

KOCAELİ
ÜNİVERSİTESİ
MÜHENDİSLİK
FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME PROJESİ

OYUN PROGRAMLAMADA
ÇEVİK YÖNTEMLER

ŞAHİN MERCAN

Prof.Dr. Nevcihan Duru Danışman, Kocaeli Üniv.

Doç.Dr. Sevinç İLHAN OMURCA Jüri Üyesi, Kocaeli Üniv.

.....

Dr. Öğr. Üyesi Orhan AKBULUT Jüri Üyesi, Kocaeli Üniv.

.....

Tezin Savunulduğu

Tarih: 26.01.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışması,

.....amacıyla gerçekleştirilmiştir.

Tez çalışmamda desteğini esirgemeyen, çalışmalarına yön veren, bana güvenen ve yüreklendiren danışmanım sonsuz teşekkürlerimi sunarım.

Tez çalışmamın tüm aşamalarında bilgi ve destekleriyle katkıda bulunan hocam teşekkür ediyorum. Tez çalışmamda gösterdiği anlayış ve destek için sayın teşekkürlerimi sunarım.

Hayatım boyunca bana güç veren en büyük destekçilerim, her aşamada sıkıntılarımı ve mutluluklarımı paylaşan sevgili aileme teşekkürlerimi sunarım.

Haziran – 2020

Bu dokümandaki tüm bilgiler, etik ve akademik kurallar çerçevesinde elde edilip sunulmuştur. Ayrıca yine bu kurallar çerçevesinde kendime ait olmayan ve kendimin üretmediği ve başka kaynaklardan elde edilen bilgiler ve materyaller (text, resim, şekil, tablo vb.) gerekli şekilde referans edilmiş ve dokümanda belirtilmiştir.

Öğrenci No:

160202069 Adı

Soyadı: SAHİN

MERCAN İmza:

SİMGELER VE KISALTMALAR DİZİNİ

$\alpha_{1,2,3}$:	Eğim için alınan açı, (°)
φ	:	Açı, (°)
θ	:	Açı, (rad)
d	:	Arıza noktasının referans baraya uzaklığı, (%)
d_{capS}	:	Seri kapasitörün S barasına uzaklığı, (%)
d_{capR}	:	Seri kapasitörün R barasına uzaklığı, (%)
d_S	:	Arıza noktasının S barasına uzaklığı, (%)
d_R	:	Arıza noktasının R barasına uzaklığı, (%)
f_0	:	İşaretin frekansı, (Hz)
f_s	:	Örnekleme frekansı, (Hz)
I^0	:	Sıfır bileşen akımı, (A)
I^1	:	Pozitif bileşen akımı, (A)
I^2	:	Negatif bileşen akımı, (A)
I_a	:	a fazı akımı, (A)
I_{ab}	:	a fazı ve b fazı akımları farkı, (A)
I_b	:	b fazı akımı, (A)
I_{bc}	:	b fazı ve c fazı akımları farkı, (A)
I_c	:	c fazı akımı, (A)
I_{ca}	:	c fazı ve a fazı akımları farkı, (A)

Öz direnç, (Empedans,öz direnç olarak adlandırılmaz.) Öz direnç, maddenin kimyasal özelliğinden dolayı direncinin artması ya da azalmasına neden olan her maddeye özgü ayırt edici bir özelliktir. Farklı maddelerin empedansları aynı olabilir ama öz dirençleri aynı olamaz. $R = Lq/Q$ dur. (Resistif Direnç= Uzunluk*öz direnç/kesit, (**empedans** ya da **elektriksel empedans** olarak da adlandırılır) **Alternatif akım**'a (İngilizce'de **AC**) karşı koyan zorluk olarak adlandırılır. İçinde **kondansatör** ve **endüktans** gibi zamanla değişen değerlere sahip olan elemanlar olan devrelerde direnç yerine öz direnç kullanılmaktadır. Öz direnç **gerilim** ve **akımın** sadece görünür **genliğini** açıklamakla kalmaz, ayrıca görünür **fazını** da açıklar. **DA** (DC) devrelerinde öz direnç ile **direnç** arasında hiçbir fark yoktur. Direnç sıfır **faz açısına** sahip öz direnç olarak adlandırılabilir.

Öz direncin genellikle **z** sembolü ile gösterilir ve hem genliğini hem de fazını ifade eden gösterim $|z| \angle \theta$ 'dir. Bununla beraber **karmaşık sayı** ifadesi **devre analizi** uygulamalarında daha sık kullanılır. *Empedans* (öz direnç) ifadesi ilk olarak Temmuz 1886'da **Oliver Heaviside** tarafından kullanıldı.^{[1][2]} 1893'te **Arthur Kennelly** de öz direnci karmaşık sayılarla ilk kullandı.^[3]

Öz direnç, **frekans domeninde** gerilimin akıma bölümüdür. Diğer ifadeyse, gerilim–akım oranı belirli ω sıklıktaki tek bir **karmaşık kuvvettir**. Öz direnç genellikle karmaşık sayıdır. Fakat bu karmaşık sayı, **Uluslararası Birimler Sistemi** (SI)'ya göre direnç ile aynı **birime** sahiptir ve o da **ohm**'dur. Sinüzoid bir akım veya gerilim için, Karmaşık öz direncin **kutupsal form**, gerilim ve akımın genlik ve fazını ifade eder. Özellikle,

- Karmaşık öz direncin genliği, voltaj genliğinin akımın genliğine oranıdır.
- Karmaşık öz direncin fazı, akımla gerilim arasındaki **faz farkıdır**.

Empedansın çarpmaya göre tersi **admittanstır**. (örn, admittans akımın gerilime oranıdır ve **siemens** olarak bilinir. Eskiden birimi mho idi).

GİRİŞ

Yazılım süreçlerine başlarken ilerleyen safhalarda değişim isteklerin oluşabileceği öngörülmektedir. Bu nedenle geleneksel yöntemler yazılım süreçlerinin en başında kapsamlı bir çalışma yaparak oluşabilecek ihtiyaçların hepsini belirleyip ilerleyen safhalarda oluşabilecek değişiklikleri önlemeyi esas almaktadır. Ancak hızla değişen, gelişen ortam ve piyasa koşullarına karşın artık proje gereksinimleri daha fazla ve hızlı bir şekilde değişmektedir. Proje başında bütün ihtiyaçların eksiksiz bir şekilde belirlenmesi neredeyse imkânsız bir hal almaktadır. Bu nedenle kullandığımız proje yönetim modelinin bu şartlara olabildiğince uyum sağlaması beklenmektedir. Uyum sağlama konusunda yeni teknikler ortaya süren çevik yöntemlerin kullanıldığı proje yönetim biçimleri ortaya çıkmıştır [1, 2]. Bu yöntemler yapılan uygulamaya göre esneklik gösterebilir, istenirse tümünde istenirse belirli bir kısmına uygulanabilir ayrıca geliştirmeyi yapan takıma uyacak şekilde kullanılabilir [3, 4, 5]. Çevik metot proje yönetimi çeşitleri ise Arık Geliştirme (AG), Uyarlanabilir Yazılım Geliştirme (UYG), Scrum, Ekstrem Programlama (EP), Kristal Yöntemleri, Özellik GÜdümlü Geliştirme (ÖGG), Açık Kaynak Kod Geliştirme (AKKG) ve Dinamik Sistem Geliştirme (DSG) yöntemleridir. Belirtilen metotlardan Arık Geliştirme dışında diğerleri 2001 yılında Çevik Yazılım Geliştirme Manifestosu oluşturulduktan sonra çevik yazılım ilkeleri tanımlanmaya başlanmıştır. Bu ilkeler aşağıda verilmiştir. 1. “Süreç ve geliştirme araçlarının yerine bireyler ve bireyler arası ilişkileri” [6] 2. “Detaylı dokümantasyon yerine koştan yazılımı” [6] 3. “Sözleşme görüşmeleri yerine müşteri iş birliğini” [6] 4. “Plan takip etmek yerine değişikliklere ayak uydurmayı” [6] Belirtilen 4 madde bu yöntemlerin ortaya çıktığı bildiride bulunmakta ve bu metotların etkili bir şekilde kullanılması için gerekli olan maddelerdir [6]

One of the problems experienced in game programming lately is that game programming requirements are constantly changing. Therefore, the inability of the process we carry out while programming the game to keep up with this change causes the product we develop to fail or to result in higher costs. Therefore, the use of agile programming, which brings a more dynamic and modern

solution to changing environment conditions, is increasing day by day and the project teams are transitioning to these processes. In this thesis, firstly historical development, content, rules and methods of Agile methods will be presented with literature researches. Using the Scrum method, which is the agile programming method, in the next section. "Grumpy Ball" and "Balloon Shooting" applications will be made. In the light of the results, it is a study on how to determine the project deadline, which is one of the disadvantages of agile methodology.

Bu yöntemin en önemli özelliği projedeki kişilere, ekipmanlar ve dokümanlardan daha fazla önem verilmesidir. [13]. Bunun nedeni geliştirmeyi yapan yazılımcılar olduğu için ayrıca geliştirmedeki ekipmanları da kullanan yazılımcılar olduğu için kişilere daha çok önem vermektedir. [14]. Bu nedenle bu metotlar geliştiren kişilerin özgüvenini arttırmayı hedefler ve bu nedenle kişilere her türlü desteği vermeye önem gösterir [14]. Çevik metotlar geleneksel metotların aksine kişilere önem verdiği için diğer ekipman belge vb. unsurları çok fazla dikkate almadan bir an önce geliştirmeye başlamayı hedeflemektedir. Bu sebeple geliştirme başlarken uğraşılacak doküman gibi benzeri şeyleri sadece istenilen durumlarda hazırlayarak daha hızlı ve esnek ilerlemeyi amaçlar [15]. Çevik metotları diğerlerinden ayıran husus ise değişen şartlara her şeyden daha çok değer vermesidir [15]. Geliştirici ekip müşteri beklentilerini karşılamaya çalıştığı için daha kısa sürede işin tamamlanacağı sonucuna varılabilir. Fakat ileride müşterinin beklentileri genel olarak farklılaştığından geliştirme başarısız da olabilir. Sonuç olarak geliştirme başında direk sonuca odaklanarak müşterinin daha sonraki aşamalarındaki beklentilerini dikkate almayarak ya da bu beklentileri olumsuz bir şekilde karşılamak geliştirmenin olumsuz sonuçlanmasına yol açar [15]. Çevik metotlar ise bu durumları önleyecek şekilde sürekli esnek bir geliştirme sürecini temel almaktadır. Çevik metotlar, dinamik, bağlama özgü, ısrarla değişmeyi zorlayan ve büyüme odaklı bir yazılım geliştirme metodudur. Bu durum metodun kullanımındaki başarıma ve kazanma isteği ile ilgilidir. Günümüzde, ortaya çıkan rekabetçi alanlarda başarılı olma, pazar payı ve müşteriler yazılım geliştiren şirketlerin odak noktası haline gelmiştir [16]. Geliştirme ekibini verimli kullanmak manevra kabiliyeti, hız ve maliyet tasarrufu kazandırır [16]. Örneğin birlikte çalışan birkaç tasarımcı, her birinin tek başına üreteceği tasarımdan daha iyi bir tasarım ürünü ortaya koyabilir [16]

OYUN PROGRAMLAMADA ÇEVİK YÖNTEMLER

ÖZET

Son dönemlerde oyun programlamada yaşanan sorunlardan birisi oyun programlama gereksinimlerinin sürekli değişmesidir. Bu yüzden oyun programlarken yürüttüğümüz sürecin bu değişime ayak uyduramaması geliştirdiğimiz ürünün başarısızlıkla ya da daha yüksek maliyetlerle sonuçlanmasına yol açmaktadır.

Bu nedenle son zamanlarda değişen ortam koşullarına karşı daha dinamik ve modern bir çözüm getiren çevik yöntemlerin proje yönetiminde uygulaması giderek artmaktadır.

Tezimizde ilk olarak çevik süreçlerin ortaya çıkışı, temel ilkeleri ve bu konu hakkındaki diğer çalışmalardan bahsedilecektir.

İlerleyen bölümlerde ise çevik programlama yöntemi olan Scrum metodu kullanılarak “Huysuz Top” ve “Balon Vurma” uygulamaları yapılacaktır. Çıkan sonuçlar ışığında çevik metodolojinin dezavantajlarından biri olan proje teslim tarihinin nasıl tespit edilebileceğine yönelik bir çalışmadır.

Anahtar kelimeler: Oyun programlama, Çevik Yöntemler , Scrum , proje teslim tarihi.