*Araştırma Makalesi / Research Article*

# Türkiye'de Çevik ve Klasik Yazılım Geliştirme Metodolojilerine Dair Kapsamlı Bir Değerlendirme

Duygu ÇALIŞKAN[1], Ahmet Faruk YAVUZ[1], Buket DOĞAN[1*], Banu ÇALIŞ USLU[2]

[1]*Marmara Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, 34722, Kadıköy, İstanbul*
[2]*Marmara Üniversitesi, Mühendislik Fakültesi Endüstri Mühendisliği, 34722, Kadıköy, İstanbul*
*(ORCID: 0000-0003-4523-2967) (ORCID: 0000-0001-6161-1778 )*
*(ORCID: 0000-0003-1062-2439) (ORCID: 0000-0001-8214-825X)*

**Öz**

Hızla gelişen teknoloji dünyasında; özellikle yazılım sektöründe geliştirilen ürünlerin, pazara hızlı sunulması ve değişen gereksinimlere çabuk yanıt vermeleri gerekmektedir. Değişen şartlara uyum sağlayabilmek için, dünyadaki yazılım endüstrisinde klasik ve çevik yaklaşım metodolojileri kullanılmaktadır. Bu çalışma kapsamında, Türkiye yazılım endüstrisindeki organizasyonlarda çevik ve geleneksel yaklaşımların hangi düzeyde kullanıldığı ve yazılım geliştiricilerin bu yaklaşımlarla ilgili görüşlerinin nasıl olduğunu belirlemeye yönelik bir anket hazırlanarak uygulanması sağlanmıştır. Hazırlanan anket, https://www.onlineanketler.com/ çevrimiçi anket sağlama hizmeti kullanılarak, 2019 Kasım ayı boyunca katılımcıların erişimine açılmıştır. Katılımcılar çeşitli sosyal ağlar ve e-posta grupları kullanılarak anket bağlantısına davet edilmişlerdir. Yazılım sektöründe yer alan birçok farklı firmadaki çalışana ulaşılarak, 193 kişinin anketi tamamlaması sağlanmıştır. Katılımcılardan alınan anket sonuçlarına göre, çevik yöntemlerin uygulanma oranının geleneksel yöntemlere göre daha fazla olduğu tespit edilmiştir. Çevik yaklaşımları kullanan katılımcıların bu yöntemin verimliliği, kaliteyi ve müşteri memnuniyetini arttırdığı görüşünde olduğu görülürken, klasik yaklaşım kullanan katılımcıların bu konularda kararsız olduğu belirlenmiştir.

**Anahtar kelimeler**: Yazılım geliştirme metodolojileri, çevik yaklaşım, çevik yazılım geliştirme, geleneksel yazılım geliştirme, yazılım sektörü.

# A Comprehensive Assessment of Agile and Classic Software Development Methodologies in Turkey

**Abstract**

In the rapidly evolving technology world, especially in the software industry, it requires that the developed products be presented to the market quickly and react quickly to changing needs. In order to adapt to changing conditions, classical and agile approach methodologies are used in the software industry in the world. In this study, a questionnaire was prepared and implemented to determine at which level agile and traditional approaches are used and how the developers' views on these approaches at organizations in the software industry in Turkey. The prepared questionnaire was made available to the participants throughout November 2019 using the online questionnaire service https://www.onlineanketler.com/. Participants were invited to the survey link using various social networks and e-mail groups. By reaching employees in many different companies in the software industry, 193 people participated in the survey. According to the survey results obtained from the participants, it has been determined that the rate of application of agile methods is higher than that of traditional methods. While it was seen that the participants using agile approaches think that this method increases the efficiency, quality and customer satisfaction, the participants using the classical approach are determined to be neutral on these issues.

**Keywords:** Software development methodologies, agile approach, agile software development, traditional software development, software industry.

## 1. Giriş

Günümüzde gelişen teknoloji yapısı ile birlikte yazılım hayatımızın vazgeçilmez bir parçası olmuştur. Teknoloji gerek ev ve iş yerlerimizde kullandığımız cihazlarda gömülü sistem şeklinde, gerekse sosyal

*Sorumlu yazar: buketb@marmara.edu.tr
*Geliş Tarihi: 13.02.2020, Kabul Tarihi: 04.06.2020*

ağlar/uygulamalar aracılığı ile cep telefonları, tabletler ve bilgisayar ile doğrudan günlük hayatımıza girerek hayatımızın vazgeçilmez bir parçası olmuştur. Teknoloji, yazılımlar ile birlikte, her geçen gün hızla gelişmekle birlikte organizasyonlar iş süreçlerini kısa sürede başarı ile tamamlayabilmek için etkin yazılım sistemlerine ihtiyaç duymaktadırlar. Bu yazılımı bilgi sistem birimleri kendi içinde (inhouse) üretir ya da hizmet alımı şeklinde (outsource) de geliştirebilmektedirler.

Yazılım projelerinin hayata geçirilmesinde kullanılan yaklaşımlar önemli bir rol oynamaktadır. Projenin başarısında geliştirilen ürünlerin etkin ve verimli olması etkili olmakla birlikte pazara sunulan ürünün geliştirilmesinde kullanılacak yaklaşıma karar verilmesi de etkilidir [1,2]. Bu çalışmanın amacı, yazılım mühendisliğinin ortaya çıkması ile birlikte klasik ve çevik (agile) yaklaşım metodolojilerinin Türkiye'de pratikte kullanımına ilişkin mevcut durumun sektör çalışanlarının görüşleri ile birlikte karşılaştırmalı olarak sunulmasıdır.

### 1.1. Yazılım Proje Yönetimi

Proje yönetimi, proje hedeflerine ulaşmak için zaman, maliyet ve kalite fonksiyonları göz önünde tutularak, mühendislik aktivitelerinin doğru olarak planlanması, organize edilmesi, kontrol edilmesi ve denetlenmesi ile projeye liderlik edilmesidir [3]. Bilgi teknolojilerinin hızlı gelişimi ve müşteri isteklerindeki değişimler, mevcut proje içerisinde değişikliğe sebep olabilmekte ve önceki projeler uygulanırken kazanılmış deneyimler ise genellikle yeterli gelmemektedir. Yazılım proje yönetimi metodolojisinin amacı; bir bilişim faaliyeti olan yazılımın (projenin) standart bir yöntemle, disiplinli, iyi yönetilen, ürünlerin ya da sonuçların kaliteli teslimini güvence altına alarak, belirli bir zaman ve bütçe kısıtı altında gerçekleştirmektir [4].

### 1.2. Yazılım Geliştirme Metodolojileri

Yazılım Geliştirme Metodolojileri, bir bilgi sisteminin geliştirme sürecinin yapımını, planlamasını ve kontrolünü sağlayan bir altyapıdır. Her farklı altyapı (framework) zayıf ve güçlü yanlarıyla bir arada zaman içerisinde olgunlaşır, bu sebeple de teknik, organizasyonel ve takım unsurlarına göre farklı özellik gösterir [5]. Özellikle karmaşık ve büyük bir sistem geliştirmek, birçok görevin yapılmak zorunda olunduğu uzun ve karmaşık bir süreçtir. Bu durumda, geliştiricilerin birbirileriyle daha koordineli çalışması gerekmektedir. Metodoloji, süreçleri küçük görevlere ayırarak, ne yapılması gerektiğini belirterek proje yönetimine, planlamaya, çizelgelere, ilerlemeyi izlemeye yardım etmektedir [6]. Metodoloji; müşterinin isteklerine göre, projenin başarısızlığa uğramaması adına geliştiricilerin başlanacak yeni projede ya da mevcut projenin geliştirilmesinde izleyeceği yol haritasıdır. Bu sebeple, bir sistem geliştirme metodolojisi her proje için uygun olmayabilir. Yazılım geliştirme metodolojilerinin tarihsel gelişimi Tablo 1'de özetlenmiştir [6,7].

**Tablo 1.** Metodolojilerin Tarihsel Gelişimi

| Yıl | Yazılım Geliştirme Metodolojileri |
|---|---|
| 1970 | 1969'dan beri yapısal programlama |
| 1980 | 1980'den itibaren yapısal sistem analizi ve tasarım yöntemi (SSADM) |
| | Bilgi İhtiyaç Analizi / Yazılım sistemleri metodolojisi |
| 1990 | "Nesne yönelimli programlama (Object Oriented Programming- OOP)" 1960'ların başlarında geliştirildi ve 1990'ların ortalarında baskın bir programlama yaklaşımı haline geldi. |
| | 1991'den beri hızlı uygulama geliştirme (Rapid Application Design-RAD) |
| | Dinamik sistem geliştirme yöntemi (DSDM), 1994'ten beri |
| | Scrum , 1995'ten beri kullanılmaktadır. |
| | 1998'den beri takım yazılımı süreci (Team software process) |
| | 1998'den beri IBM tarafından sürdürülen Rational Unified Process (RUP) |
| | Uç Programlama (Extreme programming) |
| 2000 | Çevik Birleştirilmiş Süreç (Agile Unified Process , AUP), 2005'ten bu yana Scott Ambler tarafından sürdürüldü. |
| | Disiplinli çevik teslimat (Disciplined agile delivery- DAD) |
| 2010 | Ölçekli Çevik Çerçeve (Scaled agile framework- SAFe) |
| | Büyük Ölçekli Scrum ( Large-Scale Scrum -LeSS) |

Bu çalışma kapsamında yazılım geliştirme metodolojileri temel olarak; Klasik ve Çevik yaklaşımlar olmak üzere iki temel başlık altında incelenmiştir.

### 1.2.1. Klasik (Traditional) Yaklaşımlar

1950'li yıllarda ilk yazılım geliştirme yöntemi olarak kullanılmakta olan yaklaşım, "kodla ve düzelt" diğer bir adıyla "ad hoc veya kovboy programlama" dır. Herhangi bir tasarım veya planlama aşaması olmadan ihtiyaçların kodlanması ve süreç içi veya elde edilen ürün teslimi sonrası ortaya çıkan hataların düzeltilmesi prensibine dayanmaktadır. Zaman içinde yazılım projelerinin büyüyerek bir proje yönetim metodolojisine ihtiyaç duyması, kritik sistemlerde hatalara tolerans gösterilemeyecek olması, teknolojinin ilerlemesiyle bilgisayar kapasitelerinin hızla büyüyüp, artması ve kaliteli yazılım geliştirilememesi gibi sebeplerden ötürü 1960'lı yıllarda yaşanan yazılım krizi sonucu NATO Bilim Komitesi tarafından 1968 yılında Almanya'da yeniden ele alınmıştır. 1970 yılında Dr. Winston Royce tarafından yayınlanan bir bildiride Şelale Modeli tanımlanmıştır ve yazılım geliştirme metodolojilerinin temeli haline gelmiştir [8].

Klasik yaklaşım ile proje yönetimi; yöneticinin faaliyetlerin ilerlemesini, analiz yapmasını ve sorgulamasını sağlayacak veriler sunmaması, iş dağılım düzenini sistemin tamamlayıcı bir yapısı olarak sunmaması, proje faaliyetlerinin birbiri ile olan ilişkisini tanımlayamaması gibi yönleriyle yetersiz kalmaktadır [7,8].

### 1.2.2. Çevik (Agile) yaklaşımlar

1990'lı yıllarda ortaya çıkan çevik (agile) yaklaşımlar; yazılım süreçlerini kısaltmak, daha esnek ve güçlü kılmak için kullanılan aşamalı olarak (iteratif ve artımlı) yazılım geliştirmeyi esas alan bir dizi yöntem olarak adlandırılmaktadır [9]. Çevik metotlar sayesinde, piyasaya çok hızlı şekilde ürün çıkarabilme ve değişen isteklere hızlı yanıt verme amaçlanmaktadır [9]. Geç ortaya çıkan değişim isteklerinin yönetilebilmesi, kendi kendini yöneten bir ekip, müşteri ile koordineli bir şekilde çalışılması, ekip içi iletişimin arttırılması vb. özellikler çevik süreçlerin genel karakteristik özellikleri arasında sayılabilmektedir. Geleneksel yazılım geliştirme yöntemlerinin, yeni teknoloji yazılım geliştirme faaliyetlerinin ihtiyaçlarını karşılamada yetersiz kalması nedeniyle, çevik yazılım geliştirme metotları daha popüler hale gelmiştir [10].

Geleneksel yazılım geliştirme yöntemlerinde analiz ve planlama, proje başlamadan önce proaktif bir şekilde yapılmaktadır. Çevik yaklaşımlar ile organizasyona değişikliklere kolay adapte olabilmeyi sağlayan daha esnek bir yapı sağlanmaktadır. Proje gereksinimlerindeki, teknolojideki ve organizasyon yapısındaki değişiklikler çevik yaklaşımlar ile daha iyi tahmin edilebilmekte ve yönetilebilmektedir [11]. Çevik yaklaşımlar, anahtar tedarikçiler ve anahtar müşteriler ile bilgi paylaşımına giderek, sanal bir ağ oluşturmalıdırlar. Web destekli bu sanal ağ aracılığıyla etkin biçimde bilgi paylaşımı oluşturularak hem müşterilerin ihtiyaçları hem de değişimler çok daha çabuk ve doğru şekilde teşhis edilebilmektedir [12].

Çevik yaklaşımlarda kısa süreli geliştirme evreleri sayesinde ortaya erken ve sık sürüm çıkartıldığı için, prototip halinde dahi olsa kısa sürede çalışan bir ürün görmek mümkün hale gelmiştir. Bu durum, organizasyonlar için projede çıkacak sıkıntılara karşı, sapmalara erken müdahale şansı vermesi açısından organizasyonlara avantaj sağlamaktadır[13]. Tablo 2'de yazılım geliştirme metodolojilerinin farklı açılardan değerlendirilmesine ilişkin detaylar ve değerlendirme yaklaşımları görülmektedir [14].

## 2. Materyal ve Metot

Bu çalışmada; Türkiye'deki yazılım organizasyonlarındaki çalışanların, çevik ve geleneksel yazılım geliştirme yöntemlerini ne düzeyde kullandıklarının ve yazılım geliştiricilerin çevik yaklaşımlarla ilgili görüşlerinin belirlenmesi amacıyla bir anket hazırlanmış ve uygulanması sağlanmıştır.

Tanımlayıcı incelemede amaç, değişkenler arası ilişkilerin tespit edilmesinden çok örneklemin ne kadarının belirli bir fikirde olduğu veya çeşitli olayların ne aralıkla olduğunun belirlenmesidir [15]. Bu çalışmada, örneklemin yaklaşımlara karşı düşüncesinin ne olduğu, hangi yaklaşımın seçilmesi durumunda katılımcıların durumu nasıl değerlendirildiği incelenmiştir.

**Tablo 2.** Metodolojilerin Değerlendirilmesi

| | **Klasik Metodolojiler** | | | | | | **Çevik Metodolojiler** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Açıklama** | **Şelale** | **Spiral Model** | **V Model** | **Artımlı Model** | **Prototipleme** | **Geliştirme Evrimsel** | **Scrum** | **Kanban** | **RUP** |
| Gereksinimlerin Belirlenmesi | Başlangıç | Belirli Sıklıkla | Başlangıç | Belirli Sıklıkla | Belirli Sıklıkla | Başlangıç | Belirli Sıklıkla | Belirli Sıklıkla | Belirli Sıklıkla |
| Dokümantasyon ve Eğitim Gerekliliği | Yüksek | Orta | Orta | Orta | Orta | Orta | Yüksek | Yüksek | Yüksek |
| Maliyet | Yüksek | Yüksek | Yüksek | Orta | Düşük | Düşük | Yüksek | Yüksek | Yüksek |
| Risk Duyarlılığı | Yüksek | Düşük | Orta | Düşük | Düşük | Yüksek | Düşük | Düşük | Düşük |
| Başarı Garantisi | Düşük | Yüksek | Düşük | Yüksek | Orta | Yüksek | Yüksek | Yüksek | Yüksek |
| Esneklik | Düşük | Yüksek | Düşük | Yüksek | Yüksek | Düşük | Yüksek | Yüksek | Yüksek |
| Uzmanlık Gerekliliği | Orta | Yüksek | Orta | Orta | Orta | Orta | Yüksek | Yüksek | Yüksek |
| Zaman Uzunluğu | Yüksek | Yüksek | Yüksek | Yüksek | Yüksek | Yüksek | Düşük | Orta | Orta |
| Bakım | Orta | Yüksek | Orta | Yüksek | Yüksek | Düşük | Yüksek | Yüksek | Yüksek |
| Yönetilebilirlik/Basitlik | Yüksek | Düşük | Orta | Orta | Orta | Düşük | Düşük | Düşük | Düşük |
| Uygulama Derecesi | Yüksek | Orta | Yüksek | Yüksek | Yüksek | Yüksek | Yüksek | Yüksek | Yüksek |

Araştırmada, yazılım geliştirme yöntemleri ile ilgili 35 maddelik beşli Likert türünde bir anket hazırlanmıştır. Anket üç temel kısımdan oluşmaktadır. İlk kısım katılımcılardan demografik verilerini elde etmek amacıyla hazırlanmıştır. Bu bölüm sonunda katılımcılara organizasyonlarında kullandıkları yazılım geliştirme yöntemi sorularak, çevik yöntemleri seçenlerin ikinci kısımda bulunan çevik yazılım geliştirme odaklı soruları cevaplamaları sağlanırken, geleneksel yöntemi seçenlerin ise üçüncü kısımda yer alan geleneksel yönteme yönelik soruları yanıtlamaları sağlanmıştır.

Anket soruları ve cevap seçenekleri araştırmacılar ve iki alan uzmanı ile birlikte hazırlanmıştır. Anket soruları beş kişilik pilot grup ile test edilmiş ve sonrasında yayına hazır hale getirilmiştir.

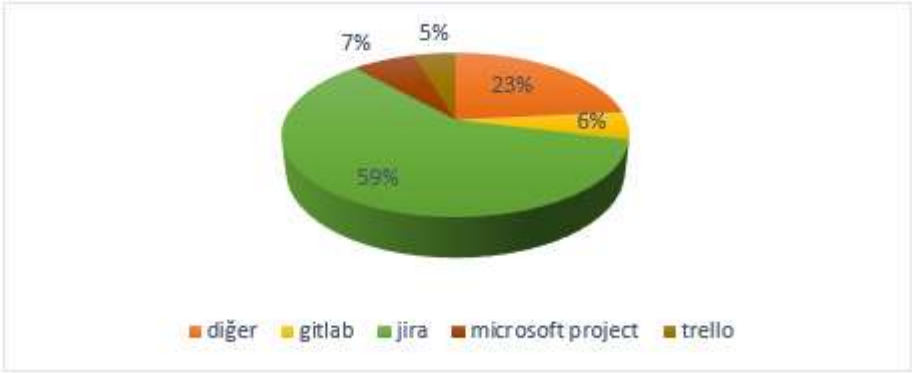
Hazırlanan anket, https://www.onlineanketler.com/ çevrimiçi anket sağlama hizmeti kullanılarak, 2019 Kasım ayında uygulamaya alınmış ve bir ay boyunca katılımcıların erişimine sunulmuştur. Katılımcılar çeşitli sosyal ağlar, e-posta grupları faydalanılarak davet edilmişlerdir. Bilişim sektöründe yer alan birçok farklı firmadaki çalışana ulaşılmıştır. Ankete 276 kişinin katıldığı, ancak anketin tamamında yer alan soruları 193 kişi tamamladığı belirlenmiştir. Kesinlikle Katılmıyorum ile Kesinlikle Katılmıyorum aralığındaki Beşli Likert tipi soruların değerlendirmesine ilişkin değer aralıkları aşağıdaki şekildedir; 1,00-1,80 aralığı "Kesinlikle Katılmıyorum", 1,81-2,60 aralığı "Katılmıyorum", 2,61-3,40 aralığı "Kararsızım", 3,41-4,20 aralığı "Katılıyorum", 4,21-5,00 aralığı ise "Kesinlikle Katılıyorum".

## 2.1.Verilerin Analizi

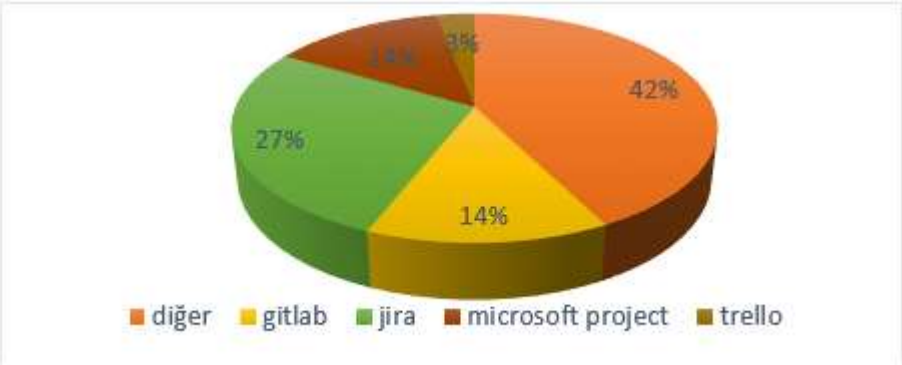
Anket çalışmasından elde edilen demografik bilgiler aşağıdaki şekilde özetlenmiştir. Yüzde değerlerin yanında parantez içerisinde yer alan değerler katılımcı sayısını ifade etmektedir.

1. Ankete katılanların %70'i anketi tamamlamıştır (193 kişi) ve ankete katılanların %72'sinin (139) erkek, %28'inin(54) kadın olduğu görülmüştür.
2. Ankete katılanların %69'u (133) lisans mezunu, %23'ü (44) yüksek lisans mezunu, %4'ü (8) yüksek okul mezunu, %3'ü (6) lise mezunu, %1'i (2) doktora mezunudur.

3. Ankete katılanların %50'si (97) Bilgi Teknolojileri alanında, %18'i (35) telekomünikasyon alanında, %14'ü (27) finans alanında, %1'i (2) sağlık alanında, %17'si (33) diğer alanlarda çalışmaktadır.
4. Ankete katılanların yazılım sürecindeki görevleri ise %45'i (87) yazılım uzmanı, %11'i (21) proje yöneticisi, %9'u (17) IT yöneticisi, %8'i (15) iş analisti, %6'sı (12) ürün sahibi, %5'i (10) test uzmanı, %5'i (10) operasyonel destek, %3'ü (6) takım üyesi ve %8'i (15) diğer görevlerde bulunmaktadır.
5. Anketi tamamlamış olan katılımcıların %26'sı (50) 2-5 yıl, %22'si (42) 1-2 yıl , %18'i (35) 0-6 ay, %15'i (29) 6-12 ay, %10'u (19) 10 yıl ve üzeri, %9'u (17) 5-10 yıl arasında bulunduğu kurumda görev yapmaktadır.
6. Katılımcıların %66'sı (127) yazılım süreç modeli olarak çevik yöntemi, %34'ü (66) klasik yaklaşımı kullanmaktadır.
7. Katılımcıların, kurumlarında çevik yaklaşım araçlarıyla proje yürütme süreleri ise; %15'i(29) 1 yıldan az, %34'ü (66) 1-2 yıl, %26'sı (50) 3-4 yıl, %22'si (42) 5-10 yıl, %3'ü (6) 10 yıldan fazla şeklinde dağılım göstermektedir.
8. Katılımcıların, kurumlarında klasik yaklaşım araçlarıyla proje yürütme süreleri ise; %11'i (21) 1 yıldan az, %8'i (15) 1-2 yıl, %9'u (17) 3-4 yıl, %18'i (35) 5-10 yıl, %54'ü (104) 10 yıldan fazla şeklinde dağılım göstermektedir.
9. Kurumlarında çevik yaklaşımı kullanan katılımcıların, çevik yaklaşım araçlarıyla tamamladığı toplam proje adetleri; %41'i (52) 31 adet ve üzeri, %25'i (32) 1-5 adet, %14'ü (18) 6-10 adet, %12'si (15) 11-15 adet ve %8'i (10) 16-30 adet olarak dağılım göstermektedir.
10. Kurumlarında klasik yaklaşımı kullanan katılımcıların, çevik yaklaşım araçlarıyla tamamladığı toplam proje adetleri; %15'i (10) 31 adet ve üzeri, %9'u (6) 1-5 adet, %47'si (31) 6-10 adet, %8'i (5) 11-15 adet, %21'i (14) 16-30 adet olarak dağılım göstermektedir.
11. Kurumlarında çevik yaklaşımı kullanan katılımcılardan %86'sı (109) süreç yönetimi için özel bir araç kullanmakta, %14'ü (18) ise kullanmamaktadır.
12. Kurumlarında klasik yaklaşımı kullanan katılımcılardan %70'i (46) süreç yönetimi için özel bir araç kullanmakta iken, %30'u (20) kullanmamaktadır.
13. Şekil 1'de görüldüğü gibi, süreç yönetimi için kullanılan özel araç olarak, kurumlarında çevik yaklaşımı kullanan katılımcılardan %59'u (75) Jira, %7'si (9) Microsoft Project, %5'i (6) Trello, %6'sı (8) Gitlab, %23'ü(29) diğer kategorisindeki araçları kullanmaktadır.
14. Şekil 2'de görüldüğü gibi, süreç yönetimi için özel araç kullanan kurumlarda klasik yaklaşımı kullanan katılımcılardan %27'si (18) Jira, %14'ü (9), Microsoft Project, %3'ü (2) Trello, %14'ü (9) Gitlab, %42'si (28) diğer kategorisindeki araçları kullanmaktadır.




**Şekil 1.** Çevik yazılım süreç yönetimi için kullanılan özel araçların dağılımı




**Şekil 2**. Klasik yazılım süreç yönetimi için kullanılan özel araçların dağılımı

## 3. Bulgular ve Tartışma

Bu bölümde katılımcıların değerlendirmeleri doğrultusunda araştırmadan elde edilen sonuçlar, verilen cevaplara göre ortalama ve frekans değerleri kullanılarak sunulmaktadır.

Tablo 3'te katılımcıların kurumlarında kullanılan yaklaşıma göre projelerin büyüklükleri göz önüne alarak yaptıkları başarı değerlendirmesi sonuçları görülmektedir. Çevik yaklaşımı kullanan katılımcıların sayısı 127, klasik yaklaşımı kullanan katılımcıların sayısı ise 66'dır. Katılımcı cevaplarına göre, Tablo 3'te görüldüğü gibi büyük projelerde çevik yaklaşımda başarı oranı %58 iken, klasik yaklaşımda bu değer %42'ye düşmektedir. En yüksek proje başarısızlığının %11 ile büyük projelerde klasik yöntem kullanıldığında ortaya çıktığı görülmektedir. Orta ve küçük projelerde de çevik yaklaşımın klasik yaklaşımdan daha fazla başarılı proje ortaya çıkmasını sağladığı görülmektedir. Küçük projelerde, en düşük proje başarısızlığı, %1 değeri ile çevik yöntem kullanıldığında görülmektedir.

**Tablo 3.** Kurumun başarı değerlendirmelerinin yaklaşımlara göre karşılaştırmalı analizi

| Proje Büyüklüğü | Yaklaşım | Başarılı | Sorunlu (Tamamlanmayan, maliyet ve süreyi aşan) | Başarısız |
|---|---|---|---|---|
| Büyük projeler | Çevik yaklaşım | 58% | %37 | %5 |
| | Klasik yaklaşım | 42% | %47 | %11 |
| Orta projeler | Çevik yaklaşım | 77% | %23 | - |
| | Klasik yaklaşım | 62% | %32 | %6 |
| Küçük projeler | Çevik yaklaşım | 86% | %13 | %1 |
| | Klasik yaklaşım | 77% | %18 | %5 |

Gerçekleştirilen çalışmada; farklı büyüklükte projelerde çevik veya klasik yaklaşımları kullanan katılımcıların verdikleri cevaplara göre, tüm proje büyüklüklerinde çevik yaklaşımın klasik yönteme göre daha fazla başarılı proje ortaya çıkarılmasını sağladığı belirtilmektedir. Bu sonuçların, Standish Group tarafından açıklanan Chaos Report 2015 sonuçları ile paralellik gösterdiği görülmektedir [16]. Chaos Report'ta çevik yaklaşımın, sırasıyla büyük, orta, küçük projelerdeki başarı oranları, %18, %27, %58 iken, gerçekleştirilen ankette bu değerler %58, %77,%86 olarak ölçülmüştür. Ayrıca, klasik yaklaşımın büyük, orta ve küçük projeler için başarısızlık oranları, Chaos Report'ta %42, %25, %11 iken, uygulanan anket sonuçlarında, %11, %6, %5 olarak görülmüştür. Sonuçlardaki farklar, aradan geçen yıllar içinde proje başarı değerlerinin değişmesinden kaynaklanabileceği gibi, araştırmanın Türkiye özelinde yapılmış olması nedeniyle bu bölgeye ait durumu yansıtması nedeniyle de ortaya çıkmış olabilir.

Tablo 4'te görüldüğü gibi, anket sorularına verilen cevapların yaklaşımlara göre yüzdelik sonuçları elde edilmiştir. Örneğin; kurumlarında çevik yaklaşım kullanan katılımcıların %59,1'i yaklaşımı kullanmanın, verimliliklerini arttırdığına katılmaktayken, kurumlarında klasik yaklaşım

kullanan katılımcıların %28,8'i yaklaşımı kullanmanın, verimliliklerini arttırdığına katılmaktadır. Kurumlarında klasik yaklaşım kullanan katılımcıların %33,3'ü ise yaklaşımı kullanmanın, verimliklerini arttırdığına katılmamaktadır. Anket çalışmasından elde edilen sonuçlar aşağıda maddeler halinde sunulmaktadır.

1. Çevik yaklaşımı kullanan katılımcıların%84'ü yaklaşımı kullanmanın verimliliklerini arttırdığını düşünürken, %6'sı artırmadığını düşünmektedir. Aynı soruya klasik yaklaşımı kullanılan katılımcıların %30'unun katıldığı, %44'ünün ise katılmadığı görülmektedir.
2. Çevik yaklaşımı kullanan katılımcılar, %76 oranında üretilen sistemlerin kalitesini arttırdığını, %9 oranında ise artırmadığını düşünürken, klasik yaklaşımı kullananlarda bu oranlar sırasıyla %29 ve %45 şeklindedir.
3. Çevik yaklaşımı kullanan katılımcılar, bu yöntemin geliştirme maliyetini azalttığına %58 oranında katılırken, %12 oranında katılmamaktadır. Klasik yaklaşımda ise aynı soruya katılımcıların %29'unun katıldığı , %49'unun ise katılmadığı görülmektedir.
4. Çevik yazılım kullanan katılımcıların %76'sı, yöntemin paydaşların memnuniyetini artırdığını belirtirken, %6'sının bu görüşe katılmadığı görülmektedir. Aynı soruya klasik yaklaşım kullanan katılımcıların %18 oranında katıldığı, %35 oranında ise katılmadığı belirlenmiştir.
5. Çevik yaklaşımı kullanan kişiler, yeterli dokümantasyon hazırlanmadığına %50 oranında katılırken, yeterli dokuman hazırlandığını düşünenlerin oranı %33 olmuştur. Klasik yaklaşımda aynı soru için katılımcılardan katılanların oranı %39, katılmayanların oranı ise %41 olarak gözlemlenmiştir.
6. Çevik yaklaşımı kullanan katılımcılar, yeterli analiz yapılmadığını %32 oranında, yapıldığını ise %47 oranında belirtirken, klasik yaklaşımda analizlerin yetersizliği %39, yeterliliği ise yine yüzde %39 oranında, eşit bir dağılım göstererek vurgulanmıştır.
7. Çevik yaklaşımı kullanan kişiler, planlama için yeterli zaman ayrılmadığını %24 oranında katılırken, %58 oranında katılmamaktadırlar. Klasik yaklaşım kullanan katılımcılarda ise, planlama için yeterli zaman ayırılmadığına katılanların oranı %35 iken, katılmayanların oranı %39 olarak gözlemlenmiştir.
8. Çevik yaklaşımı kullanan katılımcılar, bu yazılım geliştirme yaklaşımının, periyodik bir disipline uymadığına %16 oranında katılırken, %68 oranında katılmamaktadır. Klasik yaklaşımı kullanan katılımcılarda ise, bu yaklaşımın periyodik bir disipline uymadığına katılanların oranı %26, katılmayanların oranı ise %53 olarak hesaplanmıştır.
9. Çevik yaklaşımı kullanan katılımcılar, bu yazılım geliştirme yaklaşımının, sadece bir arada (aynı ofis içerisinde) çalışan takımlara uygun olduğunu görüşünü %35 oranında desteklerken, %49 oranında bu görüşe katılmadıklarını belirtmişlerdir. Klasik yaklaşım kullanan katılımcılarda ise; katılanlar %35, katılmayanlar %38 olarak gözlemlenmiştir.
10. Çevik yaklaşımı kullanan katılımcılar, yazılım geliştirme yaklaşımının, sadece küçük takımlar için uygun olduğunu fikrine %32 oranında katılırken, katılmayanların oranı %57 olmuştur. Klasik yaklaşımı kullanan katılımcılarda aynı görüşe katılanlar %36, katılmayanlar ise %39 olmuştur.
11. Çevik yaklaşımı kullanan katılımcılar, yazılım geliştirme yaklaşımı ile geliştirilen projelerin düşük kalitede olduğu fikrini %7 oranında desteklerken, kaliteyi düşürmediği fikrini savunanların oranı %83 olarak ortaya çıkmıştır. Klasik yaklaşımı kullanan katılımcılar da ise %21'i kaliteyi düşürdüğü yönünde, %53'ü ise kaliteyi düşürmediği yönünde görüş bildirmiştir.
12. Çevik yaklaşımı kullanan katılımcılar, yazılım geliştirme projelerini yönetmenin zor olduğunu %21 oranında desteklerken, %64'ü bu görüşe katılmadığını beyan etmiştir. Klasik yaklaşımı kullanan katılımcılarda görüşe destek verenlerin oranı %42 iken, katılmayanların oranı %35 olmuştur.
13. Çevik yaklaşımı kullanan katılımcılar, proje büyüklüğü arttıkça, projenin başarılı olma ihtimalinin azaldığını görüşünü %30 oranında desteklerken, azalmadığını düşünenlerin oranı %50 olmuştur. Klasik yaklaşım kullanan katılımcılarda ise bu görüşü destekleyenlerin oranı %52, desteklemeyenlerin oranı ise %20 olarak gözlemlenmiştir.

**Tablo 4.** Anket sorularına verilen cevapların yaklaşımlara göre karşılaştırmalı analizi

| | | **Çevik yaklaşım** | | | | | | **Klasik yaklaşım** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **İfade** | Kesinlikle Katılıyorum | Katılıyorum | Kararsızım | Katılmıyorum | Kesinlikle Katılmıyorum | **Ort.** | Kesinlikle Katılıyorum | Katılıyorum | Kararsızım | Katılmıyorum | Kesinlikle Katılmıyorum | **Ort.** |
| | | % | % | % | % | % | | % | % | % | % | % | |
| 1. | Yaklaşımı kullanmanın, verimliliğimizi arttırdığını düşünüyorum. | 24,4 | **59,1** | 10,2 | 3,9 | 2,4 | **3,99** | 1,5 | 28,8 | 25,8 | **33,3** | 10,6 | **2,77** |
| 2. | Üretilen sistemlerin kalitesini arttırdığını düşünüyorum. | 26 | **49,6** | 15,7 | 6,3 | 2,4 | **3,91** | 4,5 | 24,2 | 25,8 | **34,8** | 10,6 | **2,77** |
| 3. | Geliştirme maliyetini azalttığını düşünüyorum. | 14,2 | **44,1** | 29,9 | 7,9 | 3,9 | **3,57** | 3 | 25,8 | 22,7 | **39,4** | 9,1 | **2,74** |
| 4. | Paydaşların memnuniyetini arttırdığını düşünüyorum. | 21,3 | **55,1** | 18,1 | 3,9 | 1,6 | **3,91** | 1,5 | 16,7 | **47** | 27,3 | 7,6 | **2,77** |
| 5. | Yeterli dokümantasyon hazırlanmadığını düşünüyorum. | 10,2 | **39,4** | 17,3 | 26 | 7,1 | **3,20** | 18,2 | 21,2 | 19,7 | **34,8** | 6,1 | **3,11** |
| 6. | Yeterli analiz yapılmadığını düşünüyorum. | 8,7 | 22,8 | 21,3 | **35,4** | 11,8 | **2,81** | 13,6 | 25,8 | 21,2 | **33,3** | 6,1 | **3,08** |
| 7. | Planlama için yeterli zaman ayrılmadığını düşünüyorum. | 4,7 | 18,9 | 18,1 | **43,3** | 15 | **2,55** | 10,6 | 24,2 | 25,8 | **34,8** | 4,5 | **3,02** |
| 8. | Yazılım geliştirme yaklaşımının, periyodik bir disipline uymadığını düşünüyorum. | 3,9 | 11,8 | 16,5 | **44,1** | 23,6 | **2,28** | 6,1 | 19,7 | 21,2 | **37,9** | 15,2 | **2,64** |
| 9. | Yazılım geliştirme yaklaşımının, sadece bir arada (aynı ofis içerisinde) çalışan takımlara uygun olduğunu düşünüyorum. | 11 | 23,6 | 16,5 | **33,9** | 15 | **2,82** | 4,5 | 30,3 | 27,3 | **31,8** | 6,1 | **2,95** |
| 10. | Yazılım geliştirme yaklaşımının, sadece küçük takımlar için uygun olduğunu düşünüyorum. | 5,5 | 26,8 | 11 | **40,9** | 15,7 | **2,65** | 3 | **33,3** | 24,2 | 30,3 | 9,1 | **2,91** |
| 11. | Yazılım geliştirme yaklaşımı ile geliştirilen projelerin düşük kalitede olduğunu düşünüyorum. | 0,8 | 6,3 | 9,4 | **55,1** | 28,3 | **1,96** | 3 | 18,2 | 25,8 | **43,9** | 9,1 | **2,62** |
| 12. | Yazılım geliştirme projelerini yönetmenin zor olduğunu düşünüyorum. | 3,9 | 17,3 | 15 | **46,5** | 17,3 | **2,44** | 4,5 | **37,9** | 22,7 | 33,3 | 1,5 | **3,11** |
| 13. | Proje büyüklüğü arttıkça, projenin başarılı olma ihtimalinin azaldığını düşünüyorum. | 8,7 | 21,3 | 20,5 | **37,8** | 11,8 | **2,77** | 9,1 | **42,4** | 28,8 | 18,2 | 1,5 | **3,39** |

Katılımcılardan alınan cevapların ortalama değerlerinin karşılık geldiği ifadeler Tablo 5'te görülmektedir. Bu sonuçlara göre, Çevik yaklaşım kullanan katılımcıların bu yaklaşımın verimliliği ve kaliteyi artırdığına katıldıkları, klasik yaklaşım kullanan katılımcıların ise kararsızım görüşünde oldukları görülmektedir. Maliyetlerin azaltılması ile ilgili üçüncü soruda ise, çevik yaklaşım kullanan katılımcılar bu yöntemin geliştirme maliyetlerini azalttığına katılırken, klasik yaklaşım kullanan katılımcıların kararsız olduğu görülmektedir. Yeterli doküman hazırlandığı ve analiz yapıldığı konusunda ise her iki katılımcı grubunun da kararsız olduğu görülmektedir. Kullanılan yöntemle proje yönetmenin zor olduğu konusunda ise; çevik yaklaşım kullanan katılımcılar katılmıyorum görüşünde iken, klasik yaklaşım kullananların kararsız olduğu görülmektedir. Kullanılan yazılım geliştirme yönteminin sadece aynı ofis içerisindeki çalışan takımlara ve küçük takımlara uygun olduğu konusunda ise, her iki katılımcı grubunun kararsız görüşünde olduğu görülmektedir. Çevik yazılım geliştirme yaklaşımı kullanan katılımcı grubu, projelerin düşük kalitede olduğuna katılmazken, klasik yaklaşım

kullanan katılımcı grubunun kararsız olduğu görülmektedir. Proje büyüklüğü arttıkça başarılı olma ihtimalinin azaldığı konusunda ise her iki katılımcı grubunun da kararsız olduğu görülmektedir.

**Tablo 5.** Anket sorularına verilen cevapların yaklaşımlara göre ortalaması

| İfade | Çevik yaklaşım (ort.) | | Klasik yaklaşım (ort.) | |
|---|---|---|---|---|
| 1. Yaklaşımı kullanmanın, verimliliğimizi arttırdığını düşünüyorum. | 3,99 | Katılıyorum | 2,77 | Kararsızım |
| 2. Üretilen sistemlerin kalitesini arttırdığını düşünüyorum. | 3,91 | Katılıyorum | 2,77 | Kararsızım |
| 3. Geliştirme maliyetini azalttığını düşünüyorum. | 3,57 | Katılıyorum | 2,74 | Kararsızım |
| 4. Paydaşların memnuniyetini arttırdığını düşünüyorum. | 3,91 | Katılıyorum | 2,77 | Kararsızım |
| 5. Yeterli dokümantasyon hazırlamadığını düşünüyorum. | 3,2 | Kararsızım | 3,11 | Kararsızım |
| 6. Yeterli analiz yapmadığını düşünüyorum. | 2,81 | Kararsızım | 3,08 | Kararsızım |
| 7. Planlama için yeterli zaman ayrılmadığını düşünüyorum. | 2,55 | Katılmıyorum | 3,02 | Kararsızım |
| 8. Yazılım geliştirme yaklaşımının, periyodik bir disipline uymadığını düşünüyorum. | 2,28 | Katılmıyorum | 2,64 | Kararsızım |
| 9. Yazılım geliştirme yaklaşımının, sadece bir arada (aynı ofis içerisinde) çalışan takımlara uygun olduğunu düşünüyorum. | 2,82 | Kararsızım | 2,95 | Kararsızım |
| 10. Yazılım geliştirme yaklaşımının, sadece küçük takımlar için uygun olduğunu düşünüyorum. | 2,65 | Kararsızım | 2,91 | Kararsızım |
| 11. Yazılım geliştirme yaklaşımı ile geliştirilen projelerin düşük kalitede olduğunu düşünüyorum. | 1,96 | Katılmıyorum | 2,62 | Kararsızım |
| 12. Yazılım geliştirme projelerini yönetmenin zor olduğunu düşünüyorum. | 2,44 | Katılmıyorum | 3,11 | Kararsızım |
| 13. Proje büyüklüğü arttıkça, projenin başarılı olma ihtimalinin azaldığını düşünüyorum. | 2,77 | Kararsızım | 3,39 | Kararsızım |

## 4. Sonuç ve Öneriler

Bu araştırmada kapsamında, yazılım geliştirme sürecini birçok yönden önemli ölçüde geliştiren geleneksel yaklaşımla, dinamik iş ihtiyaçları ile başa çıkmayı kolaylaştırıcı çevik yaklaşımlar incelenmiştir. Her iki yaklaşımın, farklı kriterler üzerindeki etkinliklerinin değerlendirilmesi için, 35 maddelik beşli Likert türünde bir anket hazırlanmış, hazırlanan anket, yazılım sektöründe aktif rol alan 276 kişi ile paylaşılmıştır. Anketi tamamlayan 193 aktif kullanıcının görüşleri doğrultusunda, elde edilen bulgular değerlendirilmiştir.

Araştırmadan elde edilen bulgulara ilişkin olarak, küçük, orta ve büyük projelerde çevik yaklaşımın klasik yaklaşımdan daha başarılı bir proje yönetimi sağladığı katılımcı görüşleri ile de ortaya konmuştur. Klasik metodolojilere ilişkin olarak, metodolojiyi kullanan katılımcıların aktarmış oldukları bilgiler doğrultusunda, yazılım geliştirme sürecinde, belirsizlikler ve sürecin sonucuna ulaşmasında çeşitli güçlükler ortaya koyduğu yorumuna varılmıştır. Bu çalışmadaki katılımcı görüşlerine göre; çevik yaklaşım kullanan katılımcılar bu yaklaşımın verimliliği ve kaliteyi artırdığı görüşündeyken, klasik yaklaşım kullanan katılımcılar bu konularda kararsızdır. Yeterli doküman hazırlandığı ve analiz yapıldığı konusunda ise her iki katılımcı grubunun da kararsız olduğu görülmektedir. Bu durum, belgeleme ve analiz konusunda daha fazla hassasiyet gösterilmesi gerektiğini de göstermektedir.

Özellikle başarısız projelerdeki en büyük problemlerden birisi yazılım projelerindeki yetersiz analiz ve belgeleme olduğu için, yöntemlerin doğru ele alınması açısından bu sorunun çözümlenmesi önem arz etmektedir [17]. Kullanılan yöntemle proje yönetmenin zor olduğu konusunda ise; çevik yaklaşım kullanan katılımcılar çoğunlukla katılmıyorum görüşünde iken, klasik yaklaşım kullananların kararsız olduğu görülmektedir. Bu durum Çevik yöntem kullanan katılımcıların, çevik yöntemin gücünü doğru kullanabildiklerini ve proje yönetimini daha kolay yapabildiklerini göstermektedir.

Değişen dünyaya ayak uydurmak adına, klasik yöntemlerden çevik yönteme doğru olan geçiş döneminde, dönüşümü gerçekleştirmeyen şirketlerin veya konunun özelinde yazılım birimlerinin zamanla süreçlerinde kayıplar yaşayacağı görülmektedir.

Türkiye'deki organizasyonların yazılım geliştirme konusunda uluslararası düzeyde rekabet edebilmesi için kullandıkları yaklaşımlara proje başarısına etki eden faktörleri de ekleyerek projeleri hayata geçirmesi, sonuçların adapte edilebilir olması katma değeri yüksek sonuçlar doğurabilecektir. İlerideki çalışmalarda, yapılan ankete proje başarısına etki eden faktörlerin kategorize edilerek eklenmesi planlanmaktadır. Sonuçların, Türkiye pazarındaki birçok organizasyonun yazılım süreçlerinin iyileştirilmesinde alacağı kararlara yardımcı olması beklenmektedir.

## Yazarların Katkısı

Yazarlar makaleye eşit oranda katkı sağlamıştır.

## Çıkar Çatışması Beyanı

Yazarlar arasında herhangi bir çıkar çatışması bulunmamaktadır.

## Araştırma ve Yayın Etiği Beyanı

Yapılan çalışmada araştırma ve yayın etiğine uyulmuştur.

## Kaynaklar

[1] Williams L. 2010. Agile software development methodologies and practices. Advances in Computers, 80: 1-44.

[2] Flora H.K., Chande S.V. 2014. A systematic study on agile software development methodologies and practices. International Journal of Computer Science and Information Technologies, 5 (3): 3626-3637.

[3] El Bajta M., Idri A., Ros J.N., Fernández-Alemán J.L., de Gea J.M.C., García F., Toval A. 2018. Software project management approaches for global software development: a systematic mapping study. Tsinghua Science and Technology, 23 (6): 690-714.

[4] Ruhe G., Wohlin C. 2014. Software project management in a changing World. Springer-Verlag Berlin Heidelberg, 1-477.

[5] Erdem O.A., Younis A. 2014. Yazılım Projelerinin Geliştirme Sürecinde Yönetim. Bilişim Teknolojileri Dergisi, 7 (1): 1-8.

[6] Britton C., Doake J. 1993. Software Systems Development: A Gentle Introduction. McGraw-Hill, 1-214.

[7] Sommerville I. 2011. Software Engineering. 9th Edition, Pearson, 1-773.

[8] Gencer C., Kayacan A. 2017. Yazılım Proje Yönetimi: Şelale Modeli ve Çevik Yöntemlerin Karşılaştırılması. Bilişim Teknolojileri Dergisi, 10 (3): 335-352.

[9] Boehm B., Turner R. 2003. Using risk to balance agile and plan-driven methods. Computer, 36 (6): 57-66.

[10] Karlıdere T., Kalıpsız O. 2003. Yazılım Mühendisliği Projelerinde Çevik Yaklaşımların Yeri. Ulusal Yazılım Mühendisliği Sempozyumu, 23-25 Ekim 2003, İzmir.

[11] Marchesi M., Mannaro K., Uras S., Locci M. 2007. Distributed Scrum in research project management. In: International Conference on Extreme Programming and Agile Processes in Software Engineering, Springer, Berlin, Heidelberg, 240-244.

[12] Erol A.H. 2010. A heuristic solution algorithm for the quadratic assignment problems. Master thesis, Marmara University, Institute for Graduate Studies in Pure And Applied Science, İstanbul.
[13] Çamoğlu K., Akbayır D., Yücalar F., Bayraklı S. 2010. Bir Çevik Yazılım Geliştirme Sürecinin Uyarlanması ve Uygulanması. Havacılık ve Uzay Teknolojileri Dergisi, 4 (3): 57-67.
[14] Dhani H. 2016. Comparative Study and Analysis of Software Process Models on Various Merits. International Journal of Advanced Research in Computer Science and Software Engineering, 6 (9).
[15] Büyüköztürk Ş., Kılıç-Çakmak E., Akgün Ö., Karadeniz Ş., Demirel F. 2008. Bilimsel araştırma yöntemleri. Pegem Yayınları, Ankara.
[16] International S.G. 2015. The chaos report. United States of America. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf (Erişim tarihi: 02.01.2020).
[17] Kappelman L.A., McKeeman R., Zhang L. 2006. Early warning signs of IT project failure: The dominant dozen. Information Systems Management, 23 (4): 31-36.