

# Küçük Ölçekli Takımlar İçin TMM

Eren Narin, Dr. Sadık Eşmelioğlu

ODTÜ, Enformatik Enstitüsü, Ankara  
erennarin.92@gmail.com, esadik@metu.edu.tr

**Özet.** Yazılım endüstrisinde gelinen nokta, bilginin tekrar kullanımını kaçınılmaz hale getirmiştir. Özellikle uluslararası geçerliliği olan standartların süreçlerde kullanımı, projelerin maliyetlerini düşürmekte ve başarı olasılığını keskin bir şekilde arttırmaktadır. Ancak, kendini ispat etmiş ve uluslararası geçerliliğe ulaşmış standartların çoğu, büyük ölçekli organizasyonların bakış açısından geliştirilmiştir. İçinde bulunduğumuz ve “Girişimcilik Çağı” olarak adlandırılan bu çağda, küçük ölçekli organizasyonlar da yazılım endüstrisinde büyük rol oynamaktadırlar. Bu sebeple, küçük ölçekli firmaların bakışıyla geliştirilmiş standartlara olan ihtiyaç gün geçtikçe artmaktadır. Bu ihtiyaçtan yola çıkarak biz, bu çalışmada, uyarlanmış ve küçük ölçekli takımlar için özelleştirilmiş bir Test Olgunluk Modeli (TMM) geliştirmeyi hedefledik. Bu çalışma aynı zamanda, küçük takımların gereksinim, tasarım, kodlama gibi bütün süreçlerini kapsayan detaylı bir kalite rehberi oluşturma hedefimizin ilk adımıdır. TMM v2.0 dokümanı referans kaynak olarak kullanılmıştır ve bu dokümanın 2. ve 3. olgunluk seviyelerinin bütün pratiklerinin fayda/maliyet oranları, geçmiş tecrübeler ve akademik birikim kullanılarak puanlanmıştır. Bu puanlama aynı zamanda dört farklı vaka çalışması ile de desteklenmiştir. Vaka çalışmalarının katılımcıları teknokent yerleşkelerinde faaliyetlerini sürdüren küçük ölçekli yazılım firmalarından seçilmiştir. Puanlama işleminin sonunda pratikler yeniden önceliklendirilerek ve gruplandırılarak yeni olgunluk seviyeleri tanımlanmıştır. Bu yeni olgunluk seviyesi tanımları, küçük ölçekli takımların, yazılım test süreçlerini daha verimli yürütmelerinde yol gösterici olacaktır.

**Anahtar Kelimeler:** Yazılım Yönetimi, Yazılım Mühendisliği, Yazılım Test Mühendisliği, Süreç İyileştirme, TMM, CMM, Girişimcilik

# Modified TMM for Small Teams

Eren Narin, Dr. Sadık Eşmeliolu

METU, Informatics Institute, Ankara  
erennarin.92@gmail.com, esadik@metu.edu.tr

**Abstract.** The recent development in technology makes reuse of information inevitable. The employment of proven standards in software industry decreases cost of projects and sharply increases possibility of success. However, most of the standards have been shaped for large organizations. In the age of the entrepreneurship, small organizations are playing big role in the software industry. Therefore, this work aims to create a new modified Test Maturity Model (TMM) for small teams. This work is an initial step for creating comprehensive quality assurance guide for small teams. TMM v2.0 have used as reference document. All practices which are under of Level 2 and Level 3 have been analyzed and their cost/effect ratios have been determined by using experience and academic knowledge. These ratios are supported with four different case studies. Small Technopolis companies are selected for case studies. At the end of the work, analyzed practices have been clustered according to their priorities. These clusters serve as new maturity levels for small teams. With new level definitions, small teams can implement test practices more efficiently.

**Keywords:** Software Management, Software Engineering, Software Test Engineering, Process Improvement, TMM, CMM, Entrepreneurship

# 1 Giriş

## 1.1 Genel Açıklama

Yazılım endüstrisinde gelinen nokta, bilginin tekrar kullanımını kaçınılmaz hale getirmiştir. Özellikle uluslararası geçerliliği olan standartların süreçlerde kullanımı, projelerin maliyetlerini düşürmekte ve başarı olasılığını keskin bir şekilde arttırmaktadır. Ancak, kendini ispat etmiş ve uluslararası geçerliliğe ulaşmış standartların çoğu, büyük ölçekli organizasyonların bakış açısından geliştirilmiştir.

Dünya genelinde en yaygın kullanılan yazılım standartlarından biri Yetenek Olgunluk Modeli'dir (Capability Maturity Model). Bizim bu çalışmada konu aldığımız Test Olgunluk Modeli de (Test Maturity Model) Yetenek Olgunluk Modeli'nden türetilmiştir. Ancak en geniş kabul görmüş bu yazılım standartları, büyük projeler ve geniş takımlar düşünülerek geliştirilmiştir. Bu standartları eksiksiz yerine getirmeye çalışmak, eforlarının çoğunu varlıklarını sürdürmeye harcayan küçük işletmeler için çok maliyetli olabilir. Bir diğer açıdan standartların her bir pratiğinin uygulanmasının, küçük işletmeler için vazgeçilmez olduğunu söylemek de çok doğru bir yaklaşım olmayabilir. Bütün bu sebeplerden dolayı biz, bu çalışmamızda, küçük işletmelerin optimum eforla maksimum faydayı sağlayabilecekleri bir "Özelleşmiş Test Olgunluk Modeli" geliştirmeyi hedefledik. Bu hedefimiz aynı zamanda, küçük işletmelerin gereksinim, tasarım, kod geliştirme gibi bütün süreçlerini kapsayacak bir rehber oluşturma hedefimizin de ilk adımıdır.

## 1.2 Kısaltmalar

CMM: Capability Maturity Model (Yetenek Olgunluk Modeli)

TMM: Test Maturity Model (Test Olgunluk Modeli)

## 1.3 Doküman Organizasyonu

Bu dokümanın ilk başlığı yapılan çalışmalarla ilgili kısa bilgileri ve kısaltmaları içermektedir. İkinci başlıkta benzer akademik çalışmalardan bahsedilmiştir. "Küçük Takım" tanımı ve bu tanımın kaynağı da bu başlıkta verilmiştir. Üçüncü başlık, bu çalışmaya başlama motivasyonumuzu ve hedeflerimizi içermektedir. Çalışma sırasında izlediğimiz yol ise dördüncü başlıkta açıklanmıştır. Çalışma sırasında uyguladığımız vaka çalışmaları ile ilgili bütün detaylar beşinci başlıkta yer almaktadır. Altıncı ve son başlıkta ise çalışmamızın sonuçları ve gelecek çalışmalar tartışılmıştır.

# 2 Teknik Altyapı ve Tanımlar

## 2.1 Literatür Taraması

Literatürde, küçük takımlara bir rehber oluşturabilecek kadar çok çalışma bulunmamaktadır. Küçük yazılım takımlarına yönelik öncü çalışmalardan biri, Mark C. Paulk'un "Using Software CMM in Small Organizations (CMM'nin Küçük

İşletmelerde Uygulanması)” [1] çalışmasıdır. Bu çalışmada Paulk, “küçük” kelimesinin karşılığını ve bu karşılığa etki eden kişi sayısı, zaman, proje büyüklüğü, ürün kritikliği gibi parametreleri incelemektedir. Ayrıca, bizim bu çalışmadaki amaçlarımıza benzer olarak, CMM’nin bütün pratikleri küçük organizasyonlarda uygulanmalı mıdır ve “iyi süreç” tanımı organizasyondan organizasyona değişken midir gibi soruların cevaplarını da aramaktadır. Bu alandaki bir diğer öncü çalışma ise, A. Laryd and T. Orci’nin “Dynamic CMM for Small Organizations (Küçük Organizasyonlar için Dinamik CMM)” [2] çalışmasıdır. Laryd ve Orci bu çalışmada küçük yazılım organizasyonlarının karşılaştıkları zorluklardan bahsetmektedirler. Aynı zamanda küçük organizasyonların yazılım endüstrisindeki önemini vurgulayarak küçük organizasyonların süreçlerinin iyileştirilmesinin bütün endüstriye katkısı olacağını savunmaktadırlar. Bunun sonucunda da “Dinamik CMM” ismini verdikleri yeni bir standart ortaya koymaktadırlar.

## 2.2 CMM ve TMM

### CMM

CMM, etkili bir yazılım geliştirme sürecinin anahtar noktalarını tanımlayan bir yazılım geliştirme standardıdır. Yazılım geliştirme ve yazılım bakımıyla ilişkili bütün planlama, mühendislik ve yönetim aktiviteleri CMM’nin kapsamına dahildir. Bu anahtar noktalar, organizasyonun maliyet, takvim, işlev ve kalite gereksinimlerini karşılayabilme kabiliyetini arttırmaktadır [3].

CMM, 5 farklı olgunluk seviyesi içermektedir. Bu seviyelerin kısa tanımları aşağıdaki gibidir [4].

*Seviye 1 – Başlangıç:* Başarının bireysel gayretlere dayandığı, çok az sayıda sürecin tanımlı olduğu seviyedir. Bu seviye hariç diğer bütün seviyelerde o seviyelerin ilgilendiği belirli konular olmasına rağmen bu seviyede yoktur.

*Seviye 2 – Tekrarlanabilir:* Bu seviyede, ilk olarak yazılım gereksinimleri yönetilir ve bu gereksinimler ile ilgili ürünler oluşturulur. Daha önceden hizmet verilen organizasyona karşı verilen taahhütler, bu seviyede yine bu organizasyonun istek ve revizyonlarına uygun şekilde tesis edilir, iki taraf arasında gözden geçirilip kontrol edilir.

*Seviye 3 – Tanımlanmış:* Bu seviyede, isminden de anlaşılacağı üzere tüm süreçler, artık iyice tanımlanmış standartlar, prosedürler, araçlar ve metotlar ile anlatılmıştır. Bu seviyedeki tüm süreçler, 2. seviyeye göre daha kapsamlı ve detaylı olarak tanımlanmıştır.

*Seviye 4 – Yönetilen:* Bu seviye yazılım süreçlerinin, iyileştirmeler öncesi son şeklini almadan, tam anlamıyla yönetildiği ve tamamlandığı yerdir. Bu seviyede çeşitli istatistiksel ve nicel ölçüm teknikleri kullanılarak süreçlerin performansları kontrol edilir ve bu süreçlerin ileri zamanlardaki performansları öngörülebilir. Bu olay 4. seviyeyi 3. seviyeden ayıran en önemli unsurdur.

*Seviye 5 – Optimize Edilen:* 4. seviyede elde edilen veriler ve projelerin tamamlandığı süreçte gelişen teknoloji ile bu seviyede sonuçlandırılan tüm projeleri iyileştirmeye ve projelerden maksimum fayda sağlamaya çalışılır. Bu süreçte tüm organizasyon projeleri iyileştirmeye odaklanmıştır.

## TMM

TMM, ilk olarak Illinois Teknoloji Enstitüsü'nde geliştirilen ve CMM'yi temel alan bir test standardıdır. Test süreçlerinin iyileştirilmesini hedefler. TMM, farklı standartlar ile beraber uygulanabildiği gibi tek başına da uygulanabilir.

TMM de CMM gibi 5 farklı olgunluk seviyesine sahiptir. Bu seviyelerin kısa tanımları aşağıdaki gibidir.

*Seviye 1 – Başlangıç:* Bu seviyede organizasyon test için içgüdüsel yöntemler kullanmaktadır, bu nedenle tekrar edilebilir değildir ve herhangi bir kalite standardı bulunmamaktadır.

*Seviye 2 – Tanımlama:* Bu seviyede test bir süreç olarak tanımlanmıştır, tanımlı stratejiler, planlar bulunmaktadır. Bu seviyede testler, ürünün tamamlanmasıyla değil de gereksinim aşamasından başlar.

*Seviye 3 – Entegrasyon:* Bu seviyede test, yazılım döngüsüne entegre edilmiştir (Örn; V Model). Yapılan testler, risk yönetimine dayanmakta ve yazılım geliştirmeden bağımsızlaşmıştır.

*Seviye 4 – Yönetim ve Ölçüm:* Bu seviyede test, gereksinim gözden geçirmeleri ve tasarım da dahil olmak üzere yazılım yaşam döngüsünün her aşamasında yer almaktadır. Dahili ve harici kalite kriterleri bütün ürünler için ortaktır.

*Seviye 5 – Optimizasyon:* Bu seviyede, test sürecinin kendisi test edilir ve her döngüde süreç iyileştirilir. Bu seviyede hata önlemek, hata tespit etmekten daha önceliklidir.

## 2.3 CMM ve TMM

“Küçük yazılım takımı” kavramı, birçok farklı akademik konferansta ve birçok farklı akademik çalışmada tartışılmıştır. Bu tanımın birçok parametreye bağlı olarak değişebilmesine karşın, literatürde yapılan küçük takım tanımlarının çoğu birbirine yakındır. Yapılan bu tanımlar içerisinde, bizim çalışmamıza ve Türkiye'deki yazılım endüstrisine en iyi uyum sağlayan tanımı Mark C. Paulk yapmıştır. Bu tanıma göre küçük takım sınıflandırmaları aşağıdaki gibidir.

**Tablo 1.** Yazılım takımlarının büyüklük kategorizasyonu [1]

“Küçük” Tanımı	Ekipteki Kişi Sayısı
Küçük	4-5
Çok Küçük	3
Ufak	2

## 3 Motivasyon ve Hedefler

“Girişimcilik Çağı” olarak adlandırılan bu çağda, küçük organizasyonların Pazar payı günden güne artmaktadır. Özellikle yazılım endüstrisinde, küçük girişimler teknolojinin gelişmesinde oldukça kritik bir yol oynamaktadırlar. Bunlara rağmen, küçük takımların organizasyon süreçlerini iyileştirmeyi hedefleyen çok az akademik çalışma bulunmaktadır. Yazılım Mühendisliği disiplininin ilk olduğu dönemlere

dayanan bir kültürün etkisiyle, günümüzde hala kapsamlı ve hantal diyebileceğimiz standartlar baskındır. TMM de bu kapsamlı ve hantal standartlardan biridir.

Kaliteyi arttırmak için standartlar hala en iyi seçeneklerdir. Ancak küçük takımlar, söz konusu standartlardan faydalanmamalarının getireceği olumsuzlukları, çevik organizasyon yapıları ve çalışkanlıkları ile aşabileceklerini düşünmektedirler. Bu düşüncenin haklı yanları ile birlikte bu mantalite ile çalışmanın neden olacağı ekstra maliyeti de tartışmak gereklidir. Bu ekstra maliyet, küçük takımların kolay adapte olabilecekleri yeni bir standart tanımlamasını gerekli hale getirmektedir.

Bu çalışmada hedefimiz, küçük takımlar için daha kullanılabilir ve daha kolay anlaşılabilir bir özelleşmiş TMM standardı oluşturabilmektedir. Uzun vadede hedefimiz ise, küçük yazılım takımlarının bütün süreçlerinde faydalanabilecekleri kapsamlı bir kalite rehberi oluşturabilmektir. Bu çalışma da uzun vadeli hedefimizin ilk adımıdır. Bu çalışmanın ardından planladığımız kısa vadeli çalışmalar ise bu çalışmanın kapsamını genişletmeye ve doğruluğunu arttırmaya yönelik çalışmalardır.

## **4 Yöntem**

### **4.1 Pratik Seçimi**

Bu çalışmada, yeniden değerlendirme ve sınıflandırma çalışmaları için TMM standardının 2. ve 3. seviyeleri altında tanımlanan pratikler seçilmiştir. Bu iki seviye, temel yazılım geliştirme süreçlerini düzenlemeyi ve devamlı bir kalite standardı sağlamayı hedeflemektedir. TMM standardının hiçbir seviyesi bir diğerinden daha önemli değildir ancak TMM 4. ve 5. seviye pratiklerinin küçük takımlar için önemi daha düşüktür. Bu pratikler, daha fazla tecrübe ve daha fazla iş gücü gerektirmekle beraber, düzenli bir kalite sağlandıktan sonra bu kalitenin daha da artırılmasını ve süreçlerin iyileştirilmesinin hedeflemektedirler. Bu tarz çalışmaların önceliği daha düşüktür ve bizim amacımız küçük takımların minimum efor ile önceliği yüksek pratikleri uygulayabilmeleridir. Bu nedenle ilk planda yeniden değerlendirme ve sınıflandırma çalışmaları için TMM standardının 2. ve 3. seviyeleri altındaki pratikler seçilmiştir.

### **4.2 Pratik Değerlendirmesi**

Pratik seçiminden sonra bütün seçilen pratikler fayda/maliyet oranlarına göre derecelendirilmişlerdir. Bu derecelendirme işlemi sırasında endüstriyel tecrübelerden ve akademik birikimlerden faydalanılmıştır. Fayda/maliyet oranlarının tespit edilmesi süresince kişisel görüşler, vaka çalışmaları ile de desteklenmiştir.

Fayda ve maliyet katsayıları her pratik için 4 farklı seviyede değerlendirilmiştir; düşük, orta ve yüksek. Fayda ve maliyet katsayıları birbirlerinden bağımsız bir şekilde belirlenmiş, derecelendirme işleminin sonunda bileşik bir puan ilişkili pratiğe verilmiştir. Aşağıdaki tablo, fayda ve maliyet katsayılarına göre pratiklere verilen puan karşılıklarını göstermektedir.

**Tablo 2.** Fayda ve maliyet katsayılarına göre pratiklere atanan puanlar

<b>Fayda</b>	<b>Maliyet</b>	<b>Puan</b>
Düşük	Yüksek	1
Düşük	Orta	2
Düşük	Düşük	3
Orta	Yüksek	4
Orta	Orta	5
Orta	Düşük	6
Yüksek	Yüksek	7
Yüksek	Orta	8
Yüksek	Düşük	9

Bu puanlama işlemi süresince, küçük takımlar tarafından geliştirilmiş 4 farklı başarılı proje vaka çalışmaları kapsamında incelendi. Bu projelerin ortak aktivitelerinin fayda katsayıları yükselttilerek puanları artırıldı.

Maliyet katsayılarının doğru karşılığını belirlemek için vaka çalışmaları kullanılmadı. Projelerdeki benzer aktivitelerin birbirinden çok farklı sürelerde tamamlanmış olduğunun gözlemlenmesi üzerine, aşağıdaki maliyet katsayıları tablosu, endüstriyel tecrübelerden faydalanılarak belirlendi.

**Tablo 3.** Maliyet derecelendirilmesinde kullanılan maliyet katsayısı sınıflandırmaları

<b>Efor</b>	<b>Maliyet Katsayısı</b>
<b>Efor</b> < 2-3 Adam Saat	Düşük
2-3 < <b>Efor</b> < 10-12 Adam Saat	Orta
<b>Efor</b> > 10-12 Adam Saat	Yüksek

#### 4.3 Pratik Önceliklendirmesi

Puanlama işleminin ardından her pratik, Tablo 2’deki kriterlere bağlı olarak bir öncelik puanına sahip olmuş oldu. Önceliklendirmenin ardından yeniden gruplandırma, “Özel Amaç (Specific Goal)” seviyesinde yapıldı. Bu nedenle özel amaç başlığının altında yer alan pratiklerin öncelik puanlarının ortalaması, özel amaç başlığının öncelik puanı olarak atandı. Ardından bu puanları kullanarak eleme yapabilmek için bir eşik değeri belirlenmesi gerekiyordu. Bu eşik değeri organizasyonun büyüklüğü, ekibin tecrübesi, projenin zorluk derecesi gibi birçok farklı parametreye göre farklılık gösterebilir. Bu eşik değerini organizasyonlar kendi politikaları doğrultusunda değiştirerek, TMM’ye harcaacakları eforu değiştirebilirler. Bizim burada birincil hedefimiz, TMM’nin bütün pratiklerini kritiklik seviyelerine göre sıralayarak, farklı karakteristik özelliklerdeki organizasyonların farklı eşikler belirleyerek kendilerine en uygun TMM tanımını oluşturabilmeleridir.

Sektör tecrübelerimize ve vaka çalışmaları sırasında yaptığımız gözlemlere dayanarak, bu çalışmada, pratik elemesinde kullanacağımız eşik değerini 5,25 olarak belirledik. Bu değer, tamamen puanlama sonrasında yapılan gözlemlere dayanan teorik

bir deęerdir, deęiřtirilmesi bu alıřmanın ana ıktılarını etkilemez. zel amaların son puanları ve elenme durumları Tablo 4 ve Tablo 5'te verilmiřtir.

**Tablo 4.** TMM Seviye 2 zel amalar ncelik puanlandırması

<b>zel Ama</b>	<b>Puan</b>	<b>Elenme Durumu</b>
Test Politikası Oluřturma	6,0	Elenmedi
Test Stratejisi Kurma	5,6	Elenmedi
Performans Gstergelerini Oluřturma	3,0	Elendi
rn Risk Deęerlendirmesi Yapma	5,0	Elendi
Test Yaklařımı Oluřturma	5,8	Elenmedi
Test Tehminlerini Belirleme	4,3	Elendi
Test Planı Geliřtirme	5,4	Elenmedi
Plana Baęlılıęı Takip Etme	3,3	Elendi
Sreci İzleme	5,6	Elenmedi
rn Kalitesini İzleme	4,6	Elendi
Dzeltici Eylemler	4,3	Elendi
Test Analizi ve Tasarımı	6,5	Elenmedi
Test Uygulamasının Gerekleřtirilmesi	4,2	Elendi
Test Yrtme Faaliyetleri	4,7	Elendi
Test Bileřenlerinin Ynetimi	3,9	Elendi
Ortam Gereksinimlerinin Geliřtirilmesi	3,0	Elendi
Ortam Gereksinimlerinin Uygulanması	3,7	Elendi
Ortamların Ynetilmesi ve Kontrol	4,7	Elendi

**Tablo 5.** TMM Seviye 3 zel amalar ncelik puanlandırması

<b>zel Ama</b>	<b>Puan</b>	<b>Elenme Durumu</b>
Test Organizasyonu Kurma	3,3	Elendi
Test Prosedrlerini Geliřtirme	5,6	Elenmedi
Test Kariyer Yollarını Kurma	2,0	Elendi
Test Sreci İyileřtirmelerini Uygulama	5,0	Elendi
đrenilen Dersleri Sonraki Test Srelerine Dahil Etme	2,6	Elendi
Test Eđitimi Planlaması Yapma	2,5	Elendi
Test Eđitimleri Saęlama	5,3	Elenmedi
Organizasyonel Test Sreci Varlıklarını Kurma	2,5	Elendi
Test Modellerini Geliřtirme Modellerine Entegre Etme	3,3	Elendi



Ana Test Planı Oluřturma	5,1	Elendi
Ürünün İşlevsel Olmayan Risklerini Analiz Etme	5,0	Elendi
İşlevsel Olmayan Gereksinimler için Test Yaklaşımı Belirleme	5,6	Elenmedi
İşlevsel Olmayan Gereksinimler için Belirlenen Testleri Analiz Etme ve Tasarlama	5,8	Elenmedi
İşlevsel Olmayan Gereksinimler için Belirlenen Testleri Gerçekleştirme	5,0	Elendi
İşlevsel Olmayan Gereksinimler için Belirlenen Testleri Uygulama	6,3	Elenmedi
Denk Gözden Geçirmelerin Planlanması	6,0	Elenmedi
Denk Gözden Geçirmelerin Uygulanması	6,3	Elenmedi

## 5 Vaka Çalışmaları

### 5.1 Vaka Çalışmalarının Uygulanması

Vaka çalışmaları için, 4 farklı küçük ölçekli organizasyonun 4 farklı başarılı projesi seçildi. Vaka çalışmalarının temel amacı, küçük ölçekli şirketler tarafından başarıyla tamamlanmış projelerin ortak aktivitelerini tespit edebilmektir. Başarılı projelerin ortak aktiviteleriyle ilişkili pratiklerin daha faydalı olduğu görüşüyle bu pratiklerin fayda katsayıları daha yüksek olarak belirlendi. Ortak aktiviteleri belirleyebilmek için TMM'nin bütün pratikleri bir anket formu haline getirildi ve katılımcılar her bir pratik için "Uygulandı" veya "Uygulanmadı" cevabı verdiler.

### 5.2 Organizasyon Profilleri

Vaka çalışmalarına katılan her organizasyon, Ankara'daki farklı teknokent yerleşkelerinde faaliyetlerini sürdürmektedirler. Organizasyonların yazılım ekiplerinin büyüklükleri organizasyon büyüklükleri olarak kabul edildi. Katılımcı organizasyonların genel bilgileri aşağıdaki gibidir.

**Tablo 6.** Vaka çalışmalarına katılan organizasyonların büyüklükleri

Organizasyon	Büyükölük (Kiři)	Kategori
VÇ1	3	Çok Küçük
VÇ2	5	Küçük

VÇ3	2	Ufak
VÇ4	4	Küçük

Organizasyonların proje veya takım liderleri organizasyonlarını temsilen vaka çalışmalarına katıldılar. Organizasyon temsilcilerinin genel bilgileri aşağıdaki gibidir.

**Tablo 7.** Vaka çalışmalarına katılan temsilcilerin genel bilgileri

Pozisyon	Eğitim Alanı	Eğitim Seviyesi	Tecrübe (Yıl)
Proje Yöneticisi	Bilgisayar Mühendisliği	Yüksek Lisans	4
Genel Müdür	Bilgisayar Mühendisliği	Yüksek Lisans	8
Genel Müdür	Fizik Mühendisliği	Lisans	10
Genel Müdür	İktisat	Lisans	12

Vaka çalışmalarında konu alınan projelerin genel bilgileri de aşağıdaki gibidir.

**Tablo 8.** Vaka çalışmalarında konu alınan projelerin genel özellikleri

Sektör	Proje Süresi
Enerji	~6 ay
Üretim	~4 ay
Enerji	~3 ay
E-Kitap	~6 ay

### 5.3 Vaka Çalışması Çıktıları

TMM toplantılar, gözden geçirmeler gibi birçok kalabalık hareket etmeyi gerektiren etkinlik de içermektedir. Ancak bu etkinliklerin birçoğu, küçük şirketlerde, ekip üyeleri masalarını bile terk etmeden uygulanabilir. Hatta masadan masaya yapılan tartışmalar, beyin fırtınaları, detaylı planlanmış toplantılardan çok daha verimli olabilir. Bu nedenle TMM’de yer alan bireysel aktiviteler, küçük takımların gözünde ekip aktivitelerinden daha önceliklidir.

Vaka çalışmaları sırasında bir diğer göze çarpan nokta ise görev dağılımıdır. Küçük ekipler genellikle çok keskin görev tanımlarına ve iş dağılımlarına sahip değildirler. TMM, bütün iş paketlerini, projenin erken safhalarında belirleyip, görev dağılımlarının yapılmasının önermektedir. Ancak bu çalışma şekli, küçük ekipler için uygun olmayabilir. Küçük ekiplerin iş güçlerinin az olmasının yanı sıra, genellikle ekip üyelerinin de benzer kabiliyetlere sahip oldukları gözlemlenmektedir. Bu nedenle her işle herkes ilgilenebilmektedir. Bu sebeple, bütün iş paketleri projenin erken safhalarında belirlense bile, görev dağılımlarının gün gün yapılması küçük ekipler için daha verimli olabilir.

Sürdürülebilir ve iyileştirilebilir bir kalite standardı sağlayabilmek, TMM’nin temel amaçlarından biridir. Bu amaca yönelik bütün etkinlikler sadece büyük takımlar için değil küçük takımlar için de oldukça önemlidir. Buna rağmen, bu süreç iyileştirme etkinliklerinin önceliği küçük takımlar için daha düşük olabilir. Sürekli iyileştirme

endişesinin taşınması ve bu doğrultuda adımların atılması her büyüklükteki yazılım şirketleri için önemlidir. Ancak, TMM'nin içerdiği süreç iyileştirme etkinlikleri çok detaylı ve maliyetlidir. Küçük takımlar bu etkinlikleri uyguladıklarında elde edecekleri faydayı, çok daha az eforla, bir araya getirilmiş ve sadeleştirilmiş etkinliklerle sağlayabilirler. Bu nedenle bu amaca yönelik pratiklerin, daha yüksek olgunluk seviyelerine ait olması verimi arttırabilir.

## 6 Sonuç ve Gelecek Çalışmalar

Günümüzde yazılım endüstrisi, ürünlerin ve süreçlerin kalitesini arttırmak için göz ardı edilemeyecek miktarda efor sarf etmektedir. Kullanıcı beklentileri ve projelerin büyüklükleri ile karmaşıklıkları günden güne büyürken, iyileştirme yapmak da günden güne zorlaşmaktadır. Çeşitli girişimlerde ileriye dönük cesaret verici sonuçlar gözlenirse de yazılım endüstrisi hala sıfır hatadan çok uzaktadır [5].

Bütün standartlar tekrar eden tecrübelerin ve kazanımların bir ürünüdür. Bu nedenle genel kabul görmüş standartları kullanmak, sürekli bir kalite sağlayabilmenin en hızlı yoludur. Ancak standartların bütün organizasyonlara eşit derecede uygun olduğunu söylemek doğru değildir. Bütün standartlar kendi alanlarında detaylı iş paketi tanımlarına sahiptir ve bir organizasyonun bu standartların bütün yönlendirmelerini takip etmeleri çok fazla iş gücüne sahip olmalarını gerektirmektedir. Bu nedenle küçük bir organizasyon için bu standartları uygulamak demek asıl geliştirme işlemlerine vakit ayıramamaları anlamına gelebilir. Bütün bunlar, standartların ana amacını kaybetmeden hazırlanmış ve daha az iş gücü gerektiren bir alt standart tanımının olmasını gereklilik haline getirmektedir.

Kalite tanımının neredeyse bütün organizasyonlar için aynı olmasının yanı sıra kaliteyi sağlama yöntemi farklılık gösterebilir. Takımın büyüklüğü, takımın tecrübesi ve takım uyumu gibi parametreler kaliteyi sağlama yöntemlerini farklılaştırabilir. Biz bu çalışmada, takım büyüklüğünün kaliteyi sağlama yöntemi üzerindeki etkisini ele aldık.

Küçük organizasyonların pazar payları gün geçtikçe büyümektedir. Piyasa da girişimciliği desteklemekte ve her yıl yeni birçok yazılım şirketi kurulmaya devam etmektedir. Bu sınıfa dahil olan organizasyonlardaki takımlar genellikle kesin görev dağılımlarına sahip değildir. Standartlar genellikle organizasyonel bir hiyerarşinin varlığını ve kesin görev tanımlarının yapılmasını gerekli kılmaktadır. Bu nedenle standartların önerdiği yöntemler küçük takımların çalışma yöntemleriyle yeterince örtüşmemektedir. Ancak küçük takımların da kendi avantajları bulunmaktadır. Büyük organizasyonlarda çok fazla iş gücü gerektiren toplantılar veya gözden geçirmeler, küçük takımlarda çok daha hızlı ve etkili yapılabilmektedir. Bu gerçeği de yaptığımız pratik önceliklendirmesinde göz önünde bulundurduk.

Yapılan pratik değerlendirmelerinde endüstriyel tecrübelerden ve akademik birikimden faydalanıldı. Ancak yeni bir alt standart tanımının yapılabilmesi için tecrübe ve akademik birikim yeterli değildir. Bu nedenle çalışma süresince ortaya atılan görüşler, uygulanan vaka çalışmaları ile desteklendi. TMM'nin 2. ve 3. seviyeleri altında yer alan pratikler bir anket formu haline getirildi. Vaka çalışmasına katılan

organizasyonlardan ilgili pratikleri kendi projelerinde uygulayıp uygulamadıklarını beyan etmeleri istendi. Bu işlem katılımcılara bire bir röportaj yöntemi ile uygulandı. Vaka çalışmalarının sonucunda pratik değerlendirmeleri tekrar düzenlendi ve başarılı projelerin ortak pratiklerinin etki katsayıları yükseltildi. Bu yöntem ile Türkiye pazarında yer alan küçük organizasyonlara has değişkenler değerlendirme sonuçlarına yansıtılmış oldu.

Bu çalışmaların sonucunda, küçük takımlar için özelleşmiş bir TMM tanımı hazırladık. Çalışmaların tamamı, küçük takımların ve girişimcilerin gözünden yürütüldü. Hacettepe Teknokent’te ve ODTÜ Teknokent’te yer alan girişimcilerin görüşlerinden de çalışmaların yöntemine karar verilirken faydalanıldı. Ancak çalışma süresince görüşülen küçük organizasyonların sayısı oldukça kısıtlıdır. Uygulanan vaka çalışmalarının sayısını arttırarak “Özelleşmiş TMM”nin kapsamının genişletilmesi ve doğruluğunun artırılması, ileriye dönük hedeflerimizdendir.

Gelecek çalışmalarda hedefimiz, küçük yazılım takımlarının gereksinim, tasarım, geliştirme gibi bütün süreçlerinde takip edebilecekleri detaylı bir kalite rehberi hazırlamaktır. “Özelleşmiş TMM” bu hedefimizin ilk adımıdır.

## **Kaynakça**

1. Paulk, Mark C.: Using the Software CMM in Small Organizations (1998).
2. Laryd, A., Orci, T.: Dynamic CMM for Small Organizations (1999).
3. Kumta, M. D., Gita A.: Capability Maturity Model (2002).
4. Paulk, Mark C.: The Capability Maturity Model: A Summary (1999).
5. Veenendaal, E. v.: Test Maturity Model Integration (TMMi) (2012)