# Sanal Ofis Ortamında Kod Gözden Geçirme ile Kod Değerlendirmesi

Merve Kaymak[1], Mehmet Namıduru[1], Eray Tüzün[2] ve Murat Yılmaz[1,3]

[1]Oyun Araştırma ve Geliştirme Laboratuvarı, Çankaya Üniversitesi, Ankara
mervekaymak1993@gmail.com, mehmetnamiduru@gmail.com
[2]HAVELSAN A.Ş, Teknoloji ve Akademi Direktörlüğü, Ankara
etuzun@havelsan.com.tr
[3]Çankaya Üniversitesi, Bilgisayar Mühendisliği, Ankara
myilmaz@cankaya.edu.tr

**Özet.** Yazılım geliştirirken hataların erken fazlarda fark edilmesi, doğacak masrafları en aza indirgemektedir. Bu bağlamda kullanılabilecek süreçlerden biri "Kod Gözden Geçirme"dir. Yazılımcıların kod standartları, kod kalitesi, kod hataları gibi ölçütleri gözeterek birbirlerinin kodlarını değerlendirmesi, kod gözden geçirmedir. Bildiride ayrıntılarıyla bahsedilen projede, bu sürecin pratiğinin bir simülasyon ortamında takıma yeni dâhil olan pratisyene kazandırılması ve bu sayede eğitim maliyeti ve süresi azaltılırken eğlenerek öğrenme amaçlanmaktadır. Sanal ofis ortamını daha gerçekçi kılmak için ofis sesleri, objeleri, personeli, animasyonlarının yanı sıra sürecin gerçekçiliğini artırmak için verilen görevlerde bütçe ve zaman kısıtları bulunmaktadır. Kod gözden geçirmenin önemini daha iyi vurgulamak adına kod derleme ve test gibi yan unsurlara da projede yer verilmiştir. Bilgisayar mühendisliği öğrencileri ile yürütülen çalışmada, katılımcılar kendilerini bir ofis ortamında hissettiklerini ve verilen görevleri yerine getirirken eğlendiklerini belirtmişlerdir. Projeden senaryolarla gerçekleştirilen kullanılabilirlik testleri ve katılımcıların görüşleri doğrultusunda sorunlar tespit edilmiş ve bu kısımlarda iyileştirme yoluna gidilmiştir.

**Anahtar Kelimeler** – Oyunlaştırma, eğitim simülasyonu, kod gözden geçirme, kod derleme, yazılım testleri

## 1 Giriş

Simülasyonlar gerçek hayattaki riskler olmaksızın tecrübesiz kişilere gerçeğe yakın şekilde uygulamalı eğitim sağlayan ortamlardır ve eğitici yönleriyle geleneksel öğretme tekniklerine göre daha başarılı olmaktadır. Bu sebepten yazılım geliştirme süreçlerinin bir simülasyon ortamında öğretilmesi güncellik kazanmış ve pratisyenlerin eğitimi için çeşitli projeler yürütülmüştür.

Bildiride bahsedilen simülasyon projesi, kod gözden geçirme sürecini sanal bir ofis ortamında uygulamalı olarak öğretmeyi amaçlamaktadır. Kod gözden geçirme, kodu ikincil bir gözün değerlendirmesidir. Kod sahibi hariç diğer geliştiriciler kod standartları, kod kalitesi, kod hataları gibi ölçütleri gözeterek kodu inceler. Kod gözden geçirmenin kod kalitesini artırması, koddaki hataları azaltması, deneyimsiz geliştiricileri eğitmesi gibi faydaları olduğundan bu süreç, kod geliştirmenin bir parçası olmalıdır

[1]. Ayrıca kod gözden geçirme para ve zamandan tasarruf sağlar. Örneğin IBM, 20.000 satırlık bir programda test fazı yerine kod gözden geçirmede ana hataları bularak %85'ten fazla programcı eforu kazanmıştır [2].

HAVELSAN gibi savunma sektöründe faaliyet gösteren firmalarda özellikle görev kritik yazılımlarda kod gözden geçirme işlemi oldukça hassas bir şekilde uygulanmaktadır. Ancak yeni gelen takım elemanları bu işlemi tam olarak bilmediklerinden eğitimleri hem zaman kaybettirir hem de tecrübeli elemanların üretkenliğini olumsuz etkiler. Bu projeyle eğitim maliyeti ve zamanından tasarruf hedeflenmektedir.

Eğitici simülasyon örneklerinden olan SESAM [3], SimSE [4] ve SimVBSE [5], oyun tabanlı yazılım geliştirme süreci simülasyonlarıdır. Amaçları, katılımcılara sanal bir yazılım projesi sağlayarak katılımcıların duruma uygun kararlar almalarını öğretmektir. Katılımcılar personel işe alma, işten çıkarma, gerekli malzemeleri satın alma gibi bazı yetkilere sahiptir ve bu oyunlarla daha etkili proje yönetimi öğrenilebilir.

Literatürde bu bildirideki simülasyona en çok benzeyen çalışma Anukarna [6].'dır. Anukarna, kod gözden geçirmede pratik karar vermeyi öğreten bir yazılım mühendisliği simülasyonudur. Oyunda proje yöneticisi olacak kullanıcının sorularla kod gözden geçirmede uygulanacak 12 doğru yöntemi öğrenmesi amaçlanmaktadır. Anukarna daha çok proje yönetimi alanında sorularla teorik eğitim verirken, bildiride bahsedilen proje kod gözden geçirmeye odaklanmakta ve 3B ofis ortamında uygulamalı eğitim sağlamaktadır. Bildirinin 2. bölümünde eğitim simülasyonunun bölümleri anlatılmış, 3. bölümünde ise sonuçlardan bahsedilmiştir. Bildiri, özet bölümüyle sonlanmaktadır.

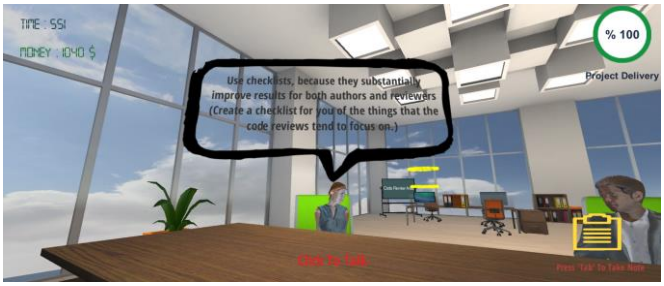
## 2 Simülasyon Bölümleri

Bu simülasyon kod gözden geçirme sürecini sanal bir ofis ortamında (Şekil 1'de görülebilir), gerçek bir yazılım projesi üzerinde çalışılıyor hissi vererek öğretmeyi amaçlamaktadır.


**Şekil 1**: Ofis ortamı

Bunun için kod gözden geçirmenin yanı sıra kod derleme ve maymun testi işlemleri de projeye dâhil edilmiştir. Maymun testi, programa verilen rastgele girdilerle programın davranışını gözlemlemeye yarayan bir test uygulamasıdır.

Gerçekçiliği artırmak için kullanıcıya başlangıç bütçesi ve zamanı verilmiştir. Şekil 1'deki örnekte sol üstte kalan zaman 30 gün ve bütçe 1000$ olarak belirlenmiştir. Her 30 saniyede bir kalan zaman 1 gün azalacaktır. Oyunda alınan yanlış kararlar ve harcanan zaman bütçeyi azaltırken, personelden bilgi almak (Şekil 2'de görülebilir) ve oyunun sonundaki soruları doğru yanıtlamak bütçeyi artırır. Oyunun her aşamasında personelden bilgi alınıp kod gözden geçirmeyle ilgili bilgiler elde edilebilir.




**Şekil 2:** Personelden bilgi alınması

Aşağıda, personelden alınabilecek iki bilgi örneği verilmiştir:

1. Kontrol listesi (checklist) kullanımı hem yazarı hem de gözden geçiren kişiyi önemli derecede geliştirmektedir [6].
2. Kod gözden geçirme süresi 60 ila 90 dakika olmalıdır [1].

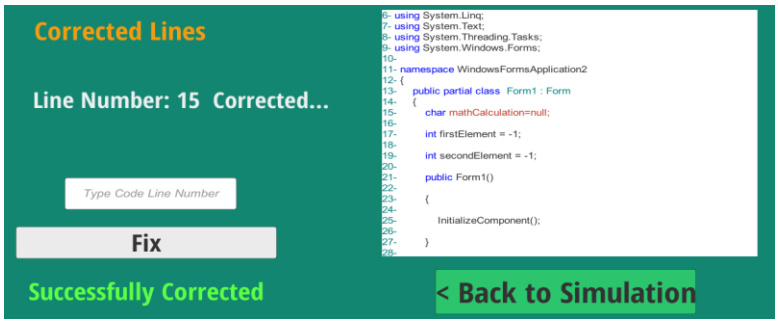
Ofiste yapılacak görevler için renklerle ayrılan farklı alanlar belirlenmiştir. Şekil 1'de yeşil alan kod seçim, kırmızı alan kod derleme**,** mavi alansa test alanıdır. Katılımcı, verilen göreve uygun yere yön tuşlarıyla gidebilir. Şekil 1'de sağ üstteki ilerleme çemberinden simülasyonun aşaması ve yüzde kaçının tamamlandığı görülebilir. Sağ alttaki not simgesine tıklanarak edinilen bilgiler not edilebilir ve notlar simülasyonun sonunda katılımcıların soruları cevaplamasına yardımcı olabilir.

### 2.1 Kod Gözden Geçirme

Bu aşamaya geçmeden önce kullanıcı yeşil alanda kod seçimi yapmalıdır. Yapılan çalışmada hali hazırda yalnızca tek bir kod parçası bulunmaktadır. Bu kod C# programlama dilinde yazılmış basit bir hesap makinesi kodudur.

Kod seçiminin ardından katılımcıdan sarı alana (kod gözden geçirme) gitmesi istenir. Sarı alandaki bilgisayara tıklayınca gelen ekran Şekil 3'teki gibidir. Katılımcı kodu gözden geçirir ve tespit ettiği hata satırlarını metin alanına birer birer girer. Girilen satırda hata varsa düzeltildi mesajı gösterilir ve o satır sistem tarafından düzeltilir, hata yoksa bütçeden 50$ düşülür. Örneğin 85. satırda bir hata varsa ve katılımcı bunu tespit ederse satır hatalı olmaktan çıkar ve ilerleyen aşamalara bu düzeltme yansır ancak hata yoksa kullanıcının bütçesinden para eksilir. Satır numarası yazılınca otomatik düzeltilen hataların katılımcılar tarafından düzeltilmesi planlanan çalışmalar arasındadır. Çoktan seçmeli bir soruyla katılımcıya hatalı satırı hangi kod satırıyla

değiştirmek istediği sorulacaktır. Kodda derleyici, mantık ve çalışma zamanı hatası kategorilerinde toplamda altı adet hata bulunmaktadır.



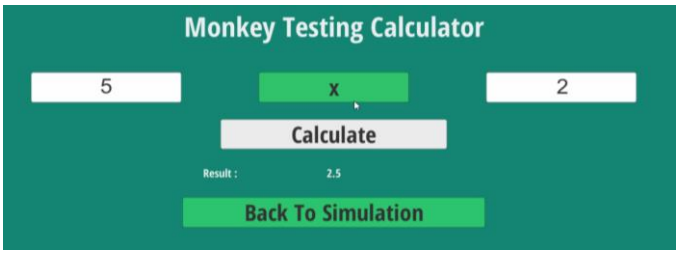

**Şekil 3**: Kod gözden geçirme ekranı

### 2.2 Kodu Derleme

Bu aşama, katılımcının asgari düzeyde kod gözden geçirme işlemini tamamlamış olmasını bir derleyici gibi kontrol eder. Kod derleme kısmından bir sonraki aşama olan test aşamasına ancak kod gözden geçirmede tüm derleyici hataları düzeltildiğinde geçilebilecektir. Kod gözden geçirmede düzeltilmeyen derleyici hataları Visual Studio 2015'ten alınan hata mesajlarıyla gösterilmektedir. Bu hataları düzeltmek için katılımcı tekrar kodu gözden geçirmek zorunda kalacaktır. Kod gözden geçirme düzgün yapılmadığında katılımcı zaman kaybedecek, bu da sonuca olumsuz yansıyacaktır.

### 2.3 Test

Test bölümünde ilk aşamadaki koda uygun bir kullanıcı arayüzü ile çalışılır. İki farklı sayı ve işlem seçiminden oluşan arayüz, kod gözden geçirmede düzeltilmeyen hatalarla işlemi gerçekleştirecektir. Örneğin Şekil 4'te de görülebileceği gibi bir mantık hatasının düzeltilmemesi "5*2" işlemini "2,5" olarak sonuçlandırmıştır. Bu hata testte fark edilmesi zor ancak kod gözden geçirmede kolaylıkla bulunabilecek türden bir hatadır çünkü yalnızca ilk sayının 5 olması durumunda testte tespit edilebilir.




**Şekil 4:** Hatalı işlem yapan test ekranı

Bu aşamada katılımcıdan hatalı olan özellikleri tespit ettikten sonra bu özelliklerin istenilen şekilde çalışması için tekrar kod gözden geçirme alanına gidip kodu daha dikkatli bir şekilde değerlendirerek hataları gidermesi beklenir. Yukarıda bahsedilen mantık hatası düzeltildikten sonra aynı işlemin sonucu "10" olacaktır.

Testte de derleme aşamasında olduğu gibi gözden geçirmenin yeterince iyi olmaması zaman kaybettirdiğinden, katılımcının performansı olumsuz etkilenecektir. Hatanın erken fazlarda bulunmasının çok daha az maliyeti olacağının altı çizilmek istendiği için iki aşamada da kullanıcı kod gözden geçirme işlemini tekrarlar.

### 2.4 Quiz

Simülasyonun sonunda 5 adet çoktan seçmeli soruyla katılımcının kazanımları ölçülecektir. Her yanlış cevap bütçeden 20$ götürürken doğru cevaplar bütçeyi 20$ artırmaktadır. Aşağıda sistemimizdeki sorulardan iki örnek verilmiştir:

1. Hangisi kod gözden geçirmenin avantajlarından biri değildir [5]?
    a. Deneyimsiz geliştiricilerin eğitilmesini sağlar.
    b. Daha kaliteli kod yazılmasına yardımcı olur.
    (c.) Yazılımın tüm hatalardan arınmasını sağlar.
    d. Test süresini kısaltır.
2. Hangi aşamada hata bulmak en az maliyetlidir?
    a. Test
    (b.) Kod gözden geçirme
    c. Bakım
    d. Dağıtım

### 2.5 Puanlama

Simülasyonun sonunda katılımcıya düzeltemediği hataların ve kalan bütçenin bilgisi verilmiştir. Katılımcı hangi bölümlerde bütçeden ne kadar para düşüldüğünün ve eklendiğinin bilgisine de erişebilecek, böylece hangi alanlarda kendisini geliştirmesi gerektiği konusunda fikir sahibi olabilecektir.

### 2.6 Yönetici Paneli

Simülasyonda ayrıcalıklı kullanıcıların olması yetkili kişilerin sisteme bilgi ve soru ekleyip sistemden çıkarabilmesi amacını taşımaktadır. Sisteme kullanıcı adı ve şifreyle giriş yapan yöneticiler gerekli kısımları doldurarak işlemlerini gerçekleştirebilirler.

## 3 Sonuçlar

Çalışma kapsamında, bilgisayar mühendisliği öğrencilerine simülasyon öncesinde yapılan anketle, öğrencilerin kod gözden geçirme sürecini ve önemini tam olarak bilmedikleri anlaşılmıştır. Ayrıca daha kullanıcı dostu bir program oluşturabilmek için öğrencilere simülasyonda tamamlamaları istenen 12 görevden oluşan bir senaryo verilmiştir. Katılımcılar bir görevi yapmakta zorlanırken, üç görevde daha az zorlanmış, diğer görevleri kolaylıkla yapabilmişlerdir. Görevler sonrasında katılımcılara memnuniyet anketi yapılmış, kullanımın genel olarak beğenildiği ancak bazı aşamalarda zorlandıkları anlaşılmıştır. Öğrencilerin önerileriyle bu noktalar iyileştirilmiştir.

## 4 Özet

Bu bildiride kod gözden geçirme sürecinin oyunlaştırılarak bir simülasyon ortamında takıma yeni dâhil olan pratisyene kazandırılması ve bu sayede eğitim maliyeti ve süresi azaltılırken eğlenerek öğreten bir uygulama anlatılmıştır.

Katılımcılar bu simülasyonu tamamlarken eğlendiklerini ve ofis ortamını gerçekçi bulduklarını da belirtmişlerdir. Literatürdeki örneklere göre üç boyutlu ortamla daha gerçekçi ve eğlenceli bir eğitim sağlanmıştır. Katılımcının proje yöneticisi bakış açısıyla değil, geliştirici açısından bakması sağlanıp kod gözden geçirmeye odaklanılmıştır. Yalnızca çoktan seçmeli sorularla değil uygulamalarla eğitim daha interaktif bir hale getirilmiştir. Ortamdaki karakterlerden alınan bilgiler ve simülasyon sonundaki sorularla içerik zenginleştirilmiş, bilgilendirici yönü de artırılmıştır.

Daha fazla senaryoyla kod gözden geçirmenin birçok pratiğinin katılımcılara daha etkili bilgi ve soru sistemiyle sağlanması planlanan çalışmalar arasındadır.

## Referanslar


1. J. Cohen *et al.* , “Resistence to Code Review”, in *Best Kept Secrets of Peer Code Review*, Canada: SmartBear Software, 2013, p. 19
2. S. Bhosale *et al.* , “Achieving Code Quality using Code Review System”, International Journal of Computer Applications, p. 24, 2012.
3. Iste.uni-stuttgart.de, “SESAM - Simulation System”, 2010. [Online]. Available:http://www.iste.uni-stuttgart.de/en/seold/forschung/schwerpunkte/sesam/system.html. [Accessed: 08- Nov- 2015].
4. E. Navarro and A. Hoek, “Comprehensive Evaluation of an Educational Software Engineering Simulation Environment”, 20th Conference on Software Engineering Education & Training (CSEET'07), pp. 195 - 201, 2007.
5. Apurva Jain and B. Boehm, “SimVBSE: Developing a Game for Value-Based Software Engineerin”, 19th Conference on Software Engineering Education & Training (CSEET'06), pp. 104-113, 2006.
6. R. Atal and A. Sureka. “Anukarna: A Software Engineering Simulation Game for Teaching Practical Decision Making in Peer Code Review”, 1st International Workshop on Case Method for Computing Education (CMCE 2015), pp. 63,70