



T.C. SANAYİ VE  
TEKNOLOJİ BAKANLIĞI

#MILLİ  
TEKNOLOJİ  
HAMLESİ



# ÖLÇME VE ANALİZ FAALİYETLERİNİN SÜREÇ İYİLEŞTİRMEDEKİ YERİ

ARAŞTIRMA SERİSİ - SAYI 12



BİLGEM

YAZILIM TEKNOLOJİLERİ ARAŞTIRMA ENSTİTÜSÜ

# Simge ve Kısaltmalar

Kısaltmalar	Açıklama
TÜBİTAK	Türkiye Bilimsel ve Teknolojik Araştırma Kurumu
BİLGEM	Bilişim ve Bilgi Güvenliği İleri Teknolojiler Araştırma Merkezi
YTE	Yazılım Teknolojileri Araştırma Enstitüsü
CMMI	Capability Maturity Model Integration (Yetenek Olgunluk Model Entegrasyonu)
ISO	International Organization for Standardization (Uluslararası Standart Organizasyonu)
PDCA	Plan-Do-Control-Act (Planla-Uygula-Kontrol Et-Önlem A)
SPICE	Software Process Improvement and Capability dEtermination (Yazılım Süreci İyileştirme ve Yetenek Belirleme)
YSİ	Yazılım Süreç İyileştirme

## Yazar

Melike TAKIL

## Yayın Koordinatörü

Kübra ERTÜRK

## Editörler

Beyza Nur ERCAN

Beyza ŞENEL

Tuğçe YILMAZ

## Tasarım

Şeyma KOÇER

©2024 - Tüm hakları saklıdır.

İletişim: 0(312) 289 92 22 - yte.bilgi@tubitak.gov.tr

<https://bilgem.tubitak.gov.tr/yte/>

Yayınlanan yazıların sorumluluğu yazarına aittir, TÜBİTAK BİLGEM sorumlu tutulamaz.

# İçindekiler

Ön Söz	4
Giriş	5
Yazılım Mühendisliği Süreçleri ve Süreç Yönetimi	6
Süreçlerin Ölçümü	8
Proje İzleme ve Değerlendirme	10
Yazılımda Süreç İyileştirme	12
Sonuç ve Öneriler	14
Kaynakça	15

# Ön Söz

TÜBİTAK BİLGEM Yazılım Teknolojileri Araştırma Enstitüsü (YTE), 2012 yılından bu yana yazılım teknolojilerinde Ar-Ge faaliyetleri yürüten bir araştırma kuruluşudur. Araştırma faaliyetlerinde elde ettiği birikimini stratejik, hassas ve kritik projeler yürüterek kamu adına hayata geçirmekte; kurumlarımıza dijital dönüşüm, yazılım geliştirme teknolojileri ve kalite süreçleri konusunda danışmanlık vermektedir.

TÜBİTAK BİLGEM YTE tarafından hazırlanan Araştırma Serisi ile kurum içi içerik üretme çalışmalarının yaygınlaştırılması ve hazırlanan içeriklerin sektörün erişimine açılması amaçlanmaktadır. Araştırma Serisi'nde yayınlanan çalışmalar TÜBİTAK BİLGEM YTE çalışanlarının projelerde elde ettiği bilgi birikimini paylaşmak adına derlenmiştir. Bu çalışmalar ile ülkemizin yazılım sektörüne katkı sağlanması hedeflenmektedir.

# Giriş

Yazılım ürünlerinin endüstrideki ve günlük yaşantımızdaki giderek artan önemiyle birlikte, yazılım mühendisliği süreçleri son yıllarda hem yazılım mühendisliği araştırmacıları hem de uygulayıcıları arasında büyük ilgi çekmektedir. Yazılım mühendisliği süreçleri, insan odaklı faaliyetler olduğundan beklenmeyen veya istenmeyen performans ve davranışlarla karşılaşma eğilimindedir. Bu doğrultuda, bir organizasyonun müşteri ve paydaş ihtiyaçlarını karşılamak adına yazılım süreçlerinin düzenli olarak değerlendirilip geliştirilmesi gerekliliği genel olarak kabul görmektedir. Sistematik iyileştirmeye yönelik olan Yazılım Süreç İyileştirme (YSİ), endüstride geniş yer bulan bir metodolojidir. YSİ, yazılım geliştirme süreç ve uygulamalarının değerlendirilmesini ve iyileştirilmesini kapsar.

Yazılım süreçlerinin ölçümü, öngörülebilir performans elde etme, yüksek olgunluğa ulaşma ve belirlenen kalite gereksinimlerini karşılama amacı doğrultusunda önemli bir faktördür [1]. Bu nedenle yazılım süreçlerinin ve ürünlerinin iyileştirilmesinde yazılım ölçümünün gerekliliği tüm yazılım paydaşları tarafından kabul edilmektedir. Çünkü süreç ve sonuç, ölçülmez ve değerlendirilmezse yazılım süreç iyileştirme çabası beklenen sonucu vermeyecektir. Yazılım süreç ölçümü, tüm yazılım süreç iyileştirme programlarının vazgeçilmez bir parçasıdır. Bu kapsamda yapılan araştırmalar, süreç iyileştirme çalışmalarının başarısı için ölçümün önemli bir faktör olduğunu göstermektedir [2]. Yazılım ölçüm ve analizi sonucu elde edilen geri bildirimlerin ve iyileştirme etkilerinin değerlendirilmesinin başlıca faydaları şunlardır:

- Sonucu görünür hale getirerek süreç iyileştirme çalışmalarına motivasyon sağlar ve ölçüm gerekliliğini haklı çıkarır.
- Yazılım süreç iyileştirme strateji ve yöntemlerinin değerlendirilmesine olanak tanır.

Bununla birlikte kararların dayandırılabilmesi için geçerli bilgileri sağlayan bir ölçüm programı oluşturmak ve uygulamak zordur. Neyin ölçülmesi gerektiği konusunda çok az fikir birliği vardır. Sistematik ve güvenilir bir ölçüm yaklaşımının olmayışı, iyileştirme çalışmalarının yüksek başarısızlıkla sonuçlanmasına sebep olan bir faktör olarak kabul edilmektedir [3]. Yazılım süreç iyileştirme çalışmalarının değerlendirilmesindeki bu sorunlardan bağımsız olarak, iyileştirme çabalarının fayda sağladığını gösteren çok sayıda kanıt da mevcuttur [4]. Bu çalışma kapsamında, yazılım geliştiren organizasyonlardaki süreç yönetimi, ölçme ve analiz konuları incelenerek süreç iyileştirme açısından ölçme ve analiz çalışmaları detaylandırılacaktır.



# Yazılım Mühendisliği Süreçleri ve Süreç Yönetimi

Yazılım geliştiren organizasyonlar, yazılım proje faaliyetlerini desteklemek, yönlendirmek ve gerektiğinde kontrol etmek için yazılım mühendisliği süreçlerini bir araç olarak benimsemektedirler. Bu nedenle birçok organizasyon, faaliyetlerini destekleyecek süreçleri uygulamak amacıyla, iyileştirme programlarına önemli oranda yatırım yapmaktadırlar [4]. Bu yatırımlar; yazılım mühendisliği süreç tanımlarının oluşturulması, standartlaşması, değerlendirilmesi ve iyileştirilmesine yönelik yöntem ve araçların çeşitliliğini sağlamaktadır.

Yazılım mühendisliği yönetimine bu şekilde süreç tabanlı yaklaşılmasının temel sebebi ise süreç kalitesinin, üretilen yazılımın kalitesini, maliyetini ve yayınlanma süresini etkilemesidir. Dolayısıyla tanımlı ve yönetilen süreçlerin kullanımıyla, yazılım ürün kalitesinin sağlanması hedeflenmektedir. Bu noktada süreçlerden, süreçleri iyileştirmek amacıyla olgunluk modellerinin kullanımından, yine süreçleri iyileştirmek amacıyla ölçümün bir süreç olarak kullanımından bahsedilebilir.

Yazılım süreç araştırmaları, yazılım süreçlerinin belirlenmesine, standartlaştırılmasına, değerlendirilmesine ve iyileştirilmesine yönelik yöntemlerin geliştirilmesi ve uygulanmasıyla ilgilenen bir alan olarak tanımlanmaktadır [5]. Yazılım süreç araştırmalarının temel amacı, belirli bir yazılım mühendisliği sistemine göre süreçleri tanımlayarak uygulanmasını sağlamaktır. Bu araştırmaların etkili olabilmesi için yazılım süreç mühendisliği yöntemlerinin uygulanan sistemle uyumlu olması önemlidir.

Yazılım geliştirmenin tanımlı süreçleri, yazılım sistemlerini tasarlamak, uygulamak, test etmek, dağıtmak ve sürdürmek için takip edilen; iyi dokümanite edilmiş, yapılandırılmış ve tekrarlanabilir aktiviteleri ifade eder. Bu süreçler, yazılım geliştirme yaşam döngüsü boyunca tutarlılık, kalite ve verimliliği sağlamak için önemlidir. Gereksinim analizi, sistem tasarımı, geliştirme, doğrulama ve geçiş gibi süreçler, yazılım geliştirmeye dâhil edilen bazı ortak süreçlerdir. Yazılım geliştirme süreçlerinde olduğu gibi, süreçlerin iyileştirilmesi için gerekli olan ölçüm ve analiz faaliyetleri de bir süreç olarak ele alınmalıdır, yani tanımlanmalı, standartlaştırılmalı, değerlendirilmeli ve sürekli iyileştirilmelidir.



Yazılım süreçlerini yönetmek ve tüm çalışma faaliyetlerini ölçmek için bilgi ve araçlara ihtiyaç duyulmaktadır. Yazılım süreçlerinin sertifikalandırılmasını sağlayan olgunluk modelleri bu doğrultuda fayda sağlayabilmektedir. Bununla birlikte yazılım geliştirmede olgunluk modeli kullanmak isteyen organizasyonlar, sürecin karmaşıklığını ve maliyetini göz önünde bulundurmalıdır.

Olgunluk modellerinin kullanımı, süreç olgunluğunun artırılması yoluyla organizasyon politikalarının ve standartlarının tüm organizasyon içinde yapılandırılmasını sağlar. Bu kapsamda olgunluk modellerini kullanmak, kurumsallığın artırılmasında önemli rol oynar. Örneğin yaygın olarak benimsenen CMMI (Capability Maturity Model Integration), yazılım süreçleri yönetimi ve yazılım süreç ölçümü için rehberlik sağlar [6]. CMMI ve benzeri modeller, yüksek olgunluk seviyesinin ancak ölçme faaliyetleri ile mümkün olduğunu ve süreçlerin optimizasyonu için bir gereklilik olduğunu savunmaktadırlar.

Yazılım geliştirme organizasyonlarının, süreç ölçümlerini bir külfet olarak görmesi sık rastlanan bir durumdur. Ancak ölçüm süreçleri; yazılım kalitesinin, süreçler üzerinden iyileştirilmesi ile organizasyonlara proaktif bir yaklaşım sunar [6]. Bu da süreç içerisinde alınacak önlemler ve aksiyonlar ile kaliteli yazılımın üretilmesini sağlamaktadır. Ölçümlerin, yazılım geliştirme sürecinden elde edilen nicel ve doğru verilere dayanması ise önemli bir noktadır.

Bir ölçüm sürecini uygulamak için öncelikle organizasyonun bilgi taleplerini tespit etmek ve ardından doğru süreçlerden toplanacak doğru ölçümleri belirlemek gereklidir. Yaygın olarak yapılan hatalardan biri, bir ölçüm sürecinin organizasyon açısından gerçek değerini değerlendirmeden karar vermektir.

Bir diğer yaygın hata ise karar desteği sağlamak için ölçüm, veri toplama, bilgi analizi ve raporlama ile ilgili kavramları bilmeyen ve süreç yönetimi konusunda uzmanlığa sahip olmayan profesyonellerin görevlendirilmesidir. Yapılan bu hatalar ölçüm sürecinin başarısını olumsuz olarak etkilemektedir.

Bir ölçüm sürecinin başarılı olabilmesi için, ölçümlerin aşağıdaki gereklilikleri sağlaması gerekmektedir:

- İş hedefleri ile yakından ilişkilendirilmeli,
- Maliyeti ve çabayı haklı çıkaracak gerekçelere, sistematik ve yaygın bir kullanıma sahip olmalı,
- Anlaşılır ve karşılaştırmaya izin verecek şekilde tanımlanmalı ve
- Tarafsız bir şekilde iletilmelidir.

Bir süreç ölçüm sisteminin hedefleri tanımlandıktan sonra aşağıdaki yönleri kapsayan yönetim süreçlerinin tanımlanması gerekmektedir:

- Ölçüm hedefleri, tanımı, amacı ve kapsamı belirtilmeli,
- Metrikler; organizasyonun iş hedefleri, amaçları ve stratejileri ile ilişkilendirilmeli,
- Yazılım projesinin bütünsel sistem görüşü göz önünde bulundurularak metrikler ve ölçümler için teknik yönler net ve eksiksiz bir şekilde tanımlanmalı,
- Ölçüm süreci ve metriklerle ilgili roller ve sorumluluklar tanımlanmalı,

- Sonuçları kullanma, analiz etme, depolama ve iletişim kurma yolları belirlenmeli,
- Süreç, organizasyondaki tüm paydaşları içermeli ve
- Ölçüm sonuçlarının ilgili paydaşlarla paylaşılması gibi güvenli iletişime yönelik politikalar geliştirilmeli ve metrik sonuçlarıyla ilgili aksiyonlar alınmalıdır.

Yazılım süreçleri için mevcut olan olgunluk modellerinde bulunan ölçüm süreci, yazılım geliştiren organizasyonlarda kararları destekleyecek ve süreçlerin iyileştirilmesini sağlayacak ölçümlerle bir süreç ölçüm sistemi yapılandırma temel olarak işlev sağlamaktadır. Her olgunluk düzeyinin hedeflerini karşılamak için gereklilikler farklı olduğundan, ölçümlere yönelik talepler olgunluk düzeyleri ile ilgilidir [7]. Farklı yazılım projelerinden ve süreçlerinden elde edilen ölçümlerin sürekli analizi, projelerde düzeltici eylemleri ve alınacak kararları destekleyecek, rekabet avantajını artıracak bilgiler sağlayabilir.

Tüm yazılım organizasyonlarının organizasyonel seviyelerinde, etkili karar vermeye yönelik ölçüm sürecinin tasarlanması zor bir görev olarak kabul edilmektedir [8]. Ölçüm sürecini tasarlariken sıkça karşılaşılan konuların aşağıdaki gibi olduğu belirtilmektedir:

- Ölçümlerin iş hedefleri ile uyumu ve stratejik planın entegrasyonu
- İyileştirme çalışmalarının faydalarının gerekçelendirilmesi

Karşılaşılan bu konular, kötü tasarlanmış ölçüm süreci, kötü kararlar alınmasına ve ekonomik kayıplara neden olabileceğinden önemlidir. Buna rağmen, bir yazılım organizasyonunun iş stratejisiyle bütünleşik bir yazılım ölçüm süreci kurması halinde; karar alma sürecinin desteklenme olasılığının daha yüksek olduğu görülmüştür [9].

## Süreçlerin Ölçümü

**“ Ölçemezseniz yönetemezsiniz, yönetemezseniz iyileştiremezsiniz. ”**

Bu söz genellikle yazılım geliştiren bir organizasyonun, ölçüm sistemlerinin tanımlamasını ve uygulamasını teşvik etmek için kullanılır. Yazılım geliştirme birçok süreci içerir ve ölçüm, bu süreçlerin karakterize edilmesini, kontrol edilmesini, tahmin edilmesini ve iyileştirilmesini sağlar. İnsan merkezli bir faaliyet olan yazılım süreçleri sorunlara eğilimlidir. Bu da müşterilerin ve paydaşların beklentilerini karşılamak için neden sürekli olarak yazılım süreçlerinin değerlendirilmesi ve iyileştirilmesi gerektiği sorusuna yanıt vermektedir. Yazılım süreç ölçümü, organizasyonlarda istenen performansa, yetkinliğe ve kalite seviyesine ulaşmada yardımcı olmaktadır. Ayrıca yazılım süreçlerinin ölçülmesi, yazılım ürününün kalitesinin öğrenilmesine de olanak tanımaktadır.

Bir yazılım sürecinin kalitesini ve zaman içindeki gelişimini değerlendirmek için belirli temel göstergelerin ölçülmesi, izlenmesi ve karşılaştırılması gerekmektedir. Bu temel göstergeler arasında güvenilirlik, verimlilik, kullanılabilirlik ve sürdürülebilirlik gibi ürün nitelikleri yer almaktadır. Bu ni-



teliklerin değerlendirilmesine yönelik veriler, testlerden, operasyon izlemelerden ve yazılımın statik analizinden alınabilir. Bu, bir sürecin ürettiği ürünün kalitesine göre değerlendirilmesi gerektiği anlamına gelir. Süreci ölçmek için ürünü ölçmek gerekir. Öte yandan üründen alınamayacak; verimlilik, takvime uyum, proje personeli eforları gibi ölçümler de vardır. Örneğin, proje personeli eforu ölçümünün proje verilerinden iş günü cinsinde alınması gerekir. Son olarak anlaşılması ve tanımlanması zor olan başka bir ölçüm daha mevcuttur: Bu, yalnızca ürünün kendisinden değil aynı zamanda sürecin yürütülme biçiminden de etkilenen kullanıcının / müşterinin memnuniyet derecesidir. Belirli periyotlarla kullanıcı veya müşterilere gönderilen anketler aracılığıyla memnuniyet derecesi ölçülebilir. Kullanıcılardan gelen yorumları da nitel olarak değerlendirmek ve analiz sonuçlarına göre gerekli aksiyonları almak müşteri memnuniyeti hususunda önemli bir rol oynar.

Özetle, bir sürecin aşağıdaki etmenler doğrultusunda ölçülebilir olduğu sonucuna varılabilir:

- Ürün kalitesi
- Verimlilik / üretkenlik
- Takvime uyum
- Kullanıcı / müşteri memnuniyeti

Yazılım süreci ölçümü, geliştirme sürecinin, projenin ve ortaya çıkan ürünün; daha iyi anlaşılması, değerlendirilmesi ve kontrol edilmesi için destek sağlamaktadır [10]. Yani müşteriye teslim edilen ürünün kalitesi, bir yazılım projesinin ilerleyişi, maliyeti ve takvim öngörüsüne uyumu hakkında çeşitli bilgiler elde etmek için kullanılabilir. Ölçüm aynı zamanda kuruluşların süreçleri hakkında fikir sahibi olmalarına, kalite ve performansını iyileştirmelerine, kalite ve performansını tahmin etmelerine olanak tanır. Bu da organizasyonlara, geliştirme süreci sırasında mümkün olduğunca erken, uygun ve bilinçli kararlar almak için daha iyi bir ortam sağlar [11].



Yazılımın ölçümü; yazılım süreçlerinin planlanmasını, izlenmesini, kontrol edilmesini ve değerlendirilmesini desteklemektedir. Ölçümlerin kullanılması aşağıdaki motivasyonları sağlar:

- Proje planlama ve tahminleme
- Proje izleme ve değerlendirme
- Kalite ve iş hedeflerini anlama
- Yazılım geliştirme süreçlerini ve araçlarını iyileştirme

Yazılım ölçümlerinin periyodik olarak proje izleme toplantıları aracılığı ile izlenmesi ve değerlendirilmesi, proje ve süreçlerde oluşabilecek sorunları önceden görme kabiliyeti sağlamaktadır. Bu, projeyi etkileyecek kararlar alınmasına olanak tanır ve süreçlerdeki iyileştirmelere öncülük eder.

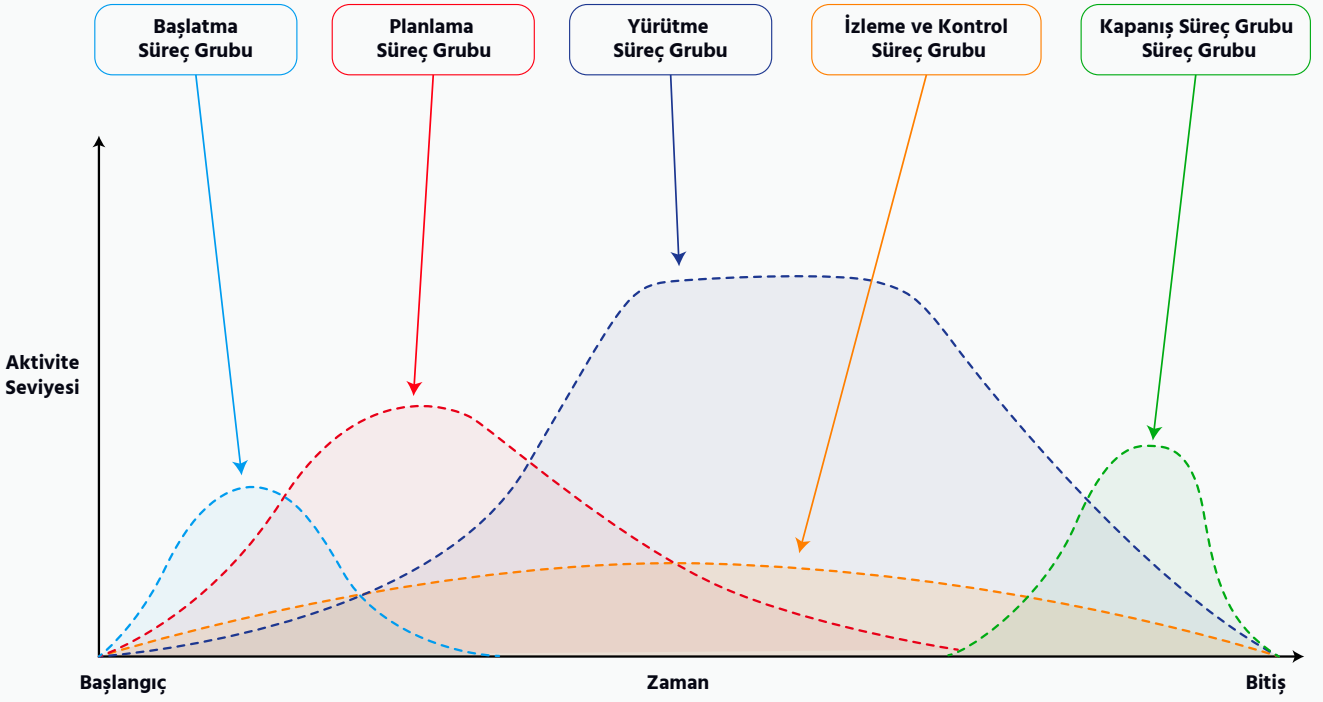
## Proje İzleme ve Değerlendirme

Uluslararası standartlara uygun, mevcut sistem yapılarıyla bütünleştirilebilir, kaliteli ve güvenilir yazılım talebi artarak devam etmektedir. Bununla birlikte artan karmaşıklık ve daha kullanıcı dostu yazılımlara duyulan gereksinim, yazılım üretim ve bakım maliyetlerini önemli ölçüde artırmaktadır. Bu sebeple bu tür yazılımların özelliklerinin değerlendirilmesi ve ölçülmesi büyük bir önem arz etmektedir.

Günümüzün hızla gelişen teknoloji ortamında yazılım projelerinin etkilerini ölçmek ve değerlendirmek gerekliliği oldukça kritik hale gelmiştir. Süreçlerin etkinliğini değerlendirebilmek, geliştirme alanlarını belirleyebilmek ve projelerin etkisini artırmak için güçlü bir izleme ve değerlendirme çerçevesine ihtiyaç duyulmaktadır.

Bir proje genellikle bir dizi hareketli parçanın zamanında bir araya gelmesini gerektirir. Bu parçalardan herhangi birinde yaşanacak bir aksama, gecikmelere ve maddi kayıplara yol açabilir. Her şeyin plana uygun ilerlemesini sağlamak ise oldukça zordur. Bu sebeple, her projenin ilerleyişini takip edecek ve değerlendirecek bir sistem oluşturması önemlidir.

Proje izleme ve değerlendirme; proje yöneticilerinin yazılım geliştirme projelerinin ilerleyişini gözlemlemelerine, performanslarını belirli amaç ve hedeflere göre değerlendirmelerine imkân tanıyan kilit bir süreçler bütünüdür. İzleme ve değerlendirme süreçleri, iyileştirme alanlarını tanımlamak, kaynakların en etkili biçimde kullanılmasını sağlamak ve proje başarısını sağlamak açısından son derece önemlidir. Şekil 1’de gösterildiği gibi izleme ve değerlendirme, proje yaşam döngüsü başlangıcından sonuna kadar devam eder.



Şekil 1. Proje yaşam döngüsünde proje yönetimi süreçleri

Proje izleme, izleme ve kontrol aşamasının izleme kısmıdır. Bu kısım, proje ayrıntılarının (örneğin bütçe, zaman, kapsam vb.) hedeflere göre ölçülerek doğru yolda olduğundan emin olunmasını içerir.

Proje değerlendirmesi, izleme ve kontrol aşamasının kontrol kısmıdır. Toplanılan izleme verilerine dayanarak kararlar almayı içerir. Örneğin, yaklaşan teslim tarihlerine yetişmek için geliştirme faaliyetlerini hızlandırmanın gerekip gerekmediği veya test faaliyetlerine daha fazla çaba harcamanın gerekip gerekmediği gibi hususlardaki kararları içerir.

Projede her şey mükemmel bir şekilde ilerlemeyebilir, bu kaçınılmazdır. Bu noktada proje izleme ve değerlendirme; projenin kapsamını, kalitesini, zamanlamasını veya bütçesini etkileyebilecek sorunların belirlenmesine ve azaltılmasına yardımcı olur. Burada tespit edilen iç görüler, daha sonra gelecekteki projeler için süreçleri iyileştirmek amacıyla kullanılabilir.



# Yazılımda Süreç İyileştirme

Yazılım ürünleri dünya ekonomisinin birçok alanında önemli bir rol oynamaktadır. Yazılım geliştiren organizasyonlar rekabetçi bir pazarda çalışırlar ve sıklıkla rekabet gücünü etkileyecek iş kararları verme konusunda zorlanırlar. Yazılım geliştiren şirketler, pazarda rekabetçi kalabilmek için zorluklarla karşı karşıyadır ve bu, diğer konuların yanı sıra müşterileri memnun etme taahhütlerine bağlıdır.

Yazılım şirketlerinden de rekabetçi kalabilmeleri için çalışma süreçlerinde sürekli iyileştirmeler talep edilmektedir. Sürekli iyileştirmeler, belirlenen süreç iyileştirme çerçeveleri kullanılarak uygulanır. Birçok yazılım süreç iyileştirme çerçevesi ve modelinin fikri PDCA (Plan-Do-Control-Act) döngüsünden kaynaklanmaktadır [12]. Bu döngü aşağıdaki gibidir:

- Bir iyileştirme planı oluşturulur
- Yeni süreç uygulanır
- Değişen süreç ölçülür
- Uygulanan değişikliklerin etkisi analiz edilir [13], [14].

Yazılım geliştirme süreçlerinde olgunluğu sağlamak için olgunluk modellerinin benimsenmesi, işlerin kalitesinin garanti altına alınacağı ve müşteri memnuniyetinin yüksek oranda sağlanacağı düşüncesiyle pazarda tanınmanın bir yolu olarak görülmektedir.

CMMI, SPICE gibi olgunluk modelleri, farklılaştırıcı ve aynı zamanda yazılım süreç kalitesini iyileştirmeyi kolaylaştırıcı birer faktör olarak benimsenmektedir. Bu modeller, yazılım süreçlerini en iyi uygulamalarla tanımlamakta ve sürekli iyileştirmeye odaklanarak operasyonel mükemmelliği hedeflemektedir.



Yapılan bir yazılım süreç araştırmasında süreç kalitesinin, doğrudan geliştirilen yazılımın kalitesiyle ilişkili olduğu yönünde ortak bir varsayıma varılmıştır [15]. Bu doğrultuda yazılım süreci iyileştirmenin amaçlarından bazıları; ürün kalitesini ve hızını artırmak, aynı zamanda ürünün pazara çıkış süresini ve üretim maliyetini azaltmaktır. Buna ek olarak, yazılım sürecinin iyileştirilmesine yatırım yapmanın birkaç temel nedeni şunlardır:

- 1. Gelişmiş İş Birliği ve İletişim:** YSİ; geliştirme ekipleri, ürün sahipleri ve paydaşlar arasındaki iş birliğini ve iletişimi teşvik etmektedir. Organizasyonlar, açık iletişim kültürünü teşvik ederek süreçlerini daha iyi anlayabilir, darboğazları belirleyebilir ve zorluklarla daha etkili bir şekilde başa çıkabilir.
- 2. Kolaylaştırılmış Geliştirme Süreçleri:** YSİ, organizasyonların verimsizlikleri belirlemesine ve yazılım geliştirme süreçlerini kolaylaştırmasına olanak tanımaktadır. Ekipler, fazlalıkları azaltarak ve iş akışlarını optimize ederek zamandan ve kaynaktan tasarruf edebilir. Sonuçta daha hızlı geliştirme döngülerine ve daha yüksek kaliteli yazılım ürünlerine katkıda bulunabilir.
- 3. Geliştirilmiş Kaynak Tahsisi:** İyi uygulanan bir YSİ girişimi, organizasyonların kaynakları daha etkili bir şekilde tahsis etmesine yardımcı olmaktadır. Yönetim, darboğazları ve verimsizlikleri tespit ederek personeli, araçları ve bütçeyi belirli alanlara yönelik olarak daha iyi tahsis edebilir. Bu da daha uygun maliyetli bir geliştirme sürecinin yolunu açar.
- 4. Daha Fazla Uyarlanabilirlik:** Sürekli iyileştirmeye vurgu yapan YSİ, organizasyonların sektördeki değişikliklere karşı daha çevik ve duyarlı olmasını sağlamaktadır. Organizasyonlar, yazılım geliştirme süreçlerini düzenli olarak gözden geçirip iyileştirerek yeni teknolojilere, metodolojilere ve pazar taleplerine kolayca uyum sağlayabilirler.
- 5. Daha İyi Risk Yönetimi:** Yazılım sürecinin iyileştirilmesi, organizasyonların potansiyel riskleri belirlemesine ve bunları proaktif olarak ele almasına yardımcı olmaktadır. Organizasyonlar, sağlam risk yönetimi pratikleri uygulayarak potansiyel sorunları daha iyi tahmin edebilir ve azaltabilir. Böylece daha az aksilikle daha başarılı yazılım geliştirme projeleri elde edilebilir.
- 6. Artan Müşteri Memnuniyeti:** YSİ'ye odaklanmak, müşterilerin ihtiyaç ve beklentilerini daha iyi karşılayan daha kaliteli yazılım ürünlerine yol açabilir. Organizasyonlar, yazılım kalitesini ve işlevselliğini geliştirerek müşteri memnuniyetini artırabilir ve pazardaki itibarını güçlendirebilir.
- 7. Endüstri Standartlarına Uyum:** YSİ girişimlerini benimsemek, genellikle ISO ve SPICE gibi uluslararası standartlara uyum sağlamayı içermektedir. Bu standartlara uygunluk, bir organizasyonun kaliteye, profesyonelliğe ve en iyi uygulamalara bağlılığını göstermektedir; bu da organizasyonun güvenilirliğini artırıp pazar konumunu iyileştirebilir.

Yazılım süreci iyileştirmesine yatırım yapmak, kolaylaştırılmış geliştirme süreçlerinden ve daha iyi kaynak tahsisinden, artan müşteri memnuniyetine kadar organizasyonlara önemli ölçüde fayda sağlamaktadır. Organizasyonlar, YSİ'yi benimseyerek yazılım ürünlerini geliştirebilir, sürekli büyüme ve yenilik kültürünü teşvik edebilir.

# Sonuç ve Öneriler

Yazılımın hayatımızdaki önemi her geçen gün artmaktadır. Önemin artması, bu alandaki kalite artışını zorunlu kılmaktadır. Yazılımda kalite artışının yolu, yazılım geliştirme süreçleri ve bu süreçleri destekleyici süreçlerin iyileştirilmesinden geçmektedir. Organizasyon ve pazar seviyesindeki ihtiyaçların sürekli artması ve değişiyor olması süreçlerin sürekli iyileştirilmesini tetiklemektedir. Sürekli iyileştirme için ölçümler kaçınılmazdır. Ürün ve süreç ölçümleri, organizasyona ait iş hedefleri ile hizalanmış olmalı, tarafsız ve anlaşılır olmalıdır. Ölçüm faaliyetleri için personel kaynağı ayrılması, ölçüm ve raporlama araçlarının kurulması ve kullanılması ve ölçüm için veri girişi sağlanması dolayısıyla harcanan eforların bir maliyeti olduğundan; bu maliyeti ve çabayı haklı çıkaracak gerekçelere, kararlara ve yaygın kullanıma sahip olan ölçümler gerekmektedir. Ölçüm sonuçları analiz edilip değerlendirilerek süreç iyileştirme tetiklenmiş olur. Ölçüm sonuçlarının periyodik olarak değerlendirilmesi önemlidir. Periyodik değerlendirme, proje izleme toplantıları ile sağlanır. Üst yönetimin katılım sağladığı bu toplantılarda ölçümler, kaynak kullanımı, bütçe, süreçlerin iyileştirilmesi ve paydaş yönetimi gibi konularda karar desteği sunar. Tetiklenen süreç iyileştirme çalışmalarının sonuçları da ayrıca değerlendirilir. Süreç iyileştirme çabasını haklı çıkaran çalışmalar, pilot uygulama olmaktan çıkarılarak organizasyon genelinde uygulanır ve böylece organizasyonel mükemmelliğin yolu açılmış olur.

# Kaynakça

---

- [1] W. A. Florac and A. D. Carleton, *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, vol. 43, no. 4, 1999. doi: 10.1198/tech.2001.s56.
- [2] T. Dyba, "An empirical investigation of the key factors for success in software process improvement," *IEEE Trans. Softw. Eng.*, vol. 31, no. 5, pp. 410–424, 2005, doi: 10.1109/TSE.2005.53.
- [3] J. Iversen and O. Ngwenyama, "Problems in measuring effectiveness in software process improvement: A longitudinal study of organizational change at Danske Data," *Int. J. Inf. Manage.*, vol. 26, no. 1, pp. 30–43, 2006, doi: 10.1016/j.jinfomgt.2005.10.006.
- [4] G. Canfora, F. García, M. Piattini, F. Ruiz, and C. A. Visaggio, "Applying a framework for the improvement of software process maturity," *Softw. - Pract. Exp.*, vol. 36, no. 3, pp. 283–304, 2006, doi: 10.1002/spe.697.
- [5] J. Lonchamp, "A structured conceptual and terminological framework for software process engineering," *Proc. 2nd Int. Conf. Softw. Process Contin. Softw. Process Improv. SPCON 1993*, pp. 41–53, 1993, doi: 10.1109/SPCON.1993.236823.
- [6] C. F. Salviano, M. Jino, and M. D. J. Mendes, "Towards an ISO/IEC 15504-Based Process Capability Profile Methodology for Process Improvement (PRO2PI)," *SPICE 2004 Conf.*, pp. 77–84, 2004.
- [7] Software Engineering Institute, "CMMI for Development, Version 1.3," *Softw. Eng. Process Manag. Progr.*, no. November, pp. 1–520, 2010.
- [8] I. A. and O. N. L. Mathiassen, "Managing Change in Software Process Improvement," *IEEE Software*, pp. 84–91, 2005. doi: 10.1109/MS.2005.159.
- [9] V. R. Basili et al., "Linking Software Development and Business Strategy Through Measurement," *Computer (Long. Beach. Calif.)*, no. April, pp. 57–65, 2010.
- [10] A. S. Christof Ebert, Manfred Bundschuh, Reiner Dumke, *Best Practices in Software Measurement: How to use metrics to improve project and process performance*. Springer Berlin, Heidelberg, 2007.
- [11] F. García et al., "Towards a consistent terminology for software measurement," *Inf. Softw. Technol.*, vol. 48, no. 8, pp. 631–644, 2006, doi: 10.1016/j.infsof.2005.07.001.
- [12] W. E. Deming, *Out of the Crisis*. The MIT Press, 1986.
- [13] T. Gorschek and C. Wohlin, "Packaging software process improvement issues: A method and a case study," *Softw. - Pract. Exp.*, vol. 34, no. 14, pp. 1311–1344, 2004, doi: 10.1002/spe.615.
- [14] F. Pettersson, M. Ivarsson, T. Gorschek, and P. Öhman, "A practitioner's guide to light weight software process assessment and improvement planning," *J. Syst. Softw.*, vol. 81, no. 6, pp. 972–995, 2008, doi: 10.1016/j.jss.2007.08.032.
- [15] A. Fuggetta, "Software process: A roadmap," *Proc. Conf. Futur. Softw. Eng. ICSE 2000*, no. August, pp. 25–34, 2000, doi: 10.1145/336512.336521.



T.C. SANAYİ VE  
TEKNOLOJİ BAKANLIĞI

#MİLLİ  
TEKNOLOJİ  
HAMLESİ



İşçi Blokları Mahallesi Muhsin Yazıcıoğlu Caddesi No:51/C 06530 Çankaya/ANKARA

+90 (312) 289 92 22 - yte.bilgi@tubitak.gov.tr

TÜBİTAK - BİLGEM Yazılım Teknolojileri Araştırma Enstitüsü (YTE)