

# Yazılım Ürün ve Süreç Ölçümlerinin Otomasyonu

**Berk AKAR<sup>1,2</sup> Kıvanç DİNÇER<sup>1</sup> Ahmet TÜMAY<sup>1</sup>**

<sup>1</sup>TÜBİTAK – UEKAE G222 Tasarım ve Ürün Geliştirme Bölümü Atatürk Bulvarı, No:221  
Kat:1 5061 00 Kavaklıdere, ANKARA

<sup>2</sup>O.D.T.Ü. Bilgisayar Mühendisliği Bölümü Nönu Bulvarı, 06531 ANKARA  
{berk.akar, kivanc.dincer, ahmet.tumay}@tubitak.gov.tr

## Özet

Yazılım yaşam döngüsü süresince uygulanan süreçlerin takibinin yapılabilmesi, iyileştirilebilmesi ve ürünün istenilen kalite seviyesinde olduğunun tespiti amacıyla çeşitli metriklerin toplanması gerekmektedir. Ölçüm ve analiz sürecinin gerektirdiği ürün ve süreç metriklerinin manuel yöntemlerle toplanması iş gücü kaybına neden olmakta ve proje bitim tarihi yaklaştıkça metriklerin toplanmasına yönelik çalışmalar çoğu zaman ertelenmektedir. Anlatılan problemlerin çözümüne yardımcı olmak için kurumsal bir otomatik metrik toplama altyapısı oluşturulmalıdır. Bu metrik toplama altyapısının temel dayanağını ölçüm ve analiz süreci, bilgi gereksinimleri ve ölçümyapıları, çeşitli proje takibi ve CASE araçları ile bunların ürettiği verileri merkezi bir veritabanında toplayan yardımcı yazılımlar oluşturmaktadır. Bu bildiride yazılım ürün ve süreç ölçümlerinin otomatik olarak yapılabilmesine olanak tanıyan kurumsal bir altyapı önerilmekte ve bu amaçla geliştirilen prototip sistem tanıtılmaktadır.

## Anahtar Kelimeler

CMMI, ISO, Yazılım Süreç Ölçümü, Yazılım Ürün Ölçümü.

## 1. Giriş

Ürün ve süreç metriklerinin toplanması, hem ürün kalitesinin hem de süreçlerin etkinliğinin ölçülebilmesi amacıyla yürütülmesi gereken bir faaliyettir. “Her bir görev için harcanan gerçek işgücü ve zamanı proje planıyla karşılaştırmadığımızda proje durumunu izleyemezsiniz. Takımımızın hataları bulma ve düzeltme oranlarını izleyemediğinizde ürününüzün yeterince güvenilir olduğunu bilemezsiniz. Metrikler yazılım projelerinin daha iyi kontrol altında olmasını sağlamaktadır” [1].

Yazılım proje planlarının içerisine ölçüm ve analiz faaliyetlerinin de eklenmesi, yazılım proje yönetimi tarafından bu faaliyetlerin karar alma konusunda zorunlu bir ihtiyaç olduğunun anlaşılmasından sonra hız kazanmıştır. Projenin izlenebilirliğinin artırılması, ürünün kalitesinin ve sürecin nasıl işlediğinin yazılım yaşam döngüsü boyunca kontrol altında tutulması anlamında kurumsal ölçümlerin oynadığı rol giderek artmaktadır. Proje yönetiminin, özellikle proje bitim tarihi yaklaştıkça doğru karar alması için somut verilere ihtiyacı olmaktadır. Proje yönetiminin bu konudaki en büyük yardımcısı ise yine proje boyunca toplanan metrikler olmaktadır.

Metriklerin toplanabilmesi için sistematik bir yaklaşım gerekmektedir. Bu sistematik yaklaşımın kaynağı ve güvencesi kuruma ait ölçüm ve analiz sürecinin oluşturulması ve bu sürecin kurum içerisinde yaygınlaştırılmasıdır. Kurumlar kendi süreçlerini oluştururken genellikle örnek bir süreç modelini (ISO, AQAP, CMMI vb.) baz alan yapıları kullanmaktadırlar. Bu süreç modelleri içerisinde ölçüm ve analiz faaliyetlerine olan vurgu son yıllarda artmıştır. Örneğin CMMI kurumsal ölçüm ve bu ölçümün değerlendirilmesinin önemini kendi içerisinde başlı başına bir Ölçüm ve Analiz (Measurement and Analysis) süreç alanı oluşturarak vurgulamıştır [2,3].

Metriklerin toplanma biçimi de giderek değişmektedir. Metriklerin manuel olarak toplanması iş gücünün bu doğrultuda harcanmasına ve insan kaynaklı ölçüm hatalarına yol açmaktadır. Hatta proje bitim tarihinin yaklaşması, proje ekibi üzerindeki baskıyı artırdığından kimi zaman metriklerin toplanması ileri bir tarihe ertelenmektedir. Bahsedilen sebeplerden dolayı metriklerin olabildiğince otomatik olarak toplanabilmesi istenmektedir.

Bir çok kurum kendi bünyesinde çeşitli proje takibi ve CASE araçlarından istifade etmektedir. Bu araçlar ile süreç ve ürün takibi belli bir noktaya kadar yapılabilir. Ancak bazen bir metriğin ölçülebilmesi için çeşitli araçlardan toplanacak olan verilerin bir araya getirilmesi gerekmektedir. Bu araçlar genellikle farklı amaçlar doğrultusunda üretildiklerinden bir metriğin ölçülebilmesi zaman ve iş gücü kaybına yol açmaktadır. Ayrıca ölçüm ve analizlerin merkezi bir veritabanında toplanamamasından dolayı projelerin tarihsel izlenebilirliği, kontrolü ve proje yönetiminin zamanında müdahalesi mümkün olamamaktadır.

Bu bildiriye yukarıda bahsedilen problemlere çözüm olarak, yazılım süreç ve ürün ölçümlerinin otomatik olarak yapılabilmesine olanak tanıyan bir kurumsal altyapı önerilmektedir. Bu altyapının olabildiğince insan faktöründen bağımsız olması hedeflenmiştir. Altyapı dört ana unsurdan oluşmaktadır:

- Ölçüm ve analiz süreci; bu süreç ölçüm işlemlerinin sistematik hale getirilmesinden sorumludur.
- Bilgi ihtiyaçları ve ölçüm yapıları; bilgi ihtiyacı doğrultusunda yapılacak olan ölçüm faaliyetlerini tanımlamaktadır.
- Proje Takibi ve CASE araçları; hali hazırda kullanılan bir çok proje takibi ve CASE aracı çeşitli metrikleri üretebilmekte veya bu metriklere doğrudan veya dolaylı yoldan veri sağlayabilmektedir.
- Merkezi Metrik Veritabanı ve Yardımcı Yazılımlar; yardımcı yazılımlar verilerin kurum bünyesinde kullanılan çeşitli proje takibi ve CASE araçlarından toplanmasını, bunların işlenerek merkezi metrik veritabanına aktarılmasını ve uygun bir grafiksel kullanıcı arayüzü ile proje paydaşlarının bilgi ihtiyacına göre işlenen sonuçların gösterilmesini sağlamaktadır.

## 2. Ölçüm ve Analiz Süreci

Önerilen ölçüm ve analiz süreci, McGarry et. al. tarafından geliştirilen Practical Software Measurement (PSM) yaklaşımı baz alınarak oluşturulmuştur. PSM başarılı bir ölçüm programı için dört ana faaliyet ortaya koymaktadır [4]. Bunlar sırasıyla “Ölçüm planlaması”, “Ölçümün uygulanması”, “Ölçüm sürecinin değerlendirilmesi”, “Ortak anlayışın oluşturulması”dır. Bu dört faaliyet sırayla aşağıda açıklanacaktır.

**Ölçüm planlaması:** Bu faaliyet esnasında hangi ölçümlerin yapılacağı, projenin ve organizasyonun bilgi gereksinimleri doğrultusunda belirlenir. Karar vericilerin bilgi gereksinimlerinin belirlenmesi,

belirlenen bilgi gereksinimlerinin ölçüm yapılacak olan süreçlerle ilişkilendirilmesi yine bu faaliyet kapsamında icra edilmektedir [5].

Ölçüm ve analiz süreci, ölçüm planlamasının yapılabilmesi için proje yöneticisi tarafından ölçüm sürecinden sorumlu bir kişinin atanmasıyla başlar. Bu kişi organizasyonda yapılacak olan tüm ölçüm ve analiz faaliyetlerinin planlanmasından ve koordinasyonundan sorumlu olmalıdır. Planlanan ölçüm ve analiz faaliyetlerinin proje planında ne şekilde yer alacağı konusunda proje yöneticisi ile yakın işbirliği içerisinde olmalıdır.

Ölçüm ve analiz sürecinden sorumlu kişi, kurumun veya projenin bilgi gereksinimlerini belirlerken proje paydaşlarının görüşlerine başvurmalıdır. Bu kişi, istenilen bilginin önceliğine bakmanın yanı sıra “kolay veri toplanması”, “uygun araçların envanterde bulunması”, “metriğin kolayca yorumlanabilmesi” gibi kriterlere bakarak bilgi gereksinimine onay verir.

Ölçüm ve analiz sürecinden sorumlu kişi alınacak ölçümleri belirledikten sonra bu ölçümlerle proje paydaşları arasındaki ilişkiyi kurar. Proje paydaşlarının ortaya konan sürece etkin katılımı ve ölçüm ve analiz sürecinin gerekliliğinin ortaya konması için çeşitli bilgilendirme toplantıları düzenlenir.

Ölçüm ve analiz sürecinden sorumlu kişi veri toplama, saklama, raporlama faaliyetlerini her bir metrik için ayrı ayrı belirlemelidir. Bunlara ek olarak ölçüm yapıları içerisinde yer alan sınır değerleri de gözden geçirmeli ve gerektiğinde projenin kalite ve süreç ihtiyaçlarına göre ölçüm yapıları içerisinde yer alan sınır değerleri güncellemelidir.

**Ölçümün Uygulanması:** Ölçüm verilerinin toplanması, toplanan metriklerin analizinin yapılması ve sonuçların gösterilmesi faaliyetleri bu kapsam altında icra edilmektedir. Analizler; tahmin, planların fizibilitesi, gerçek verilerin planlanan verilere göre performansı gibi unsurları içerebilmektedir. Örneğin performans analizi, açılan ve kapatılan hata raporlarının zaman içerisinde gösterdiği eğilimin belirlenmesini içerebilmektedir [5].

Ölçüm verilerinin otomatik metrik toplama yazılımları vasıtasıyla merkezi bir veritabanında toplanması benimsenmelidir. Yine analizlerin ve ilgili raporların otomatik metrik toplama yazılımı içerisinde ihtiyaç sahibi proje paydaşının bilgi gereksinimine göre (takım lideri, proje yöneticisi vs.) görülebilmesi sağlanmalıdır.

**Ölçüm Sürecinin Değerlendirilmesi:** Ölçüm süreci ve toplanan metrikler periyodik olarak değerlendirilmeli ve gerekiyorsa geliştirilmelidir. Örneğin açılan ve kapatılan hata raporlarının zamana göre eğilimi, yazılımın kullanıcı kabul testlerine hazır olup olmadığı konusunda yeterli bilgi veremiyorsa, başka metrikler kullanılabilir. Örneğin yüksek öncelikli hata raporlarının tamamının kapatılmış olup olmadığı gibi [5].

Ölçümlerle ilgili geri beslemeler için iletişim kanallarının önceden tanımlı olması gerekmektedir. Bir proje paydaşı analiz raporlarının yeterince tatmin edici olmadığını veya verilerin yorumlanması hakkında yanıltıcı bir durumun söz konusu olduğunu düşünüyorsa, geliştirme önerisini ölçüm sürecinden sorumlu kişiye iletebilmelidir. Bu kişi iletilen öneriyi değerlendirmek ve ölçümlerde gerekli değişiklikleri yapmakla yükümlüdür.

**Ortak Anlayışın Oluşturulması :** Ölçüm altyapısının oluşturulması için gerekli kaynakların organize edilmesi, eğitimlerin alınması ve gerekli olan araçların satın alınması gibi faaliyetler bu

kapsam da icra edilmektedir. Ancak bu safhanın esas vurgusu ölçüm verilerinin kullanılması konusunda yönetsel kararlılığın netleştirilmesidir. Kurum içerisinde ölçümler hakkında ortak anlayışın oluşturulabilmesi için proje yönetiminin de katıldığı kurum içi yaygınlaştırma toplantıları yapılmalıdır. Hata eğilimi örneğine geri dönecek olursak; toplanan verilerin, yazılımın kabul testlerinin yapılması konusunda karar verme yönünden herhangi bir etkisi olmuyorsa, bu veriyi toplama ihtiyacının gerçekte var olduğuna inanmak zordur [5].

### 3. Bilgi İhtiyaçları ve Ölçüm Yapıları

Toplanacak olan metrikler için paydaşların bilgi ihtiyaçları belirlendikten sonra ölçüm yapıları (*Measurement Constructs*) oluşturulmalıdır. Bunun için özel oluşturulmuş bir organizasyonel şablon (*Measurement Construct Template*) kullanılabilir [5].

Ölçüm yapıları tanımlanırken içerisinde ölçümün amacı, neyin ölçüleceği, organizasyon mu yoksa proje bazında mı ölçüm yapılacağı, ölçüm metodları, birimi vs. bulunmalıdır.

Ölçüm yapılarından “değişiklik isteklerinin durumu”, “hata yoğunluğu”, “planlanan hata maliyetinin gerçek hata maliyetine oranı” aşağıda örnek olarak verilmiştir.

#### 3.1. Değişiklik İsteklerinin Durumu

Bu ölçüm yapısının amacı değişiklik isteklerinin durumunu zaman içerisinde izleyebilmektir. Altı adet ölçüm metodu tanımlanmıştır.

1. Kaynağı hata olan değişiklik isteklerinin toplamı.
2. Kaynağı öneri olan değişiklik isteklerinin toplamı.
3. Kaynağı doküman isteği olan değişiklik isteklerinin toplamı.
4. Kaynağı hata olan kapatılmış değişiklik isteklerinin toplamı.
5. Kaynağı öneri olan kapatılmış değişiklik isteklerinin toplamı.
6. Kaynağı doküman isteği olan kapatılmış değişiklik isteklerinin toplamı.

Yukarıda sıralanan ölçümler periyodik olarak toplanarak değişiklik isteklerinin zaman içerisindeki durumları izlenebilmektedir. Örneğin üç ay boyunca aktif değişiklik isteklerinde bir artış varsa düzeltici faaliyet açılması öngörülebilir.

#### 3.2. Hata Yoğunluğu

Bu ölçüm yapısının amacı satır sayısı başına düşen hatayı zamana bağlı olarak izleyebilmektir. Ölçüm için iki adet parametre tanımlanmıştır:

1. Hata sayısı (HS)
2. Satır sayısı (bin olarak) (KLOC)

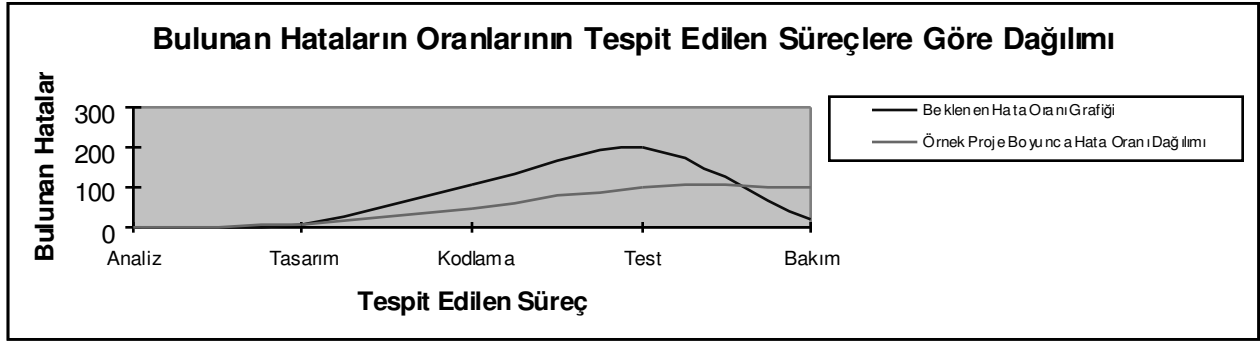
Hata yoğunluğuna (HY) denilecek olursa;

$$HY = HS / KLOC \quad (1)$$

formülünden hesaplanabilir.

Örneğin  $HY > 5$  olduğunda düzeltici faaliyet açılması şart koşulabilmektedir. Hataların hangi süreçten kaynaklandığının bulunması ise kurumlar için iyi bir pratiktir. Bu yöntemle zamana bağlı olarak hata yoğunluğu ölçülürken hangi süreçten (analiz, tasarım, kodlama, test, bakım)

kaynaklandığının bulunması o sürece yönelik olarak düzeltici faaliyetin açılmasını gündeme getirebilir. Aşağıdaki şekilde mavi renk ile çizilen grafik, projeler boyunca beklenen hata tespit oranı eğrisini göstermektedir. Pembe renk ile çizilen grafiğin örnek bir projeye ait olduğunu düşünürsek, bu grafik test sürecinde problemlerin varlığını işaret etmekte ve test süreci esnasında yakalanması gereken bir çok hatanın yazılım sahaya indikten sonra yakalandığını bize anlatmaktadır (Şekil 1).



Şekil 1. Bulunan hataların süreçlere göre dağılım grafikleri.

### 3.3. Planlanan Hata Maliyetinin Gerçek Hata Maliyetine Oranı

Bu ölçüm yapısının amacı toplam hata maliyetinin, kestirimi önceden yapılmış hata maliyetine oranına bakarak ileride yapılacak hata maliyeti kestirimleri için veri sağlamaktır. Ölçüm için iki adet parametre tanımlanmıştır:

1. Ölçüm yapılan periyoda kadar planlanan hata maliyeti (A)
2. Ölçüm yapılan periyoda kadar gerçekleşen hata maliyeti (B)

Toplam iş gücü varyansına (V) denilecek olursa;

$$V = 100 * (A - B) / A \quad (2)$$

formülünden hesaplanabilir. Varyans +/-%10'u geçerse düzeltici faaliyet açılabilmektedir.

### 4. Proje Takibi ve Case Araçları

Çeşitli firmaların bünyesinde kullanılan proje planlama araçları, iş gücü planlarının yapılması ve proje takibi amacıyla kullanılmaktadır. Özellikle bir çok metrikte kullanılan tahmin edilen iş gücü verisi proje planlarından alınmaktadır.

Projelerde harcanan gerçek iş gücünün bulunması amacıyla iş gücü takibi yazılımları (*worksheets program*s) kullanılmaktadır. Projelerde çalışanlar iş gücü takip yazılımı içerisinde bulunan projeler altında yer alan iş kalemlerini saat bazında doldurmakla yükümlüdürler.

Yazılımla ilgili değişiklik isteklerinin takibi için çok çeşitli araçlar kullanılmaktadır. Bu araçların çeşitli filtre ve sorgu özellikleri kullanılarak değişiklik isteklerinin durumları öğrenilebilmektedir.

Ürün metriklerinin toplanabilmesi için de çok çeşitli yardımcı yazılımlar kullanılabilir. Örneğin bazı tasarım araçları ürün metriklerinin toplanabilmesi için kendi içerisinde çeşitli fonksiyonlar sunabilmektedir.

## 5. Otomatik Metrik Toplama Yazılımı Prototipi ve Merkezi Veritabanı

Ölçüm ve analiz sürecinin otomatizasyonunun ne kadar sağlanabileceğinin anlaşılabilmesi ve pratikteki yararlarının değerlendirilebilmesi için bir prototip Otomatik Metrik Toplama Yazılımı (OMTY) yazılmıştır. Bu prototip ile çeşitli proje takibi ve CASE araçlarından metriklerin toplanması, toplanan metriklerin merkezi bir veritabanı üzerine kaydedilmesi ve raporlanabilmesi işlemlerini gerçekleştirebilmektedir.

### 5.1. OMTY Tasarım Yaklaşımı

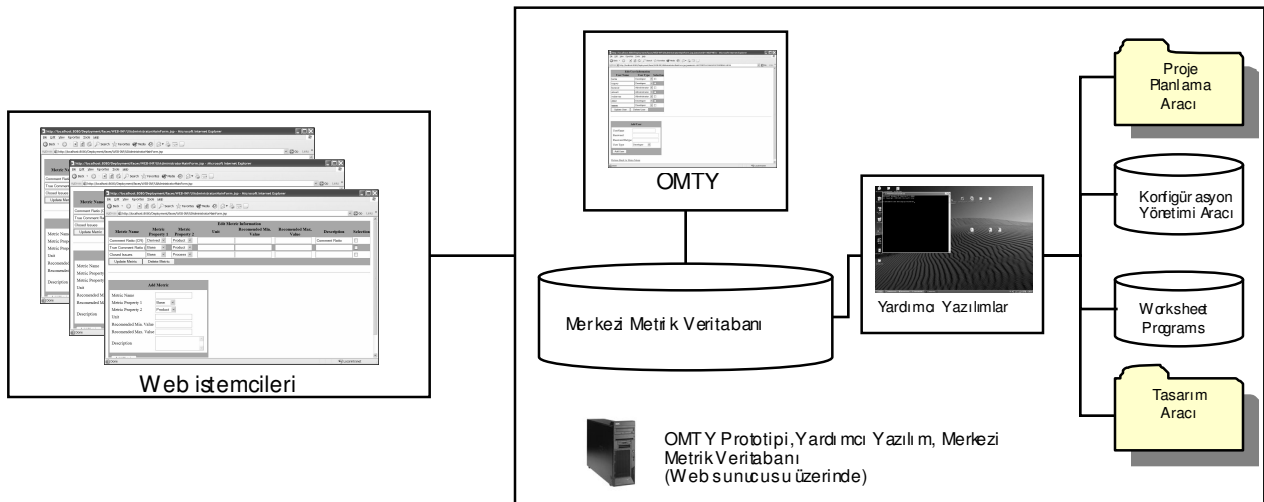
Literatürde yer alan bir çok yaklaşım mümkün olduğunca jenerik nesne modelleri kullanarak oluşturulan ölçüm altyapılarını önermektedir. Bu nesne modelleri mümkün olan tüm olasılıklar göz önünde bulundurularak oluşturulmaya çalışılmıştır [6,7].

OMTY'nin en önemli özelliği mümkün olduğunca jenerik nesne modellerden kaçınarak basit ve anlaşılması kolay bir yapı sunmasıdır. Bu yapı sayesinde OMTY'nin devreye alınma süreci çok hızlı olmuştur. Buna ek olarak jenerik nesne modellerini kullanmanın getirdiği dezavantajlardan biri olan geliştirecilerle olan bağımlılık son derece azalmıştır.

### 5.2. Mimari Tasarım

OMTY proje paydaşlarının bilgi ihtiyaçlarını en etkin bir şekilde karşılamak amacıyla web sunucusu üzerinde çalışacak şekilde yapılandırılmıştır. Proje paydaşları bilgisayarlarında yüklü bulunan tarayıcılar (web istemcileri) yardımıyla OMTY'ye erişebilmektedirler.

Web sunucusu ile aynı makinada yer alan yardımcı yazılımlar ise OMTY'nin oluşturduğu ölçüm bilgilerinden yararlanarak, proje takibi ve CASE araçlarından ölçüm verilerini toplar ve merkezi metrik veritabanına aktarır. Web sunucusunun veya OMTY'nin herhangi bir nedenden ötürü çalışmaması durumunda bile yardımcı yazılımlar görevine devam edebilmektedir. Şekil 2'de yazılıma ilişkin mimari tasarım görülmektedir.



Şekil 2. OMTY'nin mimari tasarımı.

### 5.3. Ölçümlerin Zam anlaması ve Metriklerin Araçlardan Toplanması

OMTY yöneticisi projelere ait metriklerin tanımlamalarını yaptıktan sonra ilgili metriğin ölçümüne ne zaman başlanacağını ve hangi periyotlarla bunun gerçekleştirileceğini OMTY tarafından sağlanan arayüzde gerekli konfigürasyonu yaparak tanımlar.

Ölçümlerin merkezi metrik veritabanına toplama işlemi ise yardımcı yazılımlar üstlenmektedir. Bu yazılımların OMTY'den bağımsız çalışabilecek şekilde tasarlanmasının sebebi, herhangi bir şekilde web sunucusunun görev yapamaması durumunda her zaman ayakta kalarak ölçüm görevine devam edebilmeleridir. Sürekli olarak görevlerini yapabilmeleri amacıyla işletim sistemlerinin başlangıçta çalıştırılacak yazılımlar listesine yerleştirilmektedirler. Yardımcı yazılımlar, OMTY yönetici tarafından belirlenen ölçüm konfigürasyonunu veritabanından okur ve her bir ölçümün kullandığı sınıfı çağırır (Şekil 3).

Measurement Name	Begin Date (dd/MM/yyyy)	Period	Status	Class Name	Related Project Selection
Issues Measurement	07/03/2005	Weekly	Start	connections.issueTask	Software Project <input type="checkbox"/>
X Project Product Measurement	30/05/2005	Monthly	Start	connections.productTask	Software Project <input type="checkbox"/>

Update Measurement Delete Measurement

**Add Measurement**

Measurement Name

Begin Date Time (dd/MM/yyyy)

Period

Class Name

Status

Add Measurement

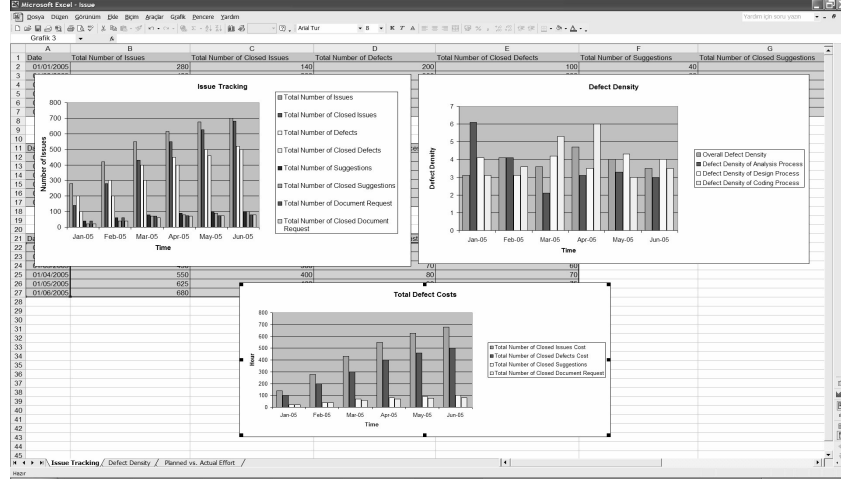
Return Back to Main Menu

Şekil 3. OMTY ölçüm konfigürasyon ekranı.

Yardımcı yazılımlar metrikleri proje planlama aracından, konfigürasyon yönetimi aracından, worksheet programlarından ve tasarım aracından toplamaktadır. “Örnek Metrikler ve Ölçüm Yapıları” bölümünde anlatılan ölçüm yapılarından değişiklik isteklerinin durumu konfigürasyon yönetimi aracından; hata yoğunluğu konfigürasyon yönetimi aracı ve tasarım aracından (satır sayısı); planlanan hata maliyetinin gerçek hata maliyetine oranı ise worksheet programlarından ve proje planlama aracından alınan verilerle hesaplanmaktadır.

### 5.4. Analizlerin ve Raporların Görüntülenmesi

Analiz sonuçlarının ve raporların ilgili kullanıcı tarafından görüntülenebilmesinin sağlanabilmesi için OMTY tarafından sağlanan fonksiyonlular kullanılmaktadır. Her bir ölçümün hangi kullanıcılar tarafından görülebileceği OMTY yöneticisi tarafından belirlenebilmektedir. Böylece herkesin kendi ilgilendiği ölçüme ulaşması kolaylaştırılmış, bunun dışında kalan ölçümlere erişim engellenmiştir. Ölçüm sonuçları rapor olarak MS Excel'e aktarılabilmektedir. Aşağıdaki şekilde ölçüm raporu örnekleri görülmektedir (Şekil 4). Karar vericilerin (proje yönetimi vs.) sadece bir ölçüm raporuna bakarak karar vermesi yerine çeşitli alternatif bakış açıları kazanılarak daha sağlıklı karar alınması sağlanmıştır. Örneğin yeni alınacak projenin maliyet planları yapılırken geçmiş projeler boyunca toplanan hata maliyetlerinin yeni plana doğrudan eklenmesi yerine, hata yoğunluğunun süreç bazında değişimi göz önüne alıp gerekli iyileştirmeleri planlayarak, yeni maliyet planı oluşturulabilir.



Şekil 4. Ölçümlere ait rapor örnekleri.

## 6. Sonuç

Metriklerin ve ölçümlerin giderek artan önemi karşısında olgunluk seviyesi yüksek yazılım organizasyonları kendi metrik toplama altyapılarını oluşturmakta ve bunları mümkün olduğunca insan faktöründen bağımsız hale getirmeye çalışmaktadır. Altyapının temel ayağını ölçüm ve analiz süreci oluşturmaktadır. Sürecin yaşama geçirilmesinin ardından ilgili metriklerin toplanabilmesi için proje paydaşlarının bilgi gereksinimleri ortaya konmalıdır. Bilgi gereksinimleri çeşitli ölçüm yapılarına dönüştürülerek ölçüm ve metrik özelliklerine ait tüm bilgilerin ortak bir yapıda saklanabilmesi sağlanmalıdır.

Kurum içerisinde yer alan mevcut proje takibi ve CASE araçları metrik toplama konusundaki en büyük yardımcılarıdır. Bu araçlardan elde edilecek verilerin merkezi bir veritabanında toplanabilmesini, işlenebilmesini ve raporlanabilmesini sağlayan yardımcı yazılımlara ihtiyaç duyulmaktadır. Prototip olarak hazırlanan OMTY, metrikleri farklı araçlardan çekerek merkezi metrik veritabanında saklamakta ve bunları MS Excel ortamında raporlayabilmektedir. Ölçüm verilerinin mümkün olduğunca otomatik ve insan faktöründen bağımsız olarak toplanmasının sonucunda ölçümler, fazladan bir iş gücü gerektirmeden karar vericilerin önüne gelmektedir.

## 7. Referanslar

- [1] Wiegers K.E., A Software Metrics Primer, [http://www.processimpact.com/articles/metrics\\_pri mer.pdf](http://www.processimpact.com/articles/metrics_pri mer.pdf), 1999.
- [2] Goldenson D. R., Jarzombek J., Rout T., Measurement and Analysis in Capability Maturity Model Integration Models and Software Process Improvement U.S. Army, Cross Talk: The Journal of Defense Software Engineering <http://www.stschill.af.mil/crosstalk/2003/07/goldenson.html>, Temmuz 2003.
- [3] CMMI Product Team. Capability Maturity Model Integration (CMMI®) Version 1.1 Continuous Representation. Pittsburgh, PA: Carnegie Mellon University, Mar. 2002.
- [4] Jones C., Making Measurement Work, U.S. Army, Cross Talk: The Journal of Defense Software Engineering <http://www.stschill.af.mil/crosstalk/2003/01/jones.html>, Ocak 2003.
- [5] McGarry J., Card D., Jones C., Layman B., Clark E., Dean J., Hall F., Practical Software Measurement, Objective Information for Decision Makers, 2001.
- [6] Palza E., Fuhrman C., Abran A. Establishing a Generic and Multidimensional Measurement Repository in CMMI Context, <http://doi.ieeecomputersociety.org/10.1109/SEW.2003.1270721>, Temmuz 2005.
- [7] Walkerdon F., A Design for a Software Metrics Repository, CAESAR TECHNICAL REPORT #96/07 <http://www.caesar.unsw.edu.au/publications/ctr96-7.zip>, Temmuz 2005.