kullanımını incelemiştir. Largo [43] çalışmasında pek çok öğrenciden gelen geri dönüşlere göre öğrenme sürecinde oyunlaştırma kullanımını incelemiştir.

Büyük çaplı kurumsal şirketlerde çalışanlar daha kompleks sistemler kullanmak durumunda kalabilir. Bunlar mühendislik yönetim araçları, finansal otomasyon araçları gibi pek çok çeşide sahip olabilir. Bu sistemleri verimli bir şekilde kullanmak ise bilgi birikimi ve deneyim gerektirir. Bu noktada çalışanlar bu sistemleri verimli kullanabilmek için gerektiğinde normalden çok daha fazla efor ve iş gücü sarf ederler. Bu süreçte oyunlaştırma kuramı kullanımı çalışanların öğrenme sürecini hızlandırır. Örneğin mühendislik alanı üzerinde Pariza yazılım testleri ve kaynak arasındaki izlenebilirlik üzerine bir oyun kuramı geliştirmiştir [44]. Bir diğer çalışmada ise test ile kod çıktıları arası izlenebilirlik için oyunlaştırma çalışması yapılmıştır [45].

Ödüllendirme mekanizmaları üzerine yapılan bu çalışmaları inceledikten sonra UYY ile oyunlaştırma kuramının harmanlanacağı bir çerçeve (framework) üzerine çalışmaya karar verdik ve ismi “Aykun” olan yazılım yaşam yönetimi için oyunlaştırma tabanlı kaynak dağıtım çerçevesi geliştirdik. Bir sonraki başlıkta bu model hakkında detaylı bilgi verilecektir.

4 Yazılım Yaşam Yönetimi için Oyunlaştırma Tabanlı Kaynak Dağıtım Çerçevesi

UYY, yazılım geliştiricilere iş ataması yapılırken kullanıcıların farklı tercihlerine göre daha verimli olmalarını sağlayacak bir sistem önermemektedir. Aykun, tüm UYY süreçlerinin oyun kuramı üzerinden yönetilmesini ve veriminin artırılmasını sağlayan ciddi oyun temeline dayanan bir modeldir. Bu model, yazılım üretkenliğinin iyileştirilebilmesi için geliştiricilere kaynak atama işini bireylerin tercihleri doğrultusunda yapılmasını hedeflemektedir. Kullanıcılar, yazılım sistemi içinde bulunan gereksinim analizinden yazılım testlerine kadar bir çok alanda tanımlanacak olan yüzlerce küçük ölçekli açık arttırmaya katılabilecektir. Dolayısıyla, katılımcılar belli kısıtlar ve kendi tercihleri doğrultusunda tanımlanan işlere talip olarak kaynak ayırımı sağlanması yoluyla, kendilerini en çok motive eden işleri seçme hakkına sahip olacaktır. Aykun, yazılım geliştiricilerin önceliklerini ve iş seçimlerini belirlemek için düşünülmüş genel bir kaynak yönetimi çerçevesidir. Sistem kaynakların en verimli şekilde iş paylaşımını ve zamanlamasını yapabilir. Şekil 1’de sistemin genel yapısı görülebilir. Açık artırma havuzunda tanımlanan açık artırmalara katılımcılar teklif verir ve sistem ilgili işleri katılımcılar ile eşler.



Şekil 1. Yazılım Yaşam Yönetimi için Açık Artırmaya Dayalı Kaynak Dağıtım Modeli

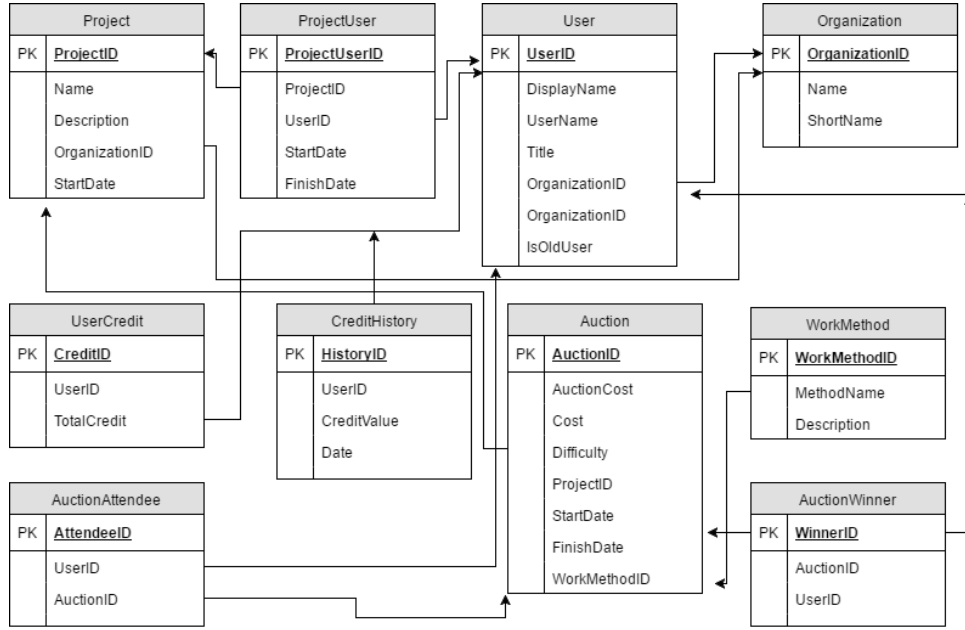
Aygun, kullanıcılar adına tanımlanmış işleri yazılım geliştiricilere duyurur. Kişiler, tasarlanmış açık artırma mekanizması yardımıyla ödedikleri kredi miktarına bağlı olarak belli işler için istekte bulunurlar. Bu istekler doğrultusunda sistem uygun adaylar arasından işi en çok isteyen bu işi en çok yapmak isteyen kişi olarak kabul eder. Bu sayede işlerin ve kaynakların arasında bir değer biçimi gerçekleşir. Sistem, bir kullanıcının kendi bütçesine göre teklif vermek zorunda kalmasını ve fiyatın zaman içinde sabit kalmasını sağlar. Sistem kullanıcılara yaptıkları işlerden dolayı puan, rozet ve lider tablosu desteğiyle birlikte motivasyon artırıcı bir fayda fonksiyonu oluşturur. Kişiler kazanımları ve başarıları doğrultusunda sistem tarafından ödüllendirilir.

Bu tekniği kullanarak bir takımdaki bireyler yapmak istedikleri iş ile ilgili mevcut kredilerini kullanarak teklif verebilirler. Bu model ile yazılım geliştirme takımlarındaki bireyler arasında verimlilik ve bilgi çeşitliliği açısından ilginç sonuçlar alınabilir. Bu noktada tüm bireylerin kendi kredi tutarları doğrultusunda teklif verme hakkı bulunur. Bu yöntem yazılım şirketlerinde risk azaltma amaçlı olarak kullanılabilir. Daha iyi açıklamak için golf handikapı isimli metaforu inceleyebiliriz. Amatör golf yarışmalarında kişiler yarışmaya bireysel olarak katılır fakat yarışma sonuçları katılımcıların yetenek değerlerine göre belirlenir. Böylece tüm katılımcılar yarışmayı kazanma yetisine göre yarışır ve herkes kişisel performansını geliştirmek için çalışır.

5 UYY Süreçleri ile Entegre Ciddi Oyun Uygulaması Geliştirimi

Bu bölümde HAVELSAN'da geliştirilmekte olan projeler üzerinde kurgulanacak olan ciddi oyun uygulamasından bahsedilecektir. Projelerin geliştirilme aşamasında farklı UYY (ALM) araçları kullanılabilir. HAVELSAN'da 2012 yılından bu yana Microsoft ürünü olan Team Foundation Server (TFS) kullanılmaktadır [46]. Bu ciddi oyun uygulaması da veri beslemesini TFS üzerinden RESTful servisleri kullanarak yapacaktır. TFS'te tanımlı olan HAVELSAN projeleri bu sistemde de tanımlı olacak ve TFS üzerindeki iş öğeleri üzerinden açık artırmalar yapılacaktır.

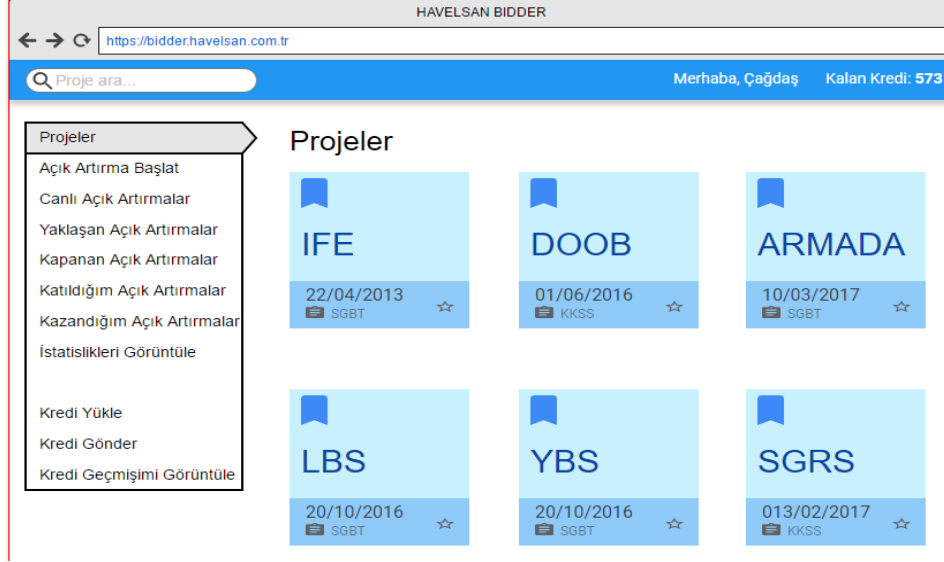
Ciddi oyun uygulaması web tabanlı uygulama geliştirme teknolojileri kullanarak geliştirilmiştir. Web arayüzü geliştirilmesinde ASP.NET / MVC, JavaScript, JQuery, Bootstrap gibi popüler teknolojiler ile sunucu tarafında nesne tabanlı C# programlama dili kullanılmış olup veritabanı seviyesinde Microsoft SQL Server 2012 teknolojisinden faydalanılmıştır. ASP.NET / MVC ile SQL Server veritabanı iletişimi için üçüncü parti herhangi bir uygulamaya veya araca ihtiyaç olmadığı için SQL Server tercih edilmiştir. Ciddi oyun uygulamasına ait veritabanı tablo yapısı ve tablolar arası ilişkiler Şekil 2’de görülmektedir.



Şekil 2. Ciddi Oyun Uygulaması Veritabanı Şeması

Kullanıcılar sisteme HAVELSAN kullanıcı adı ve şifreleri ile giriş yapabilecektir. HAVELSAN’da bu ciddi oyun uygulamasını kullanacak kullanıcılar (sistem mühendisleri, yazılım mühendisleri, test mühendisleri, proje yöneticileri vb.) farklı genel müdür yardımcılığı altında bulunan bir veya birden çok projede görev alır. Oyuncular bu projelerdeki yetki seviyelerine göre sisteme giriş yaptığında erişim sağlayabildikleri projeler ile bu projeler üzerinden tanımlanmış olan açık artırmaları görebilir. Oyuncu dilerse yeni bir açık artırmaya katılabilir ve önceden katıldığı ve kapanan açık artırmaları listeleyerek bunlardaki genel başarı durumunu grafiksel olarak görerek diğer oyuncular arasındaki sıralamasını takip edebilir.

Kullanıcı sisteme giriş yaptığında ilgili açık artırmalarına katılabileceği projelerin listesi şekil 3’teki gibi görünür.



Şekil 3. Proje Listeleme Ekranı

Ekranın sol üst köşesinden proje adına göre arama yapılarak istenilen proje bulunabilir. Proje arama alanının sağında ise kullanıcı anlık olarak kalan kredi miktarını görebilir.

Proje listesinin sol bölümünde kullanıcıların yetkileri dahilinde yapabilecekleri işlemler menü olarak listelenmiştir. Bu menü iki ana gruba ayrılır. Bunlar açık artırma işlemleri ve kredi işlemleridir. Açık artırma işlemlerinde kullanıcı eğer yetkisi var ise bir proje üzerinden yeni açık artırma başlatabilir. Bunun haricinde canlı açık artırmalar, yaklaşan açık artırmalar, kapanan açık artırmalar ile kullanıcının katıldığı ve kazandığı açık artırmalar da ayrı ayrı görülebilir. Kullanıcı, “İstatistikleri Görüntüle” menüsünden yine yetkileri dahilinde katıldığı açık artırmaların veya tüm açık artırmaların sayısal istatistiklerini grafiksel olarak görebilir. Kredi işlemlerinde ise kullanıcı ihtiyacı olan olan krediyi limiti dahilinde yükleyebilir veya başka bir kullanıcıya kredi gönderimi yapabilir. Her kullanıcı sadece kendine ait olan kredi kazanma, harcama, gönderme gibi tarihçe bilgilerini de zamana göre sıralı olarak görüntüleyebilir.

Kullanıcı, proje listesindeki projelerden birini seçerek o projeye ait açık artırmaları listeleyebilir (Şekil 4).

HAVELSAN BIDDER

https://bidder.havelsan.com.tr

Açık artırma ara... Merhaba, Çağdaş Kalan Kredi: 573

Projeler

Açık Artırma Başlat

Canlı Açık Artırmalar

Yaklaşan Açık Artırmalar

Kapanan Açık Artırmalar

Katıldığım Açık Artırmalar

Kazandığım Açık Artırmalar

İstatistikleri Görüntüle

Kredi Yükle

Kredi Gönder

Kredi Geçmişimi Görüntüle

'YBS' Projesine Ait Açık Artırmalar

3 açık artırma devam ediyor...

- Yazılım Testi
Başlama Tarihi: 10/03/2017
- Stok Modül Yazımı
Başlama Tarihi: 10/10/2016
- Veritabanı Tasarımı
June 12, 2015

Detaylar: Stok Modül Yazımı

Başlama Tarihi	10/10/2016
Kalan Zaman	15 gün 06 saat
Açık Artırma Fiyatı	550 Kredi
Satın Alma Fiyatı	2000 Kredi
Zorluk Derecesi	<input type="range"/>
Katılım Sayısı	18 kişi

[Artırmaya Katıl](#)

[Başkasına Gönder](#)

[Artırmayı Durdur](#)

Şekil 4. Seçili Projenin Açık Artırma Listesi ve Detayları Ekranı

Kullanıcı seçili projeye ait açık artırmalardan istediğini seçtiğinde ekranın alt yarısında seçili açık artırmaya ait detay bilgileri listelenir. Açık artırmının başlama ve bitiş tarihi, fiyatı, katılan kişi sayısı ve zorluk derecesi (% cinsinden) gibi bilgiler bu alanda görülebilir. Kullanıcı kendi başlattığı açık artırmayı “Artırmayı Durdur” düğmesini kullanarak sonlandırabilir, “Başkasına Gönder” düğmesi ile farklı bir kullanıcıyı bu açık artırmadan haberdar edebilir ya da “Artırmaya Katıl” düğmesi ile açık artırmaya katılabilir.

6 Sonuç ve Gelecek Çalışmalar

Bu çalışmada UYY süreçleri ile oyun kuramı UYY araçları üzerinden birleştirilmiş böylece proje süreçlerinden alınan verimin artırılması amaçlanmıştır. Bu uzun vadeli çalışmanın ilk aşamasında UYY araçlarından TFS (Team Foundation Server) baz alınmış olup ileriki aşamalarda Jira gibi araçlarla ciddi oyun uygulamasının entegrasyonuna çalışılacaktır. Böylece HAVELSAN içinde daha çok projeye ve kullanıcıya ulaşılması amaçlanmaktadır. Proje ile elde edilmesi beklenen sonuçlar şu şekildedir:

- HAVELSAN bünyesinde kullanılmakta olan Geliştirme Ortamının (GO) üretkenliğinin artırılması.
- Oyun tasarım kavramlarının (puan, rozet, ün, vb.) kullanılarak katılımcıların yapmaları beklenen en uygun işleri seçmelerinin sağlanması ve dolayısıyla bu işleri en verimli şekilde üretmelerinin sağlanması.

- Bu seçimler ve sonuçlarının proje yaşam döngüsüne olan katkılarının incelenerek iyileştirmeler önerilmesi.
- Kullanıcı ve iş dağılımları üzerine kurgulanacak bir veri modeli yardımı ile işlerin verimli bir şekilde dağıtılması ve bu dağılımların bilimsel yöntemler yardımıyla ölçülmesi.
- Yazılım geliştiricilerin proje veya alt görevlerin yazılım proje yaşam döngüsündeki durumlarının ve bu durumların projeye olan etkilerinin görsel olarak gösterilmesi.

Proje ile elde edilecek kazanımlar ise şu şekildedir:

- HAVELSAN çalışanlarının UYY araçları konusundaki yetkinliğini en üst seviyeye çıkarmak
- HAVELSAN'ın oyun kavramı, oyunlaştırma ve ciddi oyun alanlarında yetkinliğinin artırılması
- HAVELSAN çalışanlarının üretkenliğinin ve motivasyonunun artırılması
- Bu proje ile yazılım geliştirme süreçlerinin uygulanış kalitesinin artırılması planlanmaktadır.
- Proje ile kullanım oranı yüksek olan UYY araçlarının veriminin artırılması ile HAVELSAN'a katkı yapılması amaçlanmaktadır.

Referanslar

1. D. Chappell, "What is Application Lifecycle Management?", Chappell & Associates, 2008.
2. Ahsen İkbâl Aytekin, Eray Tüzün, Yagup Macit, Bedir Tekinerdoğan, Uygulama Yaşam Döngüsü Yönetimi - Sistematiik Eşleme Çalışması, UYMS, 2015
3. Kääriäinen, J., Välimäki, A.: Impact of Application Lifecycle Management – A Case Study. In: International Conference on Interoperability of Enterprise, Software and Applications (I-ESA), Berlin, Germany, March 25-28, pp. 55–67, 2008
4. Schwaber, Carey, et al. "The Changing Face of Application Lifecycle Management" Forrester Research, August 18, 2006.
5. Shaw, K. (2007) "Application Lifecycle Management for the Enterprise", Serena Software, White Paper, http://www.serena.com/Docs/Repository/company/Serena_ALM_2.0_For_t.pdf
6. J. K. and A. Välimäki, "Applying ALM for the Development of Complex Systems : Experiences from the Automation Industry," in EuroSPI, 2009, pp. 149–160.
7. Kitchenham, B. A., Dyba, T., and Jorgensen, M., "Evidence-Based Software Engineering", Proc. Of the 26th International Conference on Software Engineering (ICSE '04), Scotland, UK, pp. 273-281.
8. Y. Macit, E. Tüzün, K. Ince and A.I. Aytekin, "Büyük Ölçekli Bir Organizasyonda Uygulama Yaşam Döngüsü Yönetimi Uygulama Deneyim", UYMS, 2014.
9. Yılmaz, M. and O'Connor, Rory, Oyun Kuramı Kullanarak Yazılım Takımlarının Üretkenliğini Artırmak İçin Geliştirilen Bir Yazılım Süreç Mühendisliği Yaklaşımı, UYMS, 2011
10. S. Zahran, Software Process Improvement: Practical Guidelines for Business Success. Addison Wesley, 1998.
11. Eric Maskin, Nash equilibrium and mechanism design, Institute for Advanced Study, Princeton University, United States, 2008

12. T. Dingsøy, T. Dybå, and N. B. Moe, *Agile Software Development: Current Research and Future Directions*, 1st ed. Springer, 2010.
13. F. P. Deek, J. A. McHugh, and O. M. Eljabiri, *Strategic software engineering: an interdisciplinary approach*. CRC Press, 2005.
14. B. Lagesse, "A Game-Theoretical model for task assignment in project management," in 2006 IEEE International Conference on Management of Innovation and Technology, Singapore, 2006, pp. 678-680.
15. A. Cockburn, *Agile software development: the cooperative game*. Addison-Wesley, 2007., "A Game-Theoretical model for task assignment in project management," in 2006 IEEE International Conference on Management of Innovation and Technology, Singapore, 2006,
16. R. L. Baskerville, L. Levine, B. Ramesh, and J. Pries-Heje, "The high speed balancing game: How software companies cope with internet speed," *Scandinavian Journal of Information Systems*, vol. 16, no. 1, pp. 11–54, 2004.
17. K. Sullivan, P. Chalasani, and S. Jha, "Software design decisions as real options," University of Virginia, Tech. Rep., 1997.
18. V. Sazawal and N. Sudan, "Modeling software evolution with game theory," *Trustworthy Software Development Processes*, vol. 5543, pp. 354–365, 2009.
19. Xing Gao, Weijun Zhong, Shue Mei, *A game-theory approach to configuration of detection software with decision errors*, 2013
20. Nie Gao-hui, *Analysis on Enterprise's Software Project Management Based on Game Theory*, *Management Science and Engineering*, 2006
21. A. Soska, J. Mottok, C. Wolff, *An experimental card game for software testing: Development, design and evaluation of a physical card game to deepen the knowledge of students in academic software testing education*, *EDUCON*, 2016 IEEE, 2016
22. Oscar Pedreira, Félix García, Nieves Brisaboa, Mario Piattini, *Gamification in software engineering – A systematic mapping*, *Information and Software Technology*, v. 57, 2015
23. Elizabeth Sweedyk, Robert M. Keller, *Fun and games: a new software engineering course*, *ITiCSE '05 Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, Pages 138-142, 2005
24. N. Kitagawa, H. Hata Nara, A. Ihara, K. Kogiso, K. Matsumoto, *Code review participation: game theoretical modeling of reviewers in Gerrit datasets*, *CHASE '16 Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*, Pages 64-67, 2016
25. Claudia Szabo, *Evaluating GameDevTycoon for teaching software engineering*, *Proceeding SIGCSE '14 Proceedings of the 45th ACM technical symposium on Computer science education*, Pages 403-408, 2014
26. Yilmaz, Murat and O'Connor, Rory, *Oyun Kuramı Kullanarak Yazılım Takımlarının Üretkenliğini Artırmak İçin Geliştirilen Bir Yazılım Süreç Mühendisliği Yaklaşımı*, *Turkish National Software Engineering Symposium*, 2011
27. R. Conradi and A. Fuggetta, "Improving Software Process Improvement," *IEEE Software*, vol. 19, no. 4, pp. 92–99, 2002
28. Y. Dittrich, C. Floyd, and R. Klischewski, *Social thinking software practice*. The MIT, 2002.
29. M. Grechanik and D. E. Perry, "Analyzing Software Development as a Noncooperative Game," in *IEE Seminar Digests*, 2004, vol. 29.
30. Yilmaz, Murat and O'Connor, Rory (2010) *Maximizing the value of the software development process by game theoretic analysis*. In: 11th International Conference on Product Focused Software, 21-23 Jun 2010, Limerick, Ireland. ISBN 978-1-4503-0281-4
31. Yilmaz, Mert and Yilmaz, Murat and O'Connor, Rory and Clarke, Paul (2016) *A gamification approach to improve the software development process by exploring the*

- personality of software practitioners, *Communications in Computer and Information Science*, 609 . Springer, pp. 71-83. ISBN 978-3-319-38980-6
32. Yilmaz, Murat, 2013, A software process engineering approach to understanding software productivity and team personality characteristics, PhD Thesis, Dublin City University.
 33. Yilmaz, Murat and O'Connor, Rory and Collins, John (2010) Improving software development process through economic mechanism design. In: 17th European Software Process Improvement Conference, 1-3 Sept 2010, Greno France. ISBN 978-3-642-15666-3
 34. Yilmaz, Murat and O'Connor, Rory (2016) A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study. *Tehnicki Vjesnik (Technical Gazette)*, 23 (1). pp. 237-245. ISSN 1330-3651
 35. Yilmaz, Murat and O'Connor, Rory (2012) A market based approach for resolving resource constrained task allocation problems in a software development process. In: 19th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2012), 25-27 June 2012, Vienna, Austria.
 36. J. C. Houk, J. L. Davis, D. G. Beiser, "Reward Mechanisms", The MIT, pp. 185 - 185, 1994
 37. N. Singh, N. S. Chaudhari, "Differential Reward Mechanism Based Online Learning Algorithm for URL-based Topic Classification", IEEE, CICN, 2014 International Conference
 38. K. Lua, S. Wanga, Ling Xiea, Zhen Wanga, b, Mingchu Li, "A dynamic reward-based incentive mechanism: Reducing the cost of P2P systems", vol. 112, pp. 105 - 113, 2016
 39. H. Wang, C. Tsai, *Game Reward Systems: Gaming Experiences and Social Meanings*, 2011
 40. S. P. Walz, Sebastian Deterding, "Gamification and Learning", MIT Press, pp. 688, 2014
 41. C. S. González, A. M. Carreño, "Methodological proposal for gamification in the computer engineering teaching", IEEE, Computers in Education, 2014 International Symposium
 42. W. Qu, Y. Zhao, M. Wang, B. Liu, "Research on teaching gamification of software engineering", IEEE, Computer Science & Education, 2014 9th International Conference
 43. F. Largo, F. Durán, C. Arnedo, P. Rosique, R. Cuerda, R. Carmona, "Gamification of the learning process: lessons learned", *Revista Iberoamericana de Tecnologías del Aprendizaje*, 2016
 44. R. M. Parizi, "On the gamification of human-centric traceability tasks in software testing and coding", IEEE 14th International Conference on, 2016
 45. R. M. Parizi, A. Kasem, A. Abdullah, "Towards gamification in software traceability: Between test and code artifacts", 2015 10th International Joint Conference on, 2015
 46. https://en.wikipedia.org/wiki/Team_Foundation_Server
 47. J. Hamari, K. Huotari, and J. Tolvanen, "Gamification and Economics", 2015 MIT Express