

# Nesne Yönelimli Yazılım Tasarımının Testedilebilirliğini Ölçmeye Yönelik Bir Model

## A New Model To Measure Testability in Object Oriented Software Design

**Tolga YURGA**

METU Informatics Institute Information Systems Ph.D. Student

e-posta: tyurga@tr.net

### ÖZET

Bu araştırmanın amacı, nesne yönelimli yazılım tasarımının testedilebilirliğini, belirli tasarım parametreleri cinsinden ifade etmeye yönelik bir yaklaşım geliştirmektir. Mevcut proje verilerini kullanarak, testedilebilirlik ve tasarım parametreleri arasındaki korelasyon çıkarılmaktadır. Çoklu Doğrusal Regresyon kullanılarak da, en çok ilintili tasarım parametreleri ile testedilebilirlik endeksi arasında bir model oluşturulmaktadır. Geliştirilen model ve yaklaşımın, organizasyon için anahtar yorumlama ve değerlendirme yol göstericisi olarak, yazılım geliştirme adımlarını, daha testedilebilir hale getirmek üzere geliştirmesi hedeflenmektedir.

### ABSTRACT

This research work aims to identify a measure of testability of object-oriented software design in terms of certain design parameters. Using available project data, the correlation among the design parameters and testability is extracted. Multiple Linear Regression is then used to compose a model among the mostly correlated design parameters and testability index. Developed model is anticipated to improve development efforts through better testability that is considered as the key assessment aid for the organization.

## 1. Giriş

Bu yazıdaki çalışmada, yazılım tasarımı üzerine yoğunlaşıp, test etme işleminin kolaylaştırılması sayesinde, yazılım aşamasının nasıl geliştirilebileceği incelenecektir. Bir yazılım sisteminin ya da bileşeninin, test etme aşamasını kolaylaştırma derecesine, *testedilebilirlik* denilir. Test aşamasını dikkate alarak bir yazılım sistemini tasarlamaya ise *testedilebilirlik için tasarım* denilir. [10]

IEEE, testedilebilirliği, “Bir sistem ya da bileşenin, test kriterlerinin belirlenmesini ve bu kriterlerin karşılanma durumuna testlerin performansı sayesinde karar vermeyi kolaylaştırma derecesi” olarak tanımlamaktadır[6]. Diğer bir deyimle, bir yazılım sisteminin testedilebilirliği, sistemi test etmek için gerekli emeğin miktarını veya test ederek program içerisinde bir hata ortaya çıkarma olasılığını işaret etmektedir.

Projelerin testedilebilirliği, yazılım proje yönetiminin önemli bir niteliği olarak ortaya çıkmaktadır. Test aşaması, yazılımcılar ve yöneticiler için geliştirilen ürünü izlemek ve değerlendirmekte büyük

önem taşıyan bir faaliyettir. Endüstrinin bu konuya yapmış olduğu vurgu neticesinde, test edilebilirlik, önemli bir araştırma sahası olarak benimsenmiştir. Yazılım endüstrisi tarafından da önerildiği üzere, test edilebilirlik, kurumun daha dikkatli olması gereken alanları belirlemesi çabasında, diğer tasarım parametrelerine bağlantılı olacaktır[8].

Pratikte, test aşaması, ortalama bir yazılım geliştirme projesinde, önemli bir zaman ve emek tüketmektedir. Bir yazılımın modül veya bileşenlerini, en yeni teknolojilerle, en iyi tasarımcılarla tasarlamak, tasarımı ve sonuç olarak da tasarıma göre kodladıklarımızı test edemediğimiz sürece hiçbir anlam ifade etmeyecektir[7].

Bu makalenin amacı, nesne yönelimli tasarımın test edilebilirliğini ölçmeye ve doğrulamaya yarar yeni bir yaklaşım ve model ortaya koymaktır. Bu sayede, yazılım geliştirme yaşam döngüsünün sonraki aşamalarında, test edilemeyen tasarım nedeniyle ortaya çıkabilecek sorunların giderilmesi için gerekli işgücü ve maliyeti minimuma indirmektir. Modeli geliştirebilmek için, gerçek yaşamda kullanılmakta olan projelere ait ölçüt verileri kullanılıp, matematiksel teknikler yardımıyla bu veriler analiz edilip, model için gerekli bilgi ortaya çıkarılacaktır.

## 2. Modelin Oluşumu

### 2.1. Tasarım Parametreleri ve Ölçütleri

Bu araştırmada, bir yazılım projesinin tasarımının test edilebilirliğini ölçebilmek amacıyla, yazılım tasarımı ile dolaylı ya da doğrudan bağlantılı çok sayıda yazılım ölçütünden faydalanılmıştır. Bu aşamadaki amaç, modelin oluşumunda kullanılacak tasarım parametrelerini temsil eden ölçütleri belirlemektir. Modelin güvenilirliği ve doğruluğu açısından, parametre ve ölçüt alanımı mümkün olduğunca geniş tutmaya çalışılmıştır.

Konuyla ilgili yapılan literatür taraması sonuçları, eldeki proje verileri, yazılım endüstrisi ve akademik dünyada genel olarak kaydı tutulan ölçütlerin kapsamı neticesinde, aşağıda belirtilen tasarım parametreleri ve bunlara bağlı önemli ölçütlerin kullanımının model için uygun olduğu ortaya çıkmıştır[2,4,5,7,10]. Bu parametreler:

- Bağlaşım (Coupling)
- Yapışıklık (Cohesion)
- Kalıtım (Inheritance)
- Çok Şekillilik (Polymorphism)
- Sarma (Encapsulation)
- Boyut (Size)
- Karmaşıklık (Complexity)

#### 2.1.1. Bağlaşım (Coupling)

Nesne yönelimli bir tasarımda, bağlaşım, haberleşen modüller arasındaki ilişki ve bağımlılıkları göstermektedir. *Nesneler Arası Bağlaşım (NAB-CBO=Coupling Between Objects)*, bağlaşım özelliği için en belirleyici ölçüt olup, bir sınıfın bağlı olmuş olduğu diğer sınıfların sayısını belirtmektedir. Yüksek ölçüt değeri, sınıflar arasında yüksek derecede ilişki ve bağımlılık olduğunu göstermektedir[11].

### 2.1.2. Yapışıklık (Cohesion)

Yapışıklık, bir tasarım parçasının, tek, iyi belirlenmiş bir hedefi gerçekleştirmek için buldukları katkı derecesini göstermektedir. Sınıf seviyesinde yüksek yapışıklık değeri, sınıfın bütün elemanlarının kuvvetli olarak ilintili olduğunu göstermektedir.

*Metotlardaki Yapışıklık Eksikliği (MYE – LCOM=Lack Of Cohesion in Methods)*, yapışıklık özelliği için en belirleyici ölçüt olup, bir sınıf içerisinde ortak değişken kullanan metot çiftleri sayısı ile ortak değişken kullanmayan metot çiftleri sayısı arasındaki farkı göstermektedir. Düşük ölçüt değeri, sınıfın yapışık olduğunu gösterirken, yüksek değerler ise sınıfın, daha yüksek yapışıklık değerlerine sahip ayrı sınıflara bölünebileceğini göstermektedir[13].

### 2.1.3. Yapısal Karmaşıklık (Structural Complexity)

Bir sınıfın yapısal karmaşasını ölçmek için, bütün metotlarının karmaşa değerlerinin toplanması gerekir. Bu da, *Sınıf Başına Ağırlıklı Metod (SBAM-WMC=Weighted Methods Per Class)* ölçütünün özünü oluşturmaktadır[17]:

$$SBAM = \sum_{i=1}^n k_i$$

İfade içerisindeki  $k_i$ , sınıf içerisindeki  $n$  tane metodun ayrı karmaşa değerini ifade etmektedir.

*Siklomatik Karmaşa (SK-CC=Cyclomatic Complexity)* ölçütü, bir modülün (sınıfın alt bileşeni, örn. Metot) karar yapısının karmaşasını ölçmeye yarar. Doğrusal olarak bağımsız karar patikalarının sayısını gösterir[16].

*Çocuk Sayısı (ÇS-NOC=Number Of Children)* ise, sınıf hiyerarşisi içerisinde bir sınıfın alt sınıflarının sayısını göstermektedir. Kaç alt sınıfın, ebeveyn sınıfın metodlarını kalıtımsal olarak içereceğinin bir ölçüsüdür.

### 2.1.4. Sarma (Encapsulation)

Sarma, bir sınıfın dış arayüzü ve dahili implementasyonu arasında temiz bir ayırım olmasına çalışan bir tür soyutlama olarak tanımlanabilir.

*Yüzdesel Genel Veri Oranı (YGVO-PUB=Public Percent Data)* ölçütü, bir sınıf içerisindeki genel ve korunumlu veri yüzdesini gösterir. Genel olarak, düşük değerler yüksek sarmayı işaret eder[12].

### 2.1.5. Çok Şekillilik (Polymorphism)

Çok şekillilik, tam yapısını bilmeden, ayrı sınıfların nesnelarini, sadece ortak özelliklerinin bilgisini kullanarak etkilemek, değiştirmek ve kullanmak olarak tarif edilebilir.

*Sınıfa Yönelik Karşılık (SYC-RFC=Response For Class)* ölçütü ise, bir sınıf içerisinde bulunan metot sayısı ile kalıtım sayesinde, bir nesne tarafından ulaşılabilir durumdaki metotların sayılarının toplamından ibarettir.

### 2.1.6. Kalıtım (Inheritance)

Bir sınıfın kalıtım hiyerarşisindeki derinliği, *Kalıtım Ağacı Derinliği (KAD-DIT=Depth In Tree)*, sınıfın bulunduğu noktanın, ağacın kökünden olan maksimum uzaklığı olup, ata sınıfların sayısı ile ölçülür.

Bir sınıfın hiyerarşide daha derinlerde olması, davranışlarını tahmin etmeyi daha karmaşık hale getirecek şekilde daha çok metodu kalıtımsal olarak yapısı içerisine katmasını ifade etmektedir.

### 2.1.7. Boyut (Size)

*Sınıf Başına Toplam Satır Sayısı (SBTSS-TLOCC=Total Lines Of Code per Class)* , sınıf seviyesinde boyutu ölçmek için kullanılabilir en basit, kolay hesaplanabilir bir boyut ölçüm olup, sınıf içindeki yorum ve çalıştırılabilir kod satırları dahil tüm satır sayısını belirtir.

## 2.2. Model Çalışması

Deneysel olarak modelin oluşturulabilmesi amacıyla, NASA Independent Verification & Validation (IV & V) Facility Ölçütü Data Program'ına ait veriler kullanılmıştır. Bu verilerin içerisinde bulunan, farklı dillerle yazılmış iki adet nesne yönelimli yazılım projesine ait ölçüt ve proje izleme ve ölçüm verileri, araştırma ihtiyaçlarıyla büyük oranda örtüşmüştür.

Proje veri ambarı, araştırmada kullanılmaya karar verilen ölçütlerin ve tasarım parametrelerinin önemli bir kısmını içermekteydi. Bu veri ambarı, NASA'nın kendi ihtiyaçları doğrultusunda hazırlamış olduğu çeşitli amaçlara hizmet eden yazılımlarına ait, problem verileri, ürün verileri ve ölçüt verilerini içeren, önemli bir kaynak durumundadır.[19]

Veri ambarında bulunan ölçütlerden yararlanarak, yeni bir test edilebilirlik endeksi oluşturulmaya çalışılmıştır. Özellikle de, projenin gerçek hayatta aktif olarak kullanılması sırasında ortaya çıkan hata kayıt kütüğüne ait veriler, test edilebilirlik kavramını ölçme konusunda çok yardımcı olmuştur. Bu üç ölçüt verisi şunlardır:

- 6 Ay İçindeki Hata Kayıtları (HK6-ER6)
- 12 Ay İçindeki Hata Kayıtları (HK12-ER12)
- 24 Ay İçindeki Hata Kayıtları (HK24-ER24)

Bu ölçütler, her bir tasarım nesnesinin, projenin gerçek hayatta kullanımı sırasında ortaya çıkan, kodlama, test ve entegrasyon fazları içerisinde farkedilmemiş ve kullanıcı kullanımı sırasında farkedilen hataları göstermektedirler. Eğer bir nesne hakkında projenin yaşam döngüsü sırasında herhangi bir hata kaydı yoksa, bu o nesnenin mükemmel derecede test edilebilir olduğunu gösterir. Hata olması ise, nesnenin yeteri kadar test edilebilir olmaması sonucunda gizli bazı hataların durmaya devam ettiğini ve de test aşamasında bu hataların ortaya çıkartılmadığını gösterir.

Çeşitli deneme-yanılma çalışmaları neticesinde, bu üç ölçüt verisi kullanılarak, aşağıdaki ifadede belirtilen katılım oranları ile beraber bir Ortalama Hata Kayıt Değeri oluşturmanın doğru olduğu gözlemlenmiştir. Bir sınıfa ait Ortalama Hata Kayıt Değeri :

$$SINIF\_ORTALAMA\_HATA = \frac{4 * HK24 + 2 * HK12 + HK6}{7}$$

Zamansal Periyottaki Hata Kayıtlarının başlarındaki çarpanlar, Ortalama Değer içerisindeki ağırlıklarını göstermektedir. Uzun süre zarfında ortaya çıkan hatalar, daha zor bulunur hatalar demek olduğundan, ağırlık olarak daha fazla değer ifade etmektedirler.

Ortalama Hata Oranını, test edilebilirlik ölçümüne çevirebilmek için, çeşitli deneme-yanılma çalışmasından sonra, aşağıdaki endeksin, en doğru sonuçları verdiği gözlemlenmiştir. Her bir sınıfın test edilebilirlik endeksi şöyle hesaplanır:

Test edilebilirlik Endeksi =  
{

$$100 * MUT \left( \frac{SINIF\_ORTALAMA\_HATA - 3 * STD\_SAP(SINIF\_ORTALAMA\_HATA)}{3 * STD\_SAP(SINIF\_ORTALAMA\_HATA)} \right),$$

$$SINIF\_ORTALAMA\_HATA \leq TUM\_ORTALAMA\_HATA + 3 * STD\_SAP(SINIF\_ORTALAMA\_HATA)$$

0, diğer koşullarda}

MUT mutlak değeri, TUM\_ORTALAMA\_HATA tüm sınıfların ortalama hata değerini, ve STD\_SAP de sınıf bazında ortalama hata değerlerinden oluşan örnekleme standart sapma değerini göstermektedir. Hesaplamaları ve varsayımlarıma göre, bütün sınıfların ortalama hata değerinden 3 standart sapmasından büyük hata değerlerinin test edilebilirliği 0 olarak kabul edilmekte, bu değerden büyük değerli hatalara sahip sınıfları test edilemez derecede bir yapıya sahip olduğunu ifade etmektedir. Hata oranı 0 olan sınıflar ise, mükemmel test edilebilir olarak kabul edilmekte, %100 test edilebilir denilmektedir.

### 2.3. Matematiksel Sonuçlar

	PUB	CBO	DIT	LCOM	NOC	RFC	WMC	CCOMP	TLOCC	ORT.HATA	Test edilebilirlik
YGVO - PUB	1,00										
NAB - CBO	0,05	1,00									
KAD - DIT	-0,07	0,47	1,00								
MYE - LCOM	0,27	0,26	0,26	1,00							
ÇS - NOC	0,13	-0,03	-0,03	-0,03	1,00						
SYC - RFC	0,06	0,39	0,65	0,33	-0,05	1,00					
SBAM - WMC	0,15	0,25	0,14	0,32	0,04	0,63	1,00				
SK - CCOMP	0,02	0,63	0,56	0,20	0,01	0,30	0,11	1,00			
SBTSS - TLOCC	0,04	0,65	0,52	0,10	-0,02	0,28	0,08	0,91	1,00		
ORT.HATA	0,05	0,34	0,07	0,07	-0,07	0,12	0,09	0,26	0,30	1,00	
Test edilebilirlik	0,00	-0,44	-0,05	-0,05	0,07	-0,14	-0,13	-0,28	-0,32	-0,95	1,00

**Figure 1:** Ölçütler Ve Test edilebilirlik Arasındaki Korelasyon Katsayıları

Yukarıdaki tabloda bulunan ölçütler ve test edilebilirlik arasındaki katsayıların incelenmesi sonucunda, Bağlaşım, Boyut ve Karmaşıklık parametrelerine ait ölçütlerin test edilebilirlik endeksi ile daha güçlü bir korelasyona sahip oldukları gözlemlenmiştir. Sadece YGVO ölçütünün 0

seviyesinde bir korelasyonu olması ve diğer ölçütlerin etkilerinin ihmal edilmemesi için, Matematiksel Doğrusal Regresyon Modeli içerisinde, YGVO harici bütün ölçütlerin kullanılmasını uygun görülmüştür.

Testedilebilirlik endeksini etkileyen bireysel tasarım parametrelerinin nispi önemleri, korelasyon ilişkisine bağlı olarak orantılı olarak bir ağırlığa sahip olur. Bu ağırlık katsayılarının hesaplanabilmesi amacıyla, Matematiksel Doğrusal Regresyon Modeli uygulanmıştır.

Böyle bir model, bağımlı değişkenler ve bağımsız değişkenler arasında, doğrusal bir ilişki kurmaya yarar. Regresyon denklemi, aşağıdaki şekilde olur:

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_m x_m$$

“x”ler bağımsız değişkenleri, “y” bağımlı değişkeni, “β”lar da regresyon katsayılarını belirtmektedir. Bu katsayılar, bağımsız değişkenler sabit tutulduğundan, bağımlı değişken üzerindeki ortalama bağımlı artış miktarını göstermektedirler[9].

Sonuçlardan da görüldüğü üzere, testedilebilirlik, birden çok tasarım parametresine bağlıdır. Bileşen bazındaki etki tahmin edilebilir ve bileşen ağırlığı, regresyon denklemi ile sabitlenebilir. Bu suretle, bireysel olarak tasarım parametrelerine ait bileşen ağırlıkları, regresyon katsayıları (β’lar) cinsinden hesaplanmış olur.

Resgresyon doğrusunu oluşturmak için, NASA veri ambarında bulunan [19], biri C++, diğeri ise JAVA ile yazılmış bulunan, yaklaşık 200 sınıf içeren iki yazılım projesine ait veriler kullanılmıştır. Aşağıda, testedilebilirlik endeksini, tasarım parametrelerine ilişkilendiren denklem bulunmaktadır:

***Testedilebilirlik Endeksi (Tasarım Ölçütleri Cinsinden) =***

$$92.59 - 1.69 * NAB (CBO) + 6.94 * KAD (DIT) - 0.13 * SYC (RFC) \\ + 0.03 * MYE (LCOM) - 1.95 * ÇS (NOC) + 0.07 * SBAM (WMC) \\ + 0.03 * SK (CCOMP) - 0.30 * SBTSS (TLOCC)$$

***Testedilebilirlik Endeksi (Tasarım Parametreleri Cinsinden) =***

$$93.79 - 1.68 * Bağlaşım (Coupling) + 6.37 * Kalıtım (Inheritance) \\ - 0.10 * Çok Şekillilik (Polymorphism) + 0.03 * Yapışıklık (Cohesion) \\ + 0.30 * Karmaşıklık (Complexity) - 0.32 * Boyut (Size)$$

Karmaşıklık katsayısı, üç farklı karmaşıklık ölçütü içerisinde en güçlü korelasyona Siklomatik Karmaşa ölçütü sahip olduğu için, sadece bu ölçütü içeren bir regresyon modeli ile yeniden hesaplanmıştır. Her bir tasarım parametresi, kendisini en güçlü ve yaygın şekilde ifade eden tasarım ölçütü tarafından temsil edilmektedir. Bu tablodaki katsayılar, doğrusal denklemdeki katsayıları, kesişim ise, doğrunun sabit değerini, yani y-ekseni ile kesiştiği noktayı göstermektedir.

%95 güven seviyesi ile hesaplanan regresyon modeline ait detaylar aşağıdadır:

	<i>Katsayılar</i>	<i>Standart Sapma</i>	<i>Alt %95</i>	<i>Üst %95</i>
Kesişim	92,59	4,96	82,79	102,39
NAB - CBO	-1,69	0,41	-2,50	-0,88
KAD - DIT	6,94	2,59	1,82	12,06
MYE - LCOM	0,03	0,06	-0,08	0,14
ÇS - NOC	1,95	2,67	-3,34	7,23
SYC - RFC	-0,13	0,10	-0,33	0,07
SBAM - WMC	0,07	0,16	-0,25	0,39
SK - CCOMP	0,03	2,79	-5,49	5,55
SBTSS - TLOCC	-0,30	0,32	-0,93	0,34

**Figure 2:** Regresyon Detayları

### 3. Sonuç

Modeli oluşturmakta kullanılan veriler, tek bir kaynağın verileri olduğu ve bu kaynağın iş akışları hakkında, şu an için, herhangi bir veri içermediğinden dolayı, modelin sadece bir kurum içerisinde farklı projelerin test edilebilirliğinin karşılaştırılmasına, şirket içerisinde bir veri ambarı oluşturulmasına yararı hedeflenmektedir.

Genele uygulanabilir bir model yaratmak için, çok ve farklı, birbirine bağımlı ya da bağımsız parametrelerin de işin içine katılması gerekir. Bu parametrelere ait detaylı ölçütlere sahip olunması, ya da kolayca bu ölçütlerin elde edilebilir olması gerekir. Firmanın iş akışının, varsa, uygulamakta olduğu kalite sürecinin, işgücüne ait bilgi birikiminin, ve daha birçok önemli etkenin bu araştırma altında incelenmesi gerekmektedir. Mevcut çalışma, zaman ve hedef konusunda daha farklı bir çizgide olduğundan, ve bu tür detay verilerin, şu an itibarıyla tümünün aynı anda elde edilemeyecek kadar zor olması nedeniyle, belirli bir alanda, belirli varsayım ve kısıtlamalar yapmak zorundadır.

Nesne Yönelimli Yazılım Tasarımının Test edilebilirliğine ölçmeye yönelik olarak önerilen model, çeşitli deneme-yanılma ve varsayımlardan hareket etmektedir. Yakın zaman içerisinde, bu varsayım ve denemeler, daha güvenilir ve matematiksel sonuçlara dayandırılacaktır. Ayrıca süreçlere ilgin veriler ve test edilebilirlik ile projenin yaşam döngüsü içerisindeki fazlarla etkileşimini ortaya çıkarmaya çalışacak veriler elde edilmeye çalışılacak, bu sayede, ürün bakış açısının yanına süreç bakış açısı da katılmış olacaktır.

Modelin son şeklini almasıyla beraber, gerçek yaşamda kullanılan aynı şirkete ait projeler üzerinde uygulanması ve doğrulanması hedeflenmektedir. Çalışma, daha farklı ilişkiler ortaya çıkarmak üzerine de yoğunlaşacağı için, bulunan ilişki ve korelasyonlar, çeşitli farklı boyutlara göre genişletilecek ve irdelenecektir.

Bütün bu çalışmalar ve planlar neticesinde ortaya çıkması hedeflenen model, kesin, yaygın bir test edilebilirlik endeksi olmaktan çok, yazılım firmalarına, tasarım konusunda yaşamış oldukları sıkıntıları azaltmada yardımcı olmaya çalışacak, belirli öğüt ve yönlendirici ipuçları ile projeye ait zaman, işgücü ve maliyet kalemlerini, daha düzgün ve yerinde kullanılmasını sağlamak amaçlamaktadır.

#### 4. Referanslar

- 1) A. Arifoğlu and A. Doğru, “Yazılım Mühendisliği”, SAS Bilişim Yayınları, 2001
- 2) L. Briand, J. Wüst and H. Lounis, “Investigating Quality Factors in Object-Oriented Designs: An Industrial Case Study”, ISERN-98-29 – Version 2, 1998
- 3) B. Baudry, “Towards a 'Safe' Use of Design Patterns to Improve OO Software Testability”, Beaulieu Üniversitesi, 2001
- 4) J.M. Voas, “Factors that Affect Software Testability”, IS Bölümü, NASA LRC, 1998
- 5) R. Marinescu, “Measurement and Quality in Object-Oriented Design”, Timisoara Politeknik Üniversitesi, 2002
- 6) IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology. 1990.
- 7) M. Bruntink, “Testability of Object-Oriented Systems: a Ölçüts-based Approach”, Amsterdam Üniversitesi, 2003
- 8) R.S. Freedman, “Testability of Software Components”, IEEE Transactions on Software Engineering, Vol. 17, No. 6, Haziran, 1991, syf.553-563.
- 9) R. A. Khan and K. Mustafa, “A Model For Object Oriented Design Quality Assessment”, SPDS Kuşadası Konferansı, 2004
- 10) S. Chidamber and C. Kemerer, “A Ölçüts Suite for Object-Oriented Design”, IEEE Transactions on Software Engineering, Haziran, 1994, syf. 476-492.
- 11) V.R. Basili and W.L. Melo, “A Validation of OO Design Ölçüts as Quality Indicators”, IEEE Tran. on Software Engineering, Vol.22, No. 10, Ekim.1996.
- 12) Nancy S. Eickelmqn and Debra J. Richardson, “What Makes One Software Architecture More Testable Than Another?”, SIGSOFT' 96 Workshop, San Francisco CA USA
- 13) B. Pettichord, “Design for Testability”, Pacific Northwest Software Quality Conference, Portland, Oregon, Ekim 2002
- 14) J.M. Voas and K.W. Miller, “Improving the Software Development Process Using Testability Research”, Proc. International Conference Software Reliability (ISSRE 2002), 2002
- 15) L. Tahvildari and K. Kontogiannis, “A Ölçüt-Based Approach to Enhance Design Quality Through Meta-Pattern Transformations”, Elektrik ve Bilgisayar Müh. Dep., Waterloo Üniversitesi, 2002
- 16) M. Lorenz and J. Kidd. “Object-Oriented Software Ölçüts”. Prentice-Hall Object-Oriented Series, Englewood Cliffs, NY, 1994.
- 17) S. Jungmayr, “Testability During Design”, Teknik Rapor, Hagen Üniversitesi, 2002
- 18) S. Jungmayr, “Testability Measurement and Software Dependencies”, Teknik Rapor, Hagen Üniversitesi, 2002
- 19) <http://mdp.ivv.nasa.gov/index.html>, NASA IV&V Facility Ölçüts Data Program