

Bütünleşik Yazılım Kestirim Yöntemi ve Araç Seti

Proje No: 107E010

Doç.Dr. Onur Demirors

ŞUBAT 2010
ANKARA

Önsöz

Bu projede yazılım işlevsel büyüklük ölçümündeki, farklı işlevsel büyüklük ölçüm yöntemlerinin karşılaştırılabilirliği, işlevsel büyüklük-işgücü ilişkisi ve bireysel yapılan işlevsel büyüklük ölçümü sonuçları arasındaki farklılıklar konuları araştırılmıştır. Proje kapsamında bir birleşik işlevsel büyüklük ölçme yaklaşımı geliştirilmiş ve yaklaşım prototip aracında uygulanmıştır. Proje TÜBİTAK Bilimsel Ve Teknolojik Araştırma Projelerini Destekleme Programı(1001) ile desteklenmiştir.

İçindekiler

| | |
|--|----|
| 1. Giriş | 7 |
| 2. Genel Bilgiler | 8 |
| 2.1 İlgili Literatür Çalışmaları | 8 |
| 2.2 İşlevsel Büyüklük Ölçme Yöntemleri Üzerine Birimizde Gerçekleştirilmiş Çalışmalar | 11 |
| 3. Yöntem | 12 |
| 4. Bulgular | 12 |
| 4.1 Farklı İBÖ yöntemlerinin ve farklı yöntemlerle elde edilmiş ölçüm sonuçlarının karşılaştırılabilirliği | 13 |
| 4.1.1 İBÖ yöntemlerinin kavramsal eşleştirilmesi | 13 |
| 4.1.2 Birleşik yazılım işlevsel büyüklük yöntemlerinde formal gösterimlere yönelik sistematik gözden geçirme | 15 |
| 4.1.3 Varlık-ilişki kavramlarının IFPUG ve COSMIC işlevsel büyüklük ölçümü yöntem terminolojisine eşleştirilmesi | 16 |
| 4.1.4 Bütünleşik yazılım kestirim yaklaşımı geçerlemesi | 17 |
| 4.2 İBÖ İşlevsel Büyüklük ve İşgücü Arasındaki İlişki | 17 |
| 4.2.1 İşlevsel benzerlikler ve işgücü arasındaki ilişki üzerine yapılan çalışma | 17 |
| 4.2.2 İşlevsel Büyüklük Bazlı Regresyon Modellerinin karşılaştırılması | 19 |
| 4.2.3 İşlevsel bileşenler temelinde sinirsel ağ ve regresyon modeli yaklaşımları karşılaştırması | 19 |
| 4.3 Ölçüm Sonuçlarında Bireyler Arasındaki Sonuçlardaki Farklılıklar | 20 |
| 4.3.1 Varlık genelleştirmenin işlevsel büyüklük ölçümüne etkileri | 20 |
| 4.3.2 Bireysel yaklaşımların yazılım işlevsel büyüklük ölçümü sonuçlarına etkisi | 20 |
| 4.3.3 Kişisel varsayımların COSMIC İBÖ yönteminin uygulanmasında etkilerini belirlenmesi kontrollü deneyi | 21 |
| 4.3.4 COSMIC işlevsel büyüklük ölçüm sonuçlarının güvenilirliği durum çalışması | 22 |
| 4.4 İşlevsel Büyüklük Ölçümü İçin Araç Desteği | 23 |
| 4.4.1 Yazılım işlevsel büyüklük ölçüm sürecini destekleyen araç incelemesi | 23 |
| 4.4.2 Yazılım işlevsel büyüklüğü ölçüsünün yazılım proje yönetimi ve ilgili diğer alanlarda kullanımı | 26 |
| 4.4.3 Bütünleşik Ölçüm Araç Prototip Geliştirilmesi | 27 |
| 5. Sonuç | 27 |
| 6. Referanslar | 29 |

Tablo ve Şekiller Listesi

| | |
|---|----|
| Tablo 1- İşlevsel Büyüklük Ölçme Yöntemleri..... | 8 |
| Tablo 2-Varlık Tipleri ve İBÖ Kavramları | 16 |
| Tablo 3-İlişki Tipleri, IFPUG ve COSMIC Kavramları | 16 |
| Tablo 4. Geçerleme Durum Çalışmaları Sonuçları | 17 |
| Tablo 5-İBÖ araç kategorileri ve anahtar boyutları | 23 |
| Tablo 6- İncelenen İBÖ Araçları | 23 |
| Tablo 7-COSMIC İBÖ yöntemini destekleyen araçların değerlendirilmesi..... | 24 |
| Tablo 8-Araç Değerlendirme puan tablosu | 25 |
| Şekil 1-İBÖ yöntemlerinin gelişimi | 8 |
| Şekil 2-IFPUG, COSMIC, MARK II İBÖ yöntemleri kavramsal haritası..... | 14 |
| Şekil 3-Ölçüm sonuçlarında sapmaya yol açan etkenler..... | 22 |
| Şekil 4-COSMIC İBÖ yöntemi ölçüm prosedürü | 24 |

Özet

Yazılım büyüklük ölçme, yazılım mühendisliğinin önemli sorunlarından biri olmaya devam etmektedir. Kapsamlı yazılım geliştirme projelerinde iyi tanımlanmış bir büyüklük temel çizgisi olmadan, gerek yazılım işgücü ve maliyet kestirimi gerekse yazılım proje planlama ve kontrolü olanaksızdır. Yazılım kestiriminden kaynaklanan proje yönetimi ile ilgili sorunları çözüme kavuşturabilmek için yazılım büyüklük ölçme ve kestirimi ile ilgili birçok yaklaşım geliştirilmiştir. Bunların arasında, işlevsel büyüklüğü ölçmek üzere tasarlanmış olanları yaygın olarak kullanılmaktadır. İlk olarak, Albrecht tarafından 1979 yılında geliştirilen yöntemden bu yana geliştirilen yöntemleri iyileştirmek veya uygulanabilecek işlevsel alanları genişletmek üzere bu yöntemin farklı çeşitleri geliştirilmiştir.

Şimdiye kadar uzunca bir yol kat edilmesine karşın işlevsel büyüklük ölçümü (İBÖ) yöntemleri için hala birçok iyileştirme fırsatı bulunmaktadır. Bu araştırma projesi kapsamında, İBÖ ile ilgili farklı İBÖ yöntemlerinin ve farklı yöntemlerle elde edilmiş ölçüm sonuçlarının karşılaştırılabilirliği, İBÖ işlevsel büyüklük ve işgücü arasındaki ilişki, ölçüm sonuçlarında bireyler arasındaki farklı sonuçlar ile İBÖ yöntemleri için araç desteği problemlerine yönelik öneriler getirilmiştir.

Bu bağlamda, farklı İBÖ yöntemleri arasındaki benzerlik ve farklılıklar temel alan bir model ile birleşik bir ölçüm yaklaşımı ve bu yaklaşımla yapılan ölçümleri destekleyebilecek bir prototip araç geliştirilmiştir. Örnek çalışmalarla yaklaşım ve prototip araç geçerlenmiştir. İşgücü-işlevsel büyüklük ilişkisine yönelik, yazılımın sunduğu işlevsellikler arasındaki benzerliklerin ve farklı yazılım işlevsel bileşen tiplerinin büyüklük işgücü ilişkisine etkisi incelenmiştir. Bunun yanında bireysel ölçümlerle elde edilen sonuçlardaki farklılıklar, durum çalışmaları ile irdelenmiş, farklılıkları azaltma yönünde İBÖ yöntemlerinde ve birleşik ölçüm yaklaşımında çalışma konuları ve fırsatları belirlenmiştir.

Abstract

Functional Size Measurement is a challenge in software engineering and industry; without a solid baseline it is almost impossible to perform effective effort and cost estimates, which in turn, results in poor project planning and control. There have been various approaches to the project management problems due to inaccurate software estimation. Among them, the methods and techniques that propose the measurement of the functionality as a size attribute, have been recognized and attracted the industry and research interest. A variety of methods have been developed to increase the applicability and improve the original method Albrecht proposed in 1979. Although there have been significant improvements in FSM, there are still challenges and opportunities in FSM methods to consider.

In this research project, we analyzed and proposed solutions to the four valid problems of FSM; namely, the conceptual associations between FSM methods, Inter-measurer differences, the relation between effort and functional size and tooling support for FSM and its utilization.

A unified approach to FSM coupled with a prototype measurement tool was developed. The approach and the tool were validated with case-studies. The functional size-effort relationship was analyzed with a particular attention to the functional components and the similarities in-between, such that beyond the total functional size of a piece of software, the size measurement results and the associated concepts were taken into account. Besides, the inter-measurer differences were analyzed conducting experiments and case studies and a set of root causes to eliminate have been identified to lead the differences.

1. Giriş

Yazılım proje yöneticileri etkin bir yönetim için geliştirilen yazılım ile ilgili bilgiye ihtiyaç duyarlar. Büyüklük, mühendislik projelerinde ihtiyaç duyulan temel özelliklerden biridir ve proje yönetimini; kapsam belirleme, maliyet ve zaman kestirimi, performans, kalite ölçümü ve sözleşme yönetimi gibi alanlarda, destekler. Yazılım büyüklüğü, yazılım gereksinim, kaynak kodu, fiziksel bileşenleri, dokümantasyonu gibi yazılım ürünlerin birçok özelliği ile ilişkilidir ve yazılım geliştiriciler büyüklüğü çeşitli ölçü ve yöntemlerle ölçmüşlerdir. Yazılım büyüklüğünü uzunluk, işlevsellik ve karmaşıklık boyutları olan çok-boyutlu bir özellik olarak tanımlanabilir .

Geliştirilen çeşitli yazılım büyüklük ölçümü yaklaşımlarından, yazılımın 'işlevselliğini' nicelemeye yönelik ölçü ve yöntemler uygulamada geniş kabul görmüş ve kullanım alanı bulmuştur. Yazılım işlevsel büyüklüğü yazılımın kullanıcıya sunduğu işlevsellik (işlevsellik) miktarını ölçer. Bunun yanında, yazılım büyüklük ölçümü, yazılım mühendisliğinin önemli problemlerinden biri olmaya devam etmektedir..

Yazılım ürünlerinin büyüklük kestirimi ile ilgili çok çeşitli yaklaşımlar getirilmiştir. Güvenilir büyüklük ölçümlerine duyulan ihtiyaç nedeniyle, yazılım büyüklük kestirme/ölçme yöntemleri yazılım mühendisliği dünyasında birçok araştırmacının konusu olmuştur. İşlevsel Büyüklük Ölçme (İBÖ) yöntemleri 1970'lerin sonlarında ilk olarak yayınlanmasından bu güne kadar yazılımın işlevsel büyüklüğünü ölçmede gittikçe daha yaygın olarak kullanılmaktadırlar. Bu yöntemlerin ilgi görmesinin en önemli sebepleri; İBÖ yöntemlerinin yazılım yaşam döngüsünün erken aşamalarında kullanılabilmesi ve yazılım büyüklüğünü kod uzunluğu cinsinden ifade eden yöntemler ve metriklerin tersine, yazılım geliştirme dili ve teknolojisine bağımsız olmasıdır.

Geçen süre zarfında bu yöntemler için pek çok iyileştirme önerileri getirilmiştir. Günümüze dek İBÖ yöntemleri ile ilgili araştırmacılar ve uluslararası araştırma grupları tarafında pek çok çalışma yapılmış olmasına rağmen, bu yöntemlerle ilgili hala birçok iyileştirme fırsatları mevcuttur.

Bu araştırma projesi kapsamında, İBÖ ile ilgili,

- Farklı İBÖ yöntemlerinin ve farklı yöntemlerle elde edilmiş ölçüm sonuçlarının karşılaştırılabilirliği,
- İBÖ işlevsel büyüklük ve işgücü arasındaki ilişki,
- Ölçüm sonuçlarında bireyler arasındaki sonuçlardaki farklılıklar ile
- İBÖ yöntemleri için araç desteği

Problemlerine yönelik öneriler getirilmiştir.

Bu öneriler ışığında araştırmacının temel çıktıları farklı yöntemler arasındaki benzerlikleri ve farklılıkları üzerinden temellendirilen bir bütünleşik ölçüm yaklaşımı, bu yaklaşıma uygun ölçüm yapmayı destekleyen bir araç prototipi, işgücü-işlevsel büyüklük arasındaki ilişkide işlevsel benzerlik ve farklı tip işlevselliklerin etkisi olmuştur.

Raporun ikinci bölümünde "İşlevsel Büyüklük Ölçümü" konusunda çalışma kapsamı ile ilgili literatürde yapılmış araştırmalar hakkında genel bir bilgi verilmiş, üçüncü bölümde araştırmada izlenen yöntem açıklanmıştır. Dördüncü bölümde araştırma yapılan çalışmalar verilmiş, son bölümde araştırmacının özet sonuçları araştırmacının devamı olabilecek araştırma fırsatları açıklanmıştır.

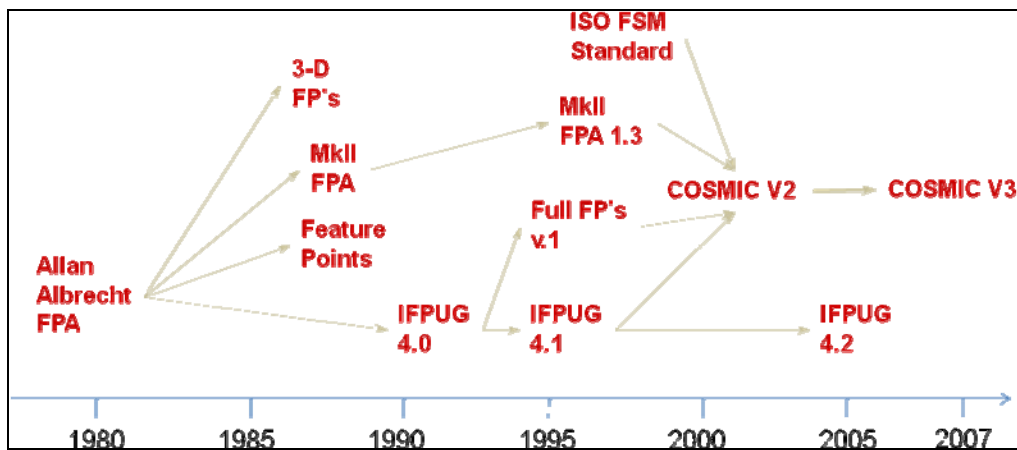
2. Genel Bilgiler

2.1 İlgili Literatür Çalışmaları

Yazılım ürünlerinin işlevsel büyüklüğünü ölçmek için önemli çalışmalar olmuştur. İlk olarak, 1979 yılında, Albrecht yazılım büyüklüğünü ölçmek için Fonksiyon Nokta (FN) metriğini ve Fonksiyon Nokta Analiz (FNA) Yöntemini tasarlamıştır [8]. Bu kullanıcılara sağlanan işlevsellik miktarının FN cinsinden ölçümü düşüncesi üzerine temellendirilmiştir. Daha sonra Albrecht ve Gaffney bu yöntemi geliştirmiştir [9]. Takip eden yıllar içinde, bazı yazarlar orijinal FNA yöntemini geliştirmek ya da uygulanabileceği işlevsel alanları genişletmek üzere yeni yöntemler önermişlerdir [76]. Tablo 1’de yazılım ürünlerinin “işlevsellik” niteliğini ölçmek üzere geliştirilmiş önemli İşlevsel Büyüklük Ölçme (İBÖ) yöntemleri listelenmiştir. de yaygın olarak kullanılan İBÖ yöntemlerinin gelişimi verilmektedir. Şekil 1’de yöntemlerin zaman içinde gelişimi gösterilmiştir.

Tablo 1- İşlevsel Büyüklük Ölçme Yöntemleri

| Yöntem | Geliştiren |
|--|--|
| Albrecht FNA Yöntemi / IFPUG FNA Yöntemi | Albrecht [8], [34], Uluslararası FN Kullanıcıları Grubu (IFPUG) [45] |
| DeMarco’s Bang Metrikleri | DeMarco [18] |
| Nitelik Nokta | Jones [50] |
| Mark II FNA Yöntemi | Symons [75], [67] [46] |
| NESMA FNA Yöntemi | Hollanda Yazılım Metrikleri Kullanıcıları Kurumu (NESMA) [69], [47] |
| ASSET- R | Reifer [72] |
| 3-D FN | Whitmire [80] |
| Nesne Nokta | Kauffman and Kumar [53], Banker et al. [11] |
| Matson, Barret ve Mellichamp FNA Yöntemi | Matson et al. [62] |
| Full FN (FFN) | University of Quebec - Software Engineering Lab. in Applied Metrics [2]. |
| Erken FNA (EFNA) | Meli [63], [64], Conte et al. [17] |
| Nesne Yönelimli FN | Caldiera et al. [16] |
| Tahmini Nesne Nokta | Teologlou [77] |
| COSMIC FFN | Ortak Yazılım Ölçme Uluslararası Konsorsiyumu (COSMIC) [1], [44], [78], [79] |
| Erken ve Hızlı COSMIC FFN | Conte et al. [17], [65] |
| Nesne Yönelimli Metot FN | Pastor and his colleagues [70] |
| Kammelar’ın Bileşen Nesne Nokta Yöntemi. | Kammelar [52] |
| FiSMA İşlevsel Büyüklük Ölçme Yöntemi | Finlandiya Yazılım Metrikleri Kurumu (FiSMA) [28] |



Şekil 1-İBÖ yöntemlerinin gelişimi

Bu yöntemlerin zamanla evrimleşmesi, yöntemler arasında kavramsal tutarsızlıklara sebep oldu. 1996 yılında, Uluslararası Standart Örgütü (ISO) İBÖ yöntemlerini kavramsal temelini ve ortak prensiplerini netleştirmek üzere bir çalışma grubu başlattı. 1998 yılında ISO/IEC 14143 yayınlandı [37] ve 2007 yılında güncellendi [38]. Bu standart İBÖ ile ilişkili İşlevsel Kullanıcı gereksinimleri (IKG), İşlevsel Büyüklük, Temel İşlevsel Bileşenler (TFB), TFB Tipi, İşlevsel Alan gibi temel kavramları ve aday bir İBÖ yönteminin karşılanması gereken İBÖ gereksinimlerini tanımladı. 2000 yılında, ISO/IEC 14143-1:1998'in IEEE yorumu ile ilgili uygulama notları tanımlandı [33].

Takip eden yıllarda bu standardın 5 bölümü daha yayınlanmıştır. Bunlar;

- ISO/IEC 14143-2: ISO/IEC 14143-1:1998'na göre yazılım büyüklük ölçüm yöntemlerinin uyumluluk değerlendirilmesi [39],
- ISO/IEC TR 14143-3: İBÖ yöntemlerinin Doğrulanması [40],
- ISO/IEC TR 14143-4: İBÖ – Referans Modeli [41],
- ISO/IEC TR 14143-5: İBÖ kullanımı için İşlevsel Alanların Kararlaştırılması [42].
- ISO/IEC FCD 14143-6: ISO/IEC 14143 ve ilgili uluslararası standartların kullanımı için rehber [43].

Araştırma projesi başlangıç dönemi itibarıyla dört yöntem, IFPUG FNA [45], Mk II FNA [46], NESMA FP [47] ve COSMIC FFN [44], ISO tarafından uluslararası İBÖ standardı olarak sertifikalandırılmıştır.

Albrecht / IFPUG FNA Yöntemi: 1979 yılında önerilen İlk İBÖ yöntemi modeli görece olarak basitti. Girdi, Çıktı, Sorgu ve Dosya olmak üzere dört tip fonksiyon ve ayarlama etmeni ile beraber, her fonksiyon için tek bir ağırlık içeriyordu. 1983 de Albrecht ve Gaffney yöntemin değiştirilmiş bir sürümünü sundular [9]. Yeni sürüm üç fonksiyon karmaşıklık seviyesi, fonksiyon tipine göre karmaşıklığın değerlendirilmesi için kuralları ve bu kurallarda kullanılmak üzere karşılık gelen ağırlıklar tablosunu getirdi. Önceki “Dosya Tipi”, “İçsel Mantık Dosyası” ve “Dışsal Arabirim Dosyası” olmak üzere iki alt tipe ayrıştırıldı. Bu sürümdeki fonksiyon tipleri Dışsal Girdi, Dışsal Çıktı, Dışsal Sorgu, İçsel Mantık Dosyası ve Dışsal Arabirim Dosyası olarak adlandırıldı.

1985 yılında, IBM kullanıcı grubu (GUIDE) Albrecht'in temel tanımlarını beş fonksiyon tipinin İşlevsel karmaşıklık derecelendirilmesi (düşük, orta ve yüksek) için kurallar koyarak, İBÖ kurallarını daha kesin, açık hale getirmek ve saptamak üzere tekrar gözden geçirdi. Derecelendirmede kullanım amaçlı, bir tanesi mantıksal dosyalar ve iki tanesi aralık değerleri önceden belirlenmiş işlemler için üç adet -iki boyutlu matris -oluşturdular. Bu uzmanlar arasında tutarlı ölçümlere olanak tanıdı [3].

1986 yılında, bu yaklaşımın direk takipçisi olmak için bir tasarım otoritesi olarak Uluslararası Fonksiyon Nokta Kullanıcıları Grubu (IFPUG) kuruldu. IFPUG, halen sayma kurallarını netleştirmektedir. Resmi IFPUG Sayma Uygulamaları El Kitabı sürümleri IFPUG 1986, 1988, 1990, 1994, 1999 ve 2004'tür.

IFPUG İBÖ'nde, temel İşlevsel bileşenler, son-kullanıcı bakışıyla, dış girdiler, dışsal çıktılar, dışsal sorgular, dışsal arabirim dosyaları ve mantıksal içsel dosyalar olarak sınıflandırılmaktadır. Daha sonra, bu bileşenler sayılır ve içerdiği Veri Eleman Tipi, Kayıt eleman Tipine ve değiştirilen dosya sayısına dayanarak, her bir sayı için bir ağırlık atanır. Sonra, bu ağırlıklar toplanır ve fonksiyon nokta cinsinden ayarlanmış bir büyüklük üretmek üzere sonuçtaki toplam, Değer Ayarlama Etmeni (DAE) ile ayarlanır. VAF, sayılan uygulamanın genel İşlevselliğini derecelendiren 14 genel sistem özelliğine üzerine [44] temellendirilmiştir.

IFPUG İBÖ, ISO/IEC 14143 ile uyumluluğu onaylanmıştır ve 2003'te uluslararası bir ISO standardı olmuştur [45].

Mark II Fonksiyon Nokta Analizi (MK II FNA): İngiliz “Mk II FNA” yöntemi, Charles Symons tarafından 1998 yılında, olağan İBÖ yönteminin yetersizliklerini gidermek amacıyla geliştirilmiştir. İngiliz yazılım metrikleri birliğine bağlı metrik uygulamaları komitesi yöntemin şu anki tasarım otoritesidir. Mark II Fonksiyon Nokta Analizi, ISO/IEC 14143 uyumlu olarak onaylanmıştır ve 2002 yılında uluslararası bir ISO standardı olmuştur [46].

Tanıtımından bu yana, MK II FNA, birçok yerde, artarak kullanılmaktadır. MK II FNA bilgi işlenmesini ölçmeyi hedefler. Bu yöntem sistemi bir mantıksal işlemler kümesi olarak görür ve yazılımın İşlevsel

Büyükliğini, her bir mantıksal işlem için Girdi Veri Eleman Tiplerini, Başvurulan Veri Varlık Tiplerini ve Çıktı Veri Eleman Tiplerini sayarak hesaplar. Albrecht/IFPUG gibi İş bilgi sistemlerini gibi ölçmek için tasarlanmıştır. Yöntemin bilimsel ve gerçek-zamanlı yazılım gibi diğer alanlarda uygulanması mümkündür ama yöntemin bazı değişikliklere gereksinim duyulabilir [67].

NESMA Fonksiyon Nokta Analizi: 1989 yılında, Hollanda Yazılım Metrikleri Kullanıcıları Kurumu (NESMA), Hollanda Fonksiyon Nokta Kullanıcıları Grubu (NEFPUG) olarak kuruldu. Bu Avrupa'daki en büyük FNA kullanıcı grubudur. Fonksiyon Nokta Analizi Uygulaması için Sayma Rehberi ve Tanımlar (NESMA CPM 1.0) el kitabı ilk sürümü 1990 yılında yayınlandı. Bu yöntem de IFPUG FNA prensipleri üzerine dayanır. İşlevselliği ölçmek için kullanılan fonksiyon tipleri dışsal girdiler, dışsal çıktılar, dışsal sorgular, dışsal arabirim dosyaları ve mantıksal içsel dosyalar olmak üzere IFPUG FNA ile aynıdır. NESMA FNA yöntemindeki fark, sayma uygulamaları el kitabının daha somut yol gösterimler, ipuçları ve örnekler vermesidir [69]. NESMA Fonksiyon Nokta Analizi, ISO/IEC 14143 uyumlu olarak onaylanmıştır ve 2003'te uluslararası bir ISO standardı olmuştur [47].

COSMIC Ful Fonksiyon Nokta: FFN yöntemini ikinci sürümü olan "COSMIC FFN" yöntemi Kasım 1999'da Ortak Yazılım Ölçümü Uluslararası Konsorsiyumu (COSMIC) tarafından yayınlanmıştır [1]. Bu grup bu yeni yöntemi, iş uygulamaları (Yönetim Bilgi Sistemleri - YBS ya da 'veri zengin') yazılımlarının ve gerçek zamanlı yazılımların her ikisi için ya da bunların karışımı olan yazılımların İşlevsel büyüklüğünü ölçecek standart bir yöntem olarak geliştirmek amacıyla kuruldu (COSMIC, 2003). Birçok saha denemeleri yapıldı ve sonuçları yayınlandı [26]. COSMIC Tam Fonksiyon Nokta ISO/IEC 14143 uyumlu olarak onaylanmıştır ve 2003 yılında uluslararası bir ISO standardı olmuştur [44].

COSMIC FFN yöntemi İşlevsel Kullanıcı Gereksinimleri ('İKG') temelli bir İşlevsel Büyüklük ölçmek için tasarlandı. İKG kalite ve teknik gereksinimleri dışlar. Bir "yazılım parçasının" İşlevsel büyüklüğü, ister yazılım sadece bir İKG ifadesi olarak var olsun, ister İKG var olan bir yazılım parçasıyla ifade edilsin ya da her iki ifade arasında bir durumda olsun, ölçülebilir. Yazılım İşlevsel Büyüklüğü Giriş, Çıkış, Okuma ve Yazma olmak üzere dört temel İşlevsel Bileşen Tipi sayısına dayanılarak ölçülür.

İşlevsel Büyüklük Ölçme Yöntemlerinin Değerlendirilmesi ve Karşılaştırılması Üzerine Yapılan Çalışmalar:

Bu yöntemlerin uluslararası İBÖ standartları olarak tanınmasına ve İşlevsel büyüklüğü kullanıcıya sağlanacak olan "işlevsellik" miktarını nicelleştirerek ölçmesine rağmen her bir yöntemin işlevselliği nasıl tanımladığı ve ölçüldüğü ile ilgili belirgin farklılıklar vardır.

İBÖ yöntemlerinin değerlendirilmesi ve karşılaştırılması üzerine belirli sayıda çalışma vardır. Lothar ve Dumke İBÖ yöntemlerini belli sayıda kritere göre değerlendirmiş ve İBÖ sorunlarını tartışmıştır [60]. Fetcke ortaklıklarını ve farklılıklarını vurgulayarak, farklı İBÖ yöntemlerinin genelleştirilmiş gösterimi olan bir model önermiştir [26]. Demirörs ve Gencel MK II FNA Yöntemi, Jones'un Çok Erken Büyüklük Tahminleme Yöntemi ve Erken FNA yöntemini, yazılım yaşam döngüsünün erken fazlarında bir örnek projeye uygulayarak değerlendirmişlerdir [19]. Diğer bir çalışmada, Demirörs ve Gencel İBÖ yöntemleri arasındaki farklılıkları tartışmış ve bu farkların işlevsel büyüklükler üzerinde oluşturduğu etkileri bir örnek proje ile açıklamışlardır [20]. Rule, IFPUG FNA ve Mk II FNA arasındaki benzerlikleri ve farklılıkları tartışmıştır [74]. Rollo, Web uygulamalarının ölçülmesi ile ilgili problemleri incelemiş ve IFPUG FNA, Mk II FNA ve COSMIC FFN yöntemlerini örnek bir e-ticaret uygulamasında uygulamıştır [73]. Mk II FNA ve COSMIC FFN yöntemlerinin gerçek zamanlı bir yazılım sistemine uygulamasıyla alınan sonuçlar gösterilmiş ve bu yöntemlerin ölçüm süreçlerindeki benzerlikler ve farklılıklar ile ölçüm boyunca karşılaşılan zorluklar tartışılmıştır [30].

Bu karşılaştırma çalışmalarının beraberinde, İBÖ yöntemlerinin kuramsal temellerinde belirgin ilerlemeler de alındı. Lokan, FNA yöntemindeki Temel İşlevsel Bileşen (TFB) tipleri arasındaki korelasyonlar üzerine çalışmıştır [58]. Korelasyonlar üzerinde daha derinleşmek üzere Uluslararası Yazılım Karşılaştırma Standartları Grubu (ISBSG) veri kümesini analiz etmiştir [35], [36]. Kitchenham ve Kansala [54] ve Jeffery ve Stathis [48] IFPUG FNA TFB tipleri arasındaki korelasyonları çalışmıştır. Bulguların bazıları önceki çalışmalar ile örtüşmesine rağmen, diğerlerinde farklı korelasyonlar bulunmuştur. Bu çalışmaların sonuçları, bu korelasyonların varlığı FNA sırasında aynı şeylerin birden fazla kere sayıldığını göstermiştir. Dahası, Kitchenham, korelasyonlar üzerinde farklı sonuçların bulunmasının, öğelerin toplamına dayanan herhangi bir önceden tahmin modelinin farklı veri kümeleri

için istikrarsızlığını gösterdiğini belirtmiştir [55]. Daha sonra Fenton, yazılım mühendisliğini daha sağlam bir temele dayandırabilmek için ölçme teorisine dayanması gerektiğini belirtmiştir [24], [25]. Bazı yazarlar ölçme teorisine göre İBÖ yöntemleri kuruluşundaki temel kusurları, özellikle ölçek tipleri [55], [3], [4] ve toplam işlevsel büyüklüğe ulaşmak için farklı TFB'lerin ya da TFB bileşen öğelerinin büyüklüklerinin eklenebilirliği konularında tartışmışlardır [55], [24], [3].

Yazılım Büyüklüğü ve Geliştirme İşgücü Arasındaki İlişki Üzerine Gerçekleştirilen Çalışmalar:

Yazılım geliştirme projelerinin planlanması, takibi ve kontrolünde gerekli olan işgücü ve maliyet kestirimi, proje yöneticileri ve araştırmacılar için üzerinde inceleme ve araştırma yapılması gereken önemli konulardan biri olma özelliğini korumaktadır. Yazılım işgücü kestirim modelleri ve teknikleri ile ilgili geniş bir literatür bulunmasına rağmen, yazılım büyüklüğünün işgücü ile ilişkisi üzerinde tartışmalar sürmektedir ([5], [12], [13], [32], [49], [51], [56]). Birçok kestirim modelinde işgücü ile ilişkili işlevsel olmayan faktörler dikkate alınmıştır - COCOMO ve türevi kestirim yöntemleri gibi [12], [13].

Leung ve Fan işgücü kestirim yöntemlerinin güçlü ve zayıf yönlerini tartışıp, tatmin edici olmadıklarını düşündükleri var olan modellerin performanslarını ve yazılım kestirimine yeni yaklaşımları değerlendirmişlerdir [57]. Benzeri şekilde işgücü ve maliyet kestirimi için gerçekleştirilmiş bazı çalışmalar ile değerlendirilip karşılaştırılmıştır [5], [14], [15], [66]. Çalışmaların tümünden, farklı gruplar ve yazılım alanlarında kullanılan modellerin henüz evrensel olarak kabul görmediği sonucu çıkarılmıştır.

Benzeri şekilde, işlevsel büyüklük ve işgücü arasındaki ilişki, birçok çalışmada incelenmiş ve herhangi bir modelin yazılım geliştirme projelerinin ihtiyaçlarını ve beklentilerini yeterince karşılamadığı görülmüştür. ISO standardı olmuş IFPUG ve MkII İBÖ yöntemleri, elde edilen işlevsel büyüklükleri işgücüne çevirmek için öneriler ve prosedürler önerirken, COSMIC yöntemi bu konuda bir öneri getirmemiş, ölçülen işlevsel büyüklüğü herhangi bir işgücü kestirimi metodu için bir girdi olarak görmüştür.

İşlevsel büyüklük ile işgücünün ilişkilendiren çalışmalar genel olarak işlevsel büyüklüğü, ölçüm sonucu elde edilen farklı Temel İşlevsel Bileşen (TFB) tiplerinin toplamından oluşmuş tek bir değer olarak görür. Abran et al. [6] yaptığı çalışmada bir yazılımdaki farklı fonksiyon tiplerini dağıtarak yazılım işlevsel profili kavramını oluşturmuş, büyüklük-işgücü ilişkisinin belirginliğini fonksiyon tiplerine göre incelemiştir. Diğer bir çalışmada ISBSG'na [35] ait COSMIC yöntemi ile ölçülmüş projelerin verilerini kullanarak işlevsel profillerin belirlenmesinin ve seçilmiş örneklerle karşılaştırılmasının, projenin profiline uygun en iyi kestirim modelinin seçilmesinde yardımcı olabileceği gözlemlenmiştir [7]. Gencel et.al. [29] çalışmada fonksiyonlute tipi büyüklüklerini çok boyutlu bir ölçüm ile tanımlamış ve her işlevsel tipin işlevsel büyüklüğü ile fonksiyonluteyi geliştirmek için gerekli olan işgücünün ilişkisini bulmak üzere deneysel çalışmalar yapılması gerektiğini belirtmiştir.

Yayınlanmış diğer çalışmalarda büyüklük-işgücü arasındaki ilişkiye etkisi olan diğer faktörler incelenmiş, ekip büyüklüğü, programlama dili, organizasyon tipi, iş alanı, uygulama tipi ve geliştirme platformu faktörleri farklı belirginlik seviyelerinde etki oluşturduğu gözlemlenmiştir [10], [27], [59], [61], [68], [71]. Aralarında en belirgin faktörlerin 'ekip büyüklüğü, iş alanı ve uygulama tipi' olduğu belirtilmiştir [10], [27].

2.2 İşlevsel Büyüklük Ölçme Yöntemleri Üzerine Birimizde¹ Gerçekleştirilmiş Çalışmalar

Araştırma projesi öncesinde birimizde gerçekleşen belli başlı çalışmalar arasında "Mimari Boyutlara Dayanan Yeni Bir Yazılım İşlevsel Büyüklük Ölçme Yöntemi **Error! Reference source not found.**" adlı bir doktora tezi ve "Yazılım İşlevsel Büyüklük Ölçme Yöntemleri İçin Birleştirme Yöntemi Ve Yöntemi Destekleyen Bir Araç **Error! Reference source not found.**" ve "İşlev Puanı Çözümleme Yönteminin Nesne Tabanlı Çözümleme Ve Tasarım Yönteminde Otomasyonu **Error! Reference source not found.**" adlı yüksek lisans tezleri mevcuttur.

Mimari Boyutlara Dayanan Yeni Bir Yazılım İşlevsel Büyüklük Ölçme Yöntemi" adlı tez çalışması; "işlevsellik" ve "işlevsel büyüklük" kavramlarının eleştirel olarak incelenmesini ve yazılım mimari öğeleri ile farklı yazılım işlevsel alan tiplerindeki bilgi işleme mantığı biçimlerini dikkate alarak "işlevsellik tipleri" nin belirlenmesini içermektedir.

¹ Yazılım Yönetimi Araştırma Grubu, Enformatik Enstitüsü, Orta Doğu Teknik Üniversitesi.

Getirilen öneriler kapsamında Mimari Boyutlara Dayanan İBÖ (ARCHI DIM) olarak adlandırılmış yeni bir yöntem geliştirilmiştir.

“Yazılım İşlevsel Büyüklük Ölçme Yöntemleri İçin Birleştirme Yöntemi Ve Yöntemi Destekleyen Bir Araç” adlı tez çalışmasında; dört farklı İBÖ yöntemi olan IFPUG FPA, Mk II FPA, COSMIC FFP ve ARCHI DIM yöntemleri arasındaki benzerlikler ve farklılıklar incelenmiştir ve ortak temel kavramlar sunulmuştur. Bu dört yöntem için bir birleştirme modeli önerilmiş ve bu modeli kullanan bir araç tasarlanmış ve geliştirilmiştir.

“İşlev Puanı Çözümleme Yönteminin Nesne Tabanlı Çözümleme Ve Tasarım Yönteminde Otomasyonu” adlı tez çalışmasında ise nesneye dayalı çözümleme ve tasarım platformu içinde, daha verimli yazılım boyut belirlemeleri için bir otomasyon yöntemi sunulmuştur. İşlev Puanlarını belirlemede MK II FPA yöntemi kullanılmıştır. Nesneye dayalı çözümleme ve tasarım sürecinin ifade edilmesi için UML yöntemi ve gösterimi uygulanmıştır. Ayrıca, MK II FPA Analizinin geliştirilen yöntemi kullanarak yapılmasını sağlayan ve Rational firmasının Rose Yazılımı ile hazırlanmış modellerin otomatik olarak kullanılabilmesini sağlayan bir yazılım geliştirilmiştir.

3. Yöntem

Araştırmada, proje kapsamındaki araştırma konuları ile ilgili çalışmaların uygulanabilir sonuçlarının yansıtılabileceği birleşik ölçüm yaklaşımı ve ölçüm prototip aracı geliştirmesi temel eksen kabul edilmiştir. Konular beş aşama altında incelenmiş ve çalışılmıştır. Bunun yanında bu aşamalarda yapılan her bir araştırma çalışmasında izlenen yöntemi (çoklu-durum çalışması, sistematik gözden geçirme, istatistiksel model vb.), çalışma içinde anlatılmıştır.

Literatür Taraması ve Oryantasyon: Aşamada, araştırma konusu olan problemlere dair yapılmış yayınlar, hazırlanmış bildiriler, teknik raporlar ve basılmış kitaplar taranarak, konu ile ilgili güncel birikimin derlenmesi amaçlanmıştır. Proje kapsamında odaklanılacak, ISO uyumlu bilinen ve yaygın yöntemler olan COSMIC FFP, IFPUG FPA ve MARK II yöntemleri ile örnek ölçüm uygulamaları yapılarak, araştırma konuları ile ilgili gözlemler yapılmış, araştırma grubuna pratik birikim kazanılmıştır. Literatür taramasında yazılım mühendisliği ile ilgili belli başlı basımevleri yayınları, dergi ve konferans bildirileri kapsama alınmıştır.

Yöntem Geliştirme

Her üç yöntem için aynı anda ölçüm yapılmasını destekleyecek birleşik bir ölçüm yaklaşımı geliştirilmesi yöntemler arası benzerliklerin ve farklılıkların analizi üzerinden yapılmıştır.

Araç Prototipi Geliştirme

Araç geliştirmede, projenin her aşamasında araç ile uygulanabilecek ölçüm önerilerinin ve ölçümlerde toplanması gerebilecek farklı ölçüm öznitelikleri belirlenebileceği öngörülerek artımlı yaşam döngüsü uygulanmıştır.

Geçerleme

Geçerleme çalışmalarında, gerçek yazılım projeleri birleşik yöntem ile elde edilen ölçüm sonuçları yöntemlerin direkt uygulanması ile elde edilen sonuçlarla karşılaştırılmıştır. Karşılaştırmalarda yalnız toplam işlevsel büyüklük değil, farklı tiplerde belirlenen işlevsel bileşenler de dikkate alınmıştır.

İyileştirme

Bu aşamada, proje boyunca gerçekleştirilen, hedef probleme yönelik çalışmaların sonuçlarının değerlendirilip, ölçüm yaklaşımında, araçta ve İBÖ yöntemlerinde yapılabilecek iyileştirmeler belirlenmiştir.

4. Bulgular

Araştırma boyunca yapılan çalışmaların özetleri ve elde edilen sonuçlar, araştırma problemleri başlıkları altında verilmiştir.

4.1 Farklı İBÖ yöntemlerinin ve farklı yöntemlerle elde edilmiş ölçüm sonuçlarının karşılaştırılabilirliği

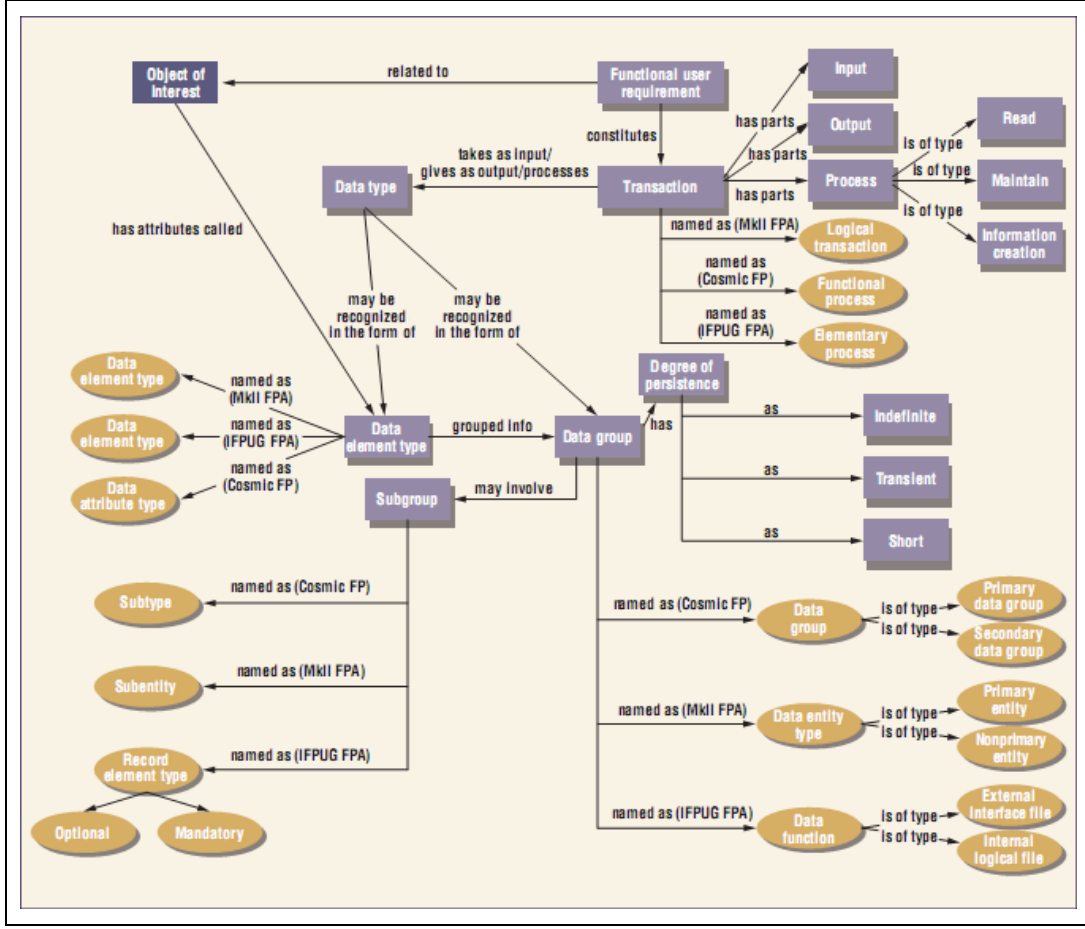
4.1.1 İBÖ yöntemlerinin kavramsal eşleştirilmesi

İBÖ yöntemleri aşağı yukarı benzer sonuçlar verse de, ne aynı nitelikleri ölçmek ne de aynı değerlendirme ölçeğini kullanmak üzere tasarlanmışlardır. Bunun sonucu olarak, bir yazılım ürünü farklı yöntemler kullanılarak ölçüldüğünde farklı işlevsel büyüklüklere sahip olmaktadır. Her yöntemin kendi işlevsel büyüklük tanımı vardır ve işlevsel kullanıcı gereksinimlerinin (İKG) farklı varlıkları için ayrı sayım şemaları kullanırlar.

Bunun da ötesinde, İBÖ yöntemleri, genel kavramlarıyla bir yazılım gelişim metodolojisi (yöntem bilimi) kullanmak yerine kendi yazılım mühendisliği kavramlarını kullanmayı tercih etmektedir [26]. Yöntemler, ölçüm sürecinde farklı kısaltmalar kullanır ya da aynı kavramlara farklı bakış açılarından yaklaşırlar [23],[31]. Bu farklılıklar nedeniyle yöntemlerin kurallarını ve sayım şemalarını birbirleriyle karşılaştırmak uygun değildir.

Yöntemlerin farklı işlevsel alan tiplerine uygulanabilirlikleri veya yazılım gelişim yaşam döngüsünün farklı fazlarına ne şekilde uygulanabilirlikleri tanımlanmamıştır. Bu da farklı vakalarda uygun ölçüm yöntemlerinin seçilmesinde zorluklar yaratmaktadır. Organizasyonlar için de bir yöntemi benimseyip geçmiş verileri topladıktan sonra o ölçüm yöntemini değiştirmek zor olmaktadır. Farklı yöntemlerle ölçülmüş olan yazılım ürünlerinin işlevsel büyüklüklerini karşılaştırmak için, ölçüm sonuçlarının birbirlerine dönüştürülebilir olması gerekmektedir. Buna ek olarak, dönüşüm formüllerinin iyi tanımlanmamış olması, karşılaştırma verilerinin yayınlanmasında ve kullanılmasında zorluklara yol açmaktadır.

Bu çalışmada [31], ölçüm süreçlerindeki benzerlikler ve farklılıklar ile IFPUG FPA, MkII FPA ve COSMIC FFP yöntemleri kavramlarının bir haritası çıkarılmış (Şekil 2), yöntem kavramları arasındaki kavramsal benzerlikler ve farklılıklar ortaya konmuştur. Bulgulardan yola çıkarak, bu yöntemler tarafından kullanılan ortak ölçüm kavramları için bir birleşim terminolojisi belirlenmiş ve bu yöntemlerin ölçüm süreçleri için uygun bir soyutlama yaparak, bir Birleşim Modeli (BM) önerilmiştir. Model, bir yazılım sisteminin sadece tek bir veri kümesi kullanılarak farklı yöntemler tarafından ölçülmesine olanak sağlamayı [23], böylece, hem işlevsel büyüklüğün dönüştürülmesine olan ihtiyacı hem de bir organizasyon ya da projede farklı ölçüm yöntemlerinin kullanılması ihtiyacını karşılamayı amaçlamaktadır.



Şekil 2-IFPUG, COSMIC, MARK II İBÖ yöntemleri kavramsal haritası ([81]'den alınmıştır)

Model, yapılan örnek olay incelemeleri ışığında ortaya konmuştur. Bu örnek olay incelemeleri, iki yazılım sisteminin işlevsel büyüklüğünün IFPUG FPA, MkII FPA ve COSMIC FFP yöntemleri ile yöntemlerin sayım kılavuzunda belirtilen kuralları ilke alınarak ölçülmesi ve önerilen Birleşim Modelinin uygulanması çalışmalarını içermektedir.

Bu çalışmalar sonucunda İBÖ yöntemlerinin birleşiminin mümkün olduğu ve farklı yöntemlerin büyüklüklerine tek bir ortak veri kümesi kullanılarak erişilebileceği ortaya konmuştur. Bu şekilde işlevsel büyüklüğün eş zamanlı olarak birden çok yöntem ile ölçülmesi mümkün olmuştur. Bunun yanında, model İBÖ yöntemlerinin arasındaki ilişkiler ile ilgili olarak biçimsel (formal) bir sunum sağladığı için, ölçümün otomasyonu mümkün olmaktadır. Örneğin, büyüklük ölçümünde düşük seviyede temel işlevsel bileşenleri kullanan yöntemlerden biri ile yapılan ölçümü diğer yöntemlerden birine dönüştürmek mümkün olmaktadır (MkII FPA den COSMIC FFP ye dönüşüm gibi).

Model aynı zamanda aynı ürünün farklı yöntemlerle ölçülmüş olan işlevsel büyüklük ölçüm değerlerinin toplanmasına da olanak sağlamaktadır. Kullanıcılar, kolaylıkla ölçüm sonuçlarından karşılaştırma verileri oluşturarak bu verileri yayınlayıp, istatistiksel yöntemler kullanılarak da dönüşüm formüllerinin geçerliliğini onaylanabilirler. Model ayrıca bir ölçüm yöntemi ile ilgili verileri toplayarak önemli bir miktarda geçmiş veri elde etmiş organizasyonların da büyüklük ölçüm yöntemini değiştirmelerinde yardımcı olmaktadır.

Sonuçlar, ölçüm sürecinde ve kurallarında uygun soyutlamalar ile farklı işlevsel ölçüm yöntemlerinin birleştirilme olasılığı olduğuna bir kez daha göstermiştir.

4.1.2 Birleşik yazılım işlevsel büyüklük yöntemlerinde formal gösterimlere yönelik sistematik gözden geçirme

Bütünleşik Yazılım İşlevsel Büyüklük yöntemi, en bilindik işlevsel büyüklük ölçme yöntemlerinde, işlevselliği ifade eden kavramların benzerlikleri üzerinde temellendirilmiştir. Doğal dille ifade edilen yöntem kavramlarının ve aralarındaki ilişkilerin, formal olarak ifade edilerek, Yöntemin iyileştirilmesinde ve bir büyüklük ölçümü otomasyon aracı ile gelişimine katkı sağlayacağı düşünülmektedir.

Bu çalışmada, işlevsel büyüklük ölçüm yöntemleri ve büyüklük ölçüm uygulamalarına dair bugüne kadar yapılmış formalizasyon çalışmalarını bir bütün olarak ele alınıp, büyüklük ölçümünde ve yöntemlerin anlaşılabilirliğinde ne gibi katkılar sunduğu değerlendirilmiştir. Çalışmada, işlevsel büyüklük ölçümünde formalizasyon getiren öneriler güncel yöntem sürümleri ile karşılaştırmalı olarak değerlendirilmiş, getirilen önerilerin uygulanabilirliği değerlendirilmiştir.

Tarama sonucu belirlenen ve analiz edilen yayınlar (dört dergi makalesi, dört konferans bildirisi, dört teknik rapor), amaçlarına göre net bir şekilde ayrıştırılabilen iki kategoriye bölünmüştür. Her iki kategorideki çalışmaların ortak olan motivasyonlarını, İBÖ yöntem kavram ve kurallarındaki belirsizlikleri ortadan kaldırmak, farklı kural ve kavram yorumlamaları ile oluşan ölçümcüler arası değişkenlikleri azaltmak, ölçüm hatalarını azaltmak ve otomasyon fırsatları oluşturmuştur. Kategoriler arası farklılaşma, formalizasyonlar uygulayarak sorunlara tepki verme biçimlerindedir. İlk kategorideki çalışmalar formalizasyonun belli bir notasyon, spesifikasyon dili yada model ile tanımlanmış yazılım işlevselliğinin ölçümüne nasıl katkıda bulunabileceğini anlatmaktadır. Bu kategori “Formel Spesifikasyonlardan İşlevsel Büyüklük Ölçümlerindeki Formalizasyon Çalışmaları” olarak adlandırılmıştır. İkinci kategori çalışmaları İBÖ yöntem kavramları ve ölçüm fonksiyonları için formal gösterimler önermiştir ve kategori “İBÖ Temellerindeki Formalizasyonlar” olarak adlandırılmıştır.

Formel Spesifikasyonlardan İBÖ Ölçümündeki Formalizasyonlar: İBÖ yöntemleri, yazılımı işlevsellikle olduğunu düşündüğü kavramları bir soyut yazılım modeli ile gösterir. Modeldeki kavramlar(model elemanları) , yöntemin kurallarının işlevsel kullanıcı gereksinimlerine uygulanmasıyla belirlenir. Bu kategorideki çalışmaların konusu, doğal dilden farklı olarak formal spesifikasyonlar (Unified Modeling Language, Entity Relationship vb.) ile gösterimleri yapılan işlevsel kullanıcı gereksinimlerinin büyüklüğünün ölçümünde, işte bu kuralların ve kavramların yorumlanıp formal gösterimlerle ifade edildiği çalışmalardır. Çalışmalarda hedef İBÖ yöntemi analizleri yapılmış, yöntemlerde yoruma açık kısımlar ve spesifikasyon dili ile eşleşmeyen kavramlar tespit edilmiştir. Önerilen formalizasyonların etkinliği yöntem kurallarının bir spesifikasyon dili için yorumlarının geçerliliği ile belirlenmektedir. Bu çalışmaların kapsamı çalışılan spesifikasyon dili için geçerli kalmıştır.

İBÖ Temellerindeki Formalizasyonlar: Bu kategori çalışmalarının konusu yöntemlerin bağlı oldukları soyut yazılım modellerinin yapılarının benzerlik temelinde formal biçimde gösterimleri olmuştur. Yöntemler arası farklar, benzerlikler formal şekilde ifade edilebilmiştir. Bu bağlamda öneriler, İBÖde yöntemler arası karşılaştırmaları daha açık yapabilme, birkaç yöntem için eş zamanlı ölçüm yapabilme ve yöntem yapılarını anlayabilme konularında etkili olabilmektedir.

Yöntemlerin model bileşenleri formal olarak ifade edilebilse de, bileşenlerin geçerliliği, her yöntemde farklı olan ve yalnız yöntemde geçerli olan bileşen kuralları formal olarak ifade edilmemektedir.

Bu çalışmada [31] değerlendirilen ve İBÖ literatüründe bulunan diğer birçok çalışma ortak bir görüşle, ölçümlerde işlevsel gereksinimlerin soyut yazılım modellerine yorumlamalar olmaksızın dönüştürülmesi, pratikte, işlevsel büyüklüğün doğru ve tam olarak ölçülemediği saptamasında bulunmaktadır. Çeşitli çalışmalar, yazılım gereksinimlerin ifade edilme şekilleri ile İBÖ yöntemleri yazılım modeli arasındaki bu “boşluğu”, yöntem kavramlarını ve kurallarını sistematik şekilde yorumlayarak ya da özelleştirerek doldurmak için öneriler getirmişlerdir. İlk kategoride değerlendirilen çalışmalar da benzeri çalışmalardır fakat bu boşluğu formal gösterimler ile doldurmakla farklılaşmaktadırlar. İşlevsel kullanıcı gereksinimleri birçok farklı gösterimle ifade edilebileceği için ölçümcünün soyutlama yaklaşımı ve yorumlama biçimine bağlı farklı ölçüm sonuçları elde edilebilir. İlk kategorideki çalışmalarda formal ifade edilen yorumlamalar, bir spesifikasyon dili için tekrarlanabilir ve tutarlı ölçümleri olanaklı kılarak soruna kısmi çözüm sağlamışlardır. Yorumlamaların öznel olması, yine bir formal gösterimle ifade edilebilen ve tutarlı sonuçlar üreten başka bir yorum ile farklı sonuçlar

bulunması sorunu beraberinde taşımaktadır. Bu nedenden dolayı ilk kategori çalışmaları bağlamında öznel ölçüm problemi geçerli kalmaktadır.

İkinci kategori yayınların temel uğraşı İBÖ yöntemlerinin soyut yazılım modellerinin, model öğelerinin, öğeler arası ilişkilerin ve sayısal değer atama fonksiyonlarının formel gösterimi ile yöntemlerin ve yöntemle ilgili sorunların daha açık ve anlaşılır kılınmasıdır. Ölçüm sürecinde “boşluk” problemi devam etmekte, öznel yorumlar yapılarak yazılım tanımlama öğeleri genelleştirilmiş model öğeleri ile tutarlı biçimde eşleştirilmektedir. Genelleştirilmiş model İBÖ kurallarını soyutlamamaktadır.

Çalışmada bu sonuçlar neticesinde, veri odaklı anlatımlarla açıklanan İBÖ yöntem kurallarının ve soyut model bileşenlerinin farklı yorumlara açıklığını azaltacak formal tanımlara kavuşturularak ve model güdümlü mimari nosyonlarından esinlenerek, model transformasyonuna açık hale getirilmesi bir gelecek çalışma konusu olarak belirlenmiştir. Böylelikle herhangi bir formal spesifikasyon dili ile yöntem eşleştirmeleri, sistematik olarak yapılabilir.

4.1.3 Varlık-ilişki kavramlarının IFPUG ve COSMIC işlevsel büyüklük ölçümü yöntem terminolojisine eşleştirilmesi

Yaygın İBÖ yöntemleri veri yönelimli soyut yazılım modelleri üzerinde tanımlanır[29]. Doğru ölçüm yapabilmek için işlevsel kullanıcı gereksinimlerinin, kriterleri İBÖ yöntemlerince belirlenmiş veri tipleri ve gruplamalarının belirlemek üzere analizi gereklidir.

Varlık-ilişki (Entity-Relationship) yöntemi yazılım uygulamalarının veri yapılarının modellenmesi amacıyla yaygın olarak kullanılan en önemli yöntemlerden biridir. Çalışmada 31, İBÖ yöntemleri kuralları kapsamında her bir yöntemin Varlık-ilişki kavramları ile nasıl eşleştirilebileceği incelenmiş ve bu ilişkiyi ifade edecek formel kurallar oluşturulmuştur. Varlık, ilişki, ilişki tipi ve yönü gibi varlık-ilişki kavramları, IFPUG ve COSMIC yöntemleri veri kavramları ile bir kavram haritasında ilişkilendirilmiştir (Tablo 2 ve Tablo 3). Böylelikle İKG de anlatılan kullanıcı verisinin, varlık-ilişki modeli cinsinden ifadesiyle, İBÖ yöntemleri özelinde ölçülen işlevsel bileşenler eşleştirme haritası kullanılarak belirlenebilecektir. Bu ölçümlere bir formalite katacağı gibi, İBÖ yöntemleri arasında dönüşümlerin sayılardan öte kavramsal olarak yapılabilmesine olanak sağlamaktadır.

Tablo 2-Varlık Tipleri ve İBÖ Kavramları

| ER Terimi | IFPUG Terimi | COSMIC Terimi |
|----------------------------|--|---------------|
| Varlık (Tip) | Mantıksal Dosya(LF) | OOI |
| İlişkilendirme Varlık Tipi | Tercihli yada zorunlu altgrup. Olası LF yada RET | OOI |
| Özneliksel Varlık Tipi | Tercihli yada zorunlu altgrup. Olası LF yada RET | OOI |
| Varlık Altıtipi | Tercihli yada zorunlu altgrup. Olası LF yada RET | OOI |

Tablo 3-İlişki Tipleri, IFPUG ve COSMIC Kavramları

| | A:B | A | B | Bağımlılık | #LF | #RET | #OOI |
|---|-----------|------|------|---------------------|-----|------|------|
| 1 | (1) : (N) | 0..1 | 0..* | A,B bağımsızdır | 2 | 2 | 2 |
| 2 | 1 : N | 1 | 1..* | B,A ya bağımlıdır. | 1 | 2 | 2 |
| 3 | 1 : N | 1 | 1..* | B A dan bağımsızdır | 2 | 2 | 2 |
| 4 | 1 : (N) | 1 | 0..* | B,A ya bağımlıdır. | 1 | 2 | 2 |
| 5 | 1 : (N) | 1 | 0..* | B A dan bağımsızdır | 2 | 2 | 2 |
| 6 | (1) : N | 0..1 | 1..* | A,B den bağımsızdır | 1 | 2 | 2 |

| | | | | | | | |
|----|-------------|--------------------|------|---------------------|---|-------------|----------------|
| 7 | (1) : N | 0..1 | 1..* | A,B den bağımsızdır | 2 | 2 | 2 |
| 8 | (1) : (1) | 0..1 | 0..1 | A,B bağımsızdır | 2 | 2 | 2 |
| 9 | 1 : 1 | 1 | 1 | A,B bağımlıdır | 1 | 1 | 1 |
| 10 | 1 : (1) | 1 | 0..1 | B,A ya bağımlıdır. | 1 | 1 yada 2 | 1 |
| 11 | 1 : (1) | 1 | 0..1 | B A dan bağımsızdır | 2 | 2 | 2 |
| 12 | (N) : (M) | 0..* | 0..* | A,B bağımsızdır | 2 | 2 | 2 |
| 13 | N : M | 1..* | 1..* | B,A ya bağımlıdır. | 1 | 2 | 2 |
| 14 | N : M | 1..* | 1..* | B,A dan bağımsızdır | 2 | 2 | 2 |
| 15 | N : (M) | 1..* | 0..* | B,A ya bağımlıdır. | 1 | 2 | 2 |
| 16 | N : (M) | 1..* | 0..* | B,A dan bağımsızdır | 2 | 2 | 2 |
| 17 | B bir A'dır | B A dan miras alır | - | | 1 | 2 | 1 & 1 alt tipi |

4.1.4 Bütünleşik yazılım kestirim yaklaşımı geçerlemesi

Daha önce gerçekleştirilen yöntemler arası kavramsal eşleştirme ve veri-varlık terminolojisi eşleştirmesi ile geliştirilen yaklaşım durum çalışmaları ile geçerlenmiştir. Durum çalışmaları kapsamında; beş ayrı kurumda geliştirilen yedi ayrı yazılım uygulamasının işlevsel büyüklükleri ilk aşamada IFPUG ve COSMIC İBÖ yöntemleri ile ikinci aşamada bütünleşik yöntem ile ölçülmüş; ilk aşama sonuçları, bütünleşik yöntemin sunduğu IFPUG ve COSMIC değerleri ile karşılaştırılmıştır [84], [85], [86], [87]. Ölçümler, yöntemler hakkında ilgili eğitimlerin verildiği araştırmacılar tarafından gerçekleştirilmiş, ölçme süreç ve ürünleri konu uzmanları tarafından gözden geçirilmiştir. Durum çalışmaları sonuçları Tablo 4'de sunulmuştur:

Tablo 4. Geçerleme Durum Çalışmaları Sonuçları

| Durum Çalışması | IFPUG Fonksiyon Nokta | | | COSMIC Fonksiyon Nokta | | |
|---|-----------------------|------------|------|------------------------|------------|------|
| | Manüel | Bütünleşik | Fark | Manüel | Bütünleşik | Fark |
| Web-tabanlı Finans Yönetim Uygulaması | 287 | 294 | 7 | 211 | 211 | 0 |
| İnsan Kaynakları Yöntemi Uygulaması – Destek Modülü | 423 | 423 | 0 | 371 | 371 | 0 |
| Abone Bilgi Yönetim Sistemi Uygulaması | 345 | 345 | 0 | 274 | 274 | 0 |
| Finans Yönetimi Uygulaması – Bütçe Modülü | 212 | 212 | 0 | 175 | 175 | 0 |
| İnternet Bankacılığı Uygulaması | 331 | 313 | 18 | 379 | 379 | 0 |
| Vergi Borç Sistemi | 205 | 205 | 0 | 208 | 208 | 0 |
| Kredi Kartı Vergi Tahsilat Uygulaması | 106 | 106 | 0 | 135 | 135 | 0 |

Tablo 4'de de görüleceği gibi, durum çalışmalarının ilk aşamasında manüel olarak gerçekleştirilen IFPUG ve COSMIC ölçümleri ile ikinci aşamada bütünleşik model ile yapılan ölçümler birbirlerine oldukça yakın değerler vermektedir. Bu başarılı sonuçların yanında, iki projede IFPUG yöntemi sonuçlarında ortaya çıkan farklar ile ilgili durum değerlendirmeleri yapılmış, yaklaşım ve araç ile ilgili iyileştirilmesi gereken noktalar tespit edilmiştir.

4.2 İBÖ İşlevsel Büyüklük ve İşgücü Arasındaki İlişki

4.2.1 İşlevsel benzerlikler ve işgücü arasındaki ilişki üzerine yapılan çalışma

Yazılımın geliştirilmesi için gereken iş gücü ve maliyetinin belirlenmesi için kullanılan yöntemlerde yazılımın işlevsel büyüklüğü birincil girdi olmasına rağmen, pratikte işlevsel büyüklük ile yazılımın geliştirilmesi için gereken iş gücü arasındaki ilişki net olarak kurulamamaktadır. İlişkinin kesin olarak

kurulamamasının sebeplerinden birinin yazılım içerisindeki işlevsel benzerlikler olduğunu düşünülmektedir. İşlevsel benzerliklerin büyüklüklerinin hesaplanması ile bu sorunun kısmi olarak çözümlenebileceği öngörülmüştür. Bu çalışmada [89] işlevsel benzerliklerin hesaplanması için beş ayrı projeye farklı yaklaşımlar uygulanmıştır. Bu yaklaşımların uygulanabilirliği ise işlevsel benzerliklerin yazılım geliştirme iş güçleri ve kod satır sayıları üzerindeki etkileri gözlemlenerek değerlendirilmiştir. Durum çalışmaları sırasında karşılaşılan güçlükler ve ortaya çıkan fırsatlar tartışılmıştır.

Santillo ve Abran'ın[88] yönteminin değerlendirmesi sırasında karşılaşılan zorluklardan biri yöntemde belirtilen işlevsel benzerlik belirleme tekniklerinin hata yapmaya çok açık ve zaman gerektirici olmasıdır. Bu nedenle yöntem, özellikle büyük yazılım projeleri için uygun değildir. Yukarıda belirtilen sorunların üstesinden gelebilmek için değerlendirme sürecini kısmen de olsa otomatikleştiren bir program geliştirilmiştir. Bu program, metodun her iki aşaması için de uygulanabilmekte, hata olasılığını ve değerlendirme süreci için gereken iş gücünü önemli ölçüde azaltmaktadır.

Kavramsal Modelleme Aracı(KAMA) yazılım projesine ait örnek uygulamada işlevsel süreçler arasındaki benzerliklerin uygulamadan beklenen benzerlik oranını yansıtmamasından dolayı işlevsel süreçleri alt birimler altında gruplayarak ikinci bir benzerlik belirleme çalışması yapılmıştır. Bu iki yaklaşımın sonuçları arasındaki ciddi farklılık, analizi yapan kişinin yorumuna bağlı olarak işlevsel benzerlik oranlarının büyük farklılıklar gösterebileceğini ortaya koymuştur.

Abran ve Santillo tarafından belirlenen yöntemin uygulanmasında karşılaşılan bir başka problem, ikinci aşama için tanımlanan 'veri işlemlerinin' (data manipulations) nasıl sınıflandırılacağı ve bu verilerin kullanıcı gereksinimlerinden nasıl çıkartılacağına COSMIC kılavuzunda detaylı olarak anlatılmamış olmasıdır. "Veri değerlendirmeleri" terimi çok genel olarak kullanılmakta ve bu kapsamda herhangi bir hesaplama içeren iki süreç benzer kabul edilebilmektedir. Oysa hesaplamaların ardındaki algoritma farklılıkları yazılımın büyüklüğünü önemli ölçüde etkilemektedir.

Benzer süreçleri ayıklama amacı ile uyguladığımız diğer yöntem olan "Varlık Genelleştirme" yöntemi ilk ölçümde elde ettiğimiz 964 CFP lik büyüklük yerine 97 CFP lik bir büyüklüğe ulaşmamıza neden olmuştur. Bunun sonucu olarak varlık genelleştirme yönteminin iş gücü ve işlevsel büyüklük arasındaki ilişkinin kurulmasına katkıda bulunacağı öngörülmektedir.

Bu çalışmada ayrıca tekrar kullanımın uygulamalardaki kaynak kod satır sayısı (SLOC) miktarını nasıl etkilediğini de incelenmiştir. İşlevsel benzerlikleri göz önünde bulundurarak ulaştığımız işlevsel büyüklükleri, uygulamaların SLOC değerleri ile karşılaştırdık. İşlevsel benzerlik oranı arttıkça uygulamalardaki satır sayısının azalmakta olduğunu gördük. Fakat "ortalama benzerlik yüzdesi ile SLOC" arasındaki oranın aynı oranda azalmadığını da gözlemledik.

İncelenen önemli konulardan biri de COSMIC ile ölçtüğümüz yazılım büyüklüğünün ürün geliştirme için harcanan iş gücü ile olan ilişkisi idi. Aynı yazılım ekibinin geliştirdiği projeler üzerinde yaptığımız bu değerlendirmede işlevsel benzerlikleri dikkate almadan ulaştığımız verimlilik değerlerinin birbirinden oldukça uzak olduğunu gözlemledik. Bir başka deyişle yeni bir projede aynı ekibin verimlilik değerini kullanarak iş gücü tahmini yapabilmek mümkün değildi. Benzerlikleri dikkate alarak yazılım büyüklüğünü yeniden hesapladığımızda ise verimlilik değerlerinin birbirine nasıl yakınsadığını ve iyileştirdiğini gördük. Bu çalışma bizi; yazılım büyüklüğü ve harcanacak iş gücü arasında bir bağlantı olduğuna, fakat ilişkinin ancak işlevsel benzerlikler dikkate alındığı zaman daha doğru kurulabileceği sonucuna götürmüştür.

Çalışma devamında, proje araştırmacılarından Özden Özcan, proje yürütücüsü Doç. Dr. Onur Demirörs danışmanlığında hazırladığı tez çalışmasında [90],benzer fonksiyonların aynı ürün içerisinde tekrar kullanılma fikrinden yola çıkarak işlevsel büyüklük ile işgücü arasındaki ilişkinin yüksek olmasını sağlayacak bir yöntem geliştirmiştir(SiRFuS). Yöntem, yazılım ürünü içerisindeki işlevsel benzerliklerin ve benzerlik yüzdelerinin belirlenmesi ve Benzerlik Etkisindeki İşlevsel Büyüklüklerin hesaplanarak yazılım işgücü ve işlevsel büyüklük arasındaki ilişkiyi güçlendirecek uyarlanmış bir büyüklük elde edilmesi esasına dayanmaktadır. Benzerlik etkisindeki işlevsel büyüklükler, uyarılma yaklaşımının özelliklerine göre "Ayrıık Benzerlik Etkili İşlevsel Büyüklük" ve "Devamlı Etkili İşlevsel Büyüklük" olarak adlandırılmıştır.

SiRFuS yöntemi üç aşamadan oluşmaktadır:

- (1) Yazılım ürününün COSMIC İşlevsel Büyüklük Ölçüm (İBÖ) yöntemi ile belirlenmesi,
- (2) Ölçüm sonuçlarından işlevsel benzerliklerin elde edilmesi ve
- (3) Benzerlik etkisindeki işlevsel büyüklüklerin hesaplanması.

Benzer fonksiyonların bulunarak işlevsel süreçler arasındaki benzerliklerin yüzdelerinin belirlenmesi ve benzerlik etkisindeki uyarlanmış işlevsel büyüklüklerin bulunmasını kolaylaştırmak için bir araç geliştirilmiştir (SR Matlab Tool). Gelişim fırsatlarını belirleyebilmek ve yöntemin ve aracın uygulanabilirliğini değerlendirebilmek için iki durum çalışması yapılmıştır. Durum çalışmaları sonucunda, geliştirilen yöntem (SiRFuS) ile elde edilen üretkenlik değerleri arasındaki sapmanın azaldığı gözlenmiştir. Ayrıca SiRFuS ile elde edilen üretkenlik değerleri Capers Jones (1998) tarafından yayımlanan üretkenlik değerleri aralığında yer aldığı için yöntem ile elde edilen verilerin geçerlendiğini de gözlemlenmiştir.

Sonuç olarak, iş gücü kestirimi için SiRFuS ile elde edilen uyarlanmış işlevsel büyüklük değerleri kullanıldığı takdirde gerekli işgücü, daha başarılı olarak tahmin edilebilecektir. Tez çalışmasının devamı olarak sürdürülen çalışmada [91], temel olarak benzer fonksiyonların aynı ürün içerisinde tekrar kullanılma fikrinden yola çıkarak işlevsel büyüklük ile işgücü arasındaki ilişkinin yüksek olmasını sağlayacak SiRFuS yöntemi ve yöntemi doğrulamak için gerçekleştirilen tekli ve çoklu durum çalışmaları anlatılmaktadır. Yöntem, yazılım ürünü içerisindeki işlevsel benzerliklerin ve benzerlik yüzdelerinin belirlenmesi ve Benzerlik Etkisindeki İşlevsel Büyüklüklerin hesaplanarak yazılım işgücü ve işlevsel büyüklük arasındaki ilişkiyi güçlendirecek uyarlanmış bir büyüklük elde edilmesi esasına dayanmaktadır. Durum çalışmaları, SiRFuS ile elde edilen uyarlanmış işlevsel büyüklük değerlerinin işgücü kestiriminde daha gerçekçi sonuçlar elde edilmesine olanak sağladığını göstermiştir.

Yapılan diğer bir çalışma [92] kapsamında bugüne dek geliştirilen işlevsel benzerlik yöntemleri araştırıldı. Bu yöntemlerin genellikle küçük ölçekli işletme yönetim sistemleri alanında geliştirilen yazılımlarda geçerlendiği ve bu geçerlemede iş gücü ile büyüklük arasındaki ilişkinin ele alınmadığı değerlendirildi. Bu eksikleri gidermek üzere bu yöntemler aynı yazılım organizasyonunda farklı uygulama alanlarında geliştirilen 18 yazılım sistemi üzerinde uygulandı. Bu uygulamaların sonucu olarak, geliştirilen yöntemlerin kullanım zorlukları, hangi alanlarda daha etkin olarak kullanılabileceği ve iş gücü kestiriminde nasıl kullanılabileceği yönünde bulgular elde edildi.

4.2.2 İşlevsel Büyüklük Bazlı Regresyon Modellerinin karşılaştırılması

Bu çalışmada, işgücü tahminleri için literatürde önerilen İşlevsel Büyüklük bazlı Regresyon Modelleri incelenmiştir. Bu modeller aynı organizasyonda geliştirilen 17 adet tamamlanmış proje üzerinde denenmiştir. Projelerin İşlevsel büyüklük ölçümleri Cosmic 3.0 kullanılarak ölçülmüştür. İşgücü değerleri hesaplanırken ilgili modellerin önerisinde kullanılan IFPUG vb diğer ölçüm yöntemleri arasında tutarlılık sağlamak üzere çevrim işlemleri yapılmıştır. Modellerin ürettiği işgücü değerleri ile gerçek işgücü değerleri arasında karşılaştırma yapılmıştır. Ayrıca mevcut data kullanılarak linear ve nonlinear 5 ayrı model baz alınarak, yeni regresyon modelleri de oluşturulmuştur. Toplam 12 model üzerinde yapılan incelemenin doğruluğu “ortalama hata”, “bağımsız parametrenin modeli tanımlama yetisi”, “tüm veri seti üzerinde tahminlerin gerçek işgücüne yakınlığı” parametreleri ile incelenmiştir. Regresyon modelleri arasında en iyi sonucu veri seti üzerinden oluşturulan Linear model vermiştir. Fakat bu modelin bile doğruluk değerleri yeterli değildir. Sadece yazılımın işlevsel büyüklüğünü kullanan regresyon modelleri ile işgücü değerini gerçekçi tahmin etmek mümkün gözükmemektedir. Standart Regresyon modeli yerine başka modeller oluşturulması yönünde çalışmalara başlanmıştır.

Ayrıca işlevsel benzerliğin, mevcut veri seti ile oluşturulan regresyon modelleri üzerindeki etkisi incelenmiştir. İşlevsel büyüklük ölçümünün ortalama hata değerini düşürdüğü ve bağımsız parametrenin modeli tanımlama yetkililiğini artırdığı bulgusuna erişilmiştir.

4.2.3 İşlevsel bileşenler temelinde sinirsel ağ ve regresyon modeli yaklaşımları karşılaştırması

Bu çalışma [93] regresyon modelleri üzerinde yapılan önceki çalışmanın devamı olarak aynı data set üzerinde gerçekleştirilmiştir. Çalışmada Yazılımın işlevsel büyüklüğü alt parçalarının işgücü tahmini

üzerindeki etkisi incelenmiştir. İşgücü modeli oluşturmak için sinir ağları ve çok girdili regresyon modelleri kullanılmıştır. Yazılım alt parçalarında COSMIC ölçüm yöntemi kullanılmakla birlikte, yöntemin IFPUG gibi başka işlevsel ölçüm yöntemlerinde de kullanılması mümkündür. Sinir ağları için işgücü modelinde yazılım büyüklük ölçümünün alt parçalarının kullanımı literatürde önerilmediğinden bu çalışmada elde edilen sonuçlar ilk olması bakımından önemlidir. Bu çalışmada Uygulama tipinin etkisi de incelenmiş, 3 ayrı uygulama tipinin (Grafiksel kullanıcı Arayüzü, Gömülü Sistemler, Donanım sürücü yazılımları) hem sinir ağları hem de çok girdili regresyon modellerinde ne kadar iyileştirme sağladığı gözlenmiştir. İşlevsel benzerlik kavramı önceki çalışmaların bir devamı olarak yeni oluşturulan bu modeller üzerinde de incelenmiştir. Oluşturulan modeller ile yapılan tahminlerin sonuçları önceki çalışmada elde edilen sonuçlar ve birbiri ile karşılaştırılmıştır. Buna göre; alt parçaların kullanımı her iki yöntemde de tüm işlevsel büyüklüğü kullanmaktan daha iyi sonuçlar vermektedir. En iyi sonuç sinir ağlarında elde edilmektedir. Ayrıca İşlevsel benzerliğin hesaplara katılması sinir ağlarında doğru modelin oluşturulmasını hızlandırmakta, model oluşturulurken gereken döngü sayısını düşürmektedir. Uygulama tipinin etkisi, işgücü üzerinde işlevsel alt parçalar kadar yüksektir ve tahmin doğruluğunu hem sinir ağlarında hem regresyon modellerinde arttırmaktadır.

4.3 Ölçüm Sonuçlarında Bireyler Arasındaki Sonuçlardaki Farklılıklar

Yazılım maliyet, zaman kestirimlerinde, kullanılan yazılım büyüklüğü ölçümünü yönteminin doğru öğrenilmesi ve uygulanması, kestirimin başarısı ile direkt ilgilidir. Literatürde de rastlandığı üzere işlevsel büyüklük ölçüm yöntemlerinin eleştirilen problemlerinden biride ölçüm yöntemi olmanın gereği olan nesneliliği (İng.objectivity) yeterince gösterememesi ve öznel (İng.subjective) sonuçlara açık olmasıdır. Bütünleşik yöntemin geliştirilmesi kapsamında yapılan çalışmalarda gerçekleştirilen büyüklük ölçümlerinde de, yöntemlerin öğrenilmesi ve uygulanması sırasında kişilerin bilgi seviyesi, kullanıcı gereksinimlerini yorumlama biçimleri ve yöntemin öğrenildiği kaynaklara göre ölçümlerin kalitesi ve sonuçlarında farklılıklar gözlemlenmiştir. Bu bölümde bu probleme yönelik çalışmalar ve bulgular açıklanmıştır.

4.3.1 Varlık genelleştirmenin işlevsel büyüklük ölçümüne etkileri

Bu çalışmada [94], yazılım geliştirme projelerinde sıkça uygulanan varlık genelleştirme (veya özelleştirme) pratiklerinin, yaygın kullanılan İBÖ yöntemleri kapsamında ürünün işlevsel büyüklüğünün hesaplanmasına olan etkileri incelenmiştir. Bu etkinin her bir yaygın model için ortaya konması amacıyla, varlık genelleştirmenin büyük ölçekte kullanıldığı bir yazılım aracının (KAMA-Kavramsal Modelleme Aracı) işlevsel büyüklüğü IFPUG FPA ve COSMIC FFP yöntemleri ile ölçülmüş ve ölçümede ortaya çıkan farklılıklar ve karşılaşılan zorluklar gözlemlenmiştir. Buna göre, varlık genelleştirme, incelenen yöntemlerde farklı şekillerde ele alınmakta ve yazılımın işlevsel büyüklüğüne farklı ölçülerde etki etmektedirler. Örnek olay incelemesi, varlık genelleştirmenin, her yöntem için aynı olması beklenen temel işlevsel bileşenlerinin (base functional component - BFC) farklı şekillerde tanımlanmasına sebep olabileceğini göstermiştir. Dolayısıyla, bu yöntemlerle ölçülmüş değerlerdeki farklılıkların, yöntemlerin ölçümede uyguladıkları prosedür farklılıklarının yanı sıra, bundan daha esas olarak, temel yapı taşlarının farklı şekilde tanımlanması sebebine dayanacağı sonucunu doğurmuştur. Bu durum, ölçüm yapan farklı bireylerin varlık genelleştirmeye dair farklı kabullenmelerinin yada yaklaşımlarının, ölçüm sonuçlarında farklılığa yol açacağı ve yöntemlerin bu tip farklılıklara konuda açık olduğunu göstermiştir. Dolayısıyla bu çalışma, bu yöntemleri biçimsel ve tanımlı bir yapıda birleştiren bir yöntemin geliştirilmesi gerekliliğini bir kez daha göstermiştir.

4.3.2 Bireysel yaklaşımların yazılım işlevsel büyüklük ölçümü sonuçlarına etkisi

Bu çalışmada [94], bireylerin farklı yaklaşımlarının ve varsayımlarının ölçüm sonuçlarını önemli ölçüde etkileyecek farklılıklara sebep olup olmadığını gözlemlenmek amacıyla iyi bilenen ve sıklıkla kullanılan İBÖ yöntemlerinin kullanıldığı çoklu durum çalışmaları gerçekleştirilmiştir.

Bu amaçla, aynı yazılım ürününün büyüklüğünün bağımsız olarak ölçülebilmesi için büyüklük ölçüm konusunda uzman kişilerden oluşan 3 ayrı takım oluşturmuştur. İkişer kişiden oluşan takımlar aynı yazılım ürününe sırasıyla COSMIC, MkII FNA (fonksiyon nokta analiz), ve IFPUG FNA yöntemlerini uygulamışlardır. Çalışmada, sıklıkla kullanılan bu üç ölçüm yöntemi farklı bakış açılarıyla değerlendirilmiş; ölçüm süreciyle ilgili olan problemler ortaya konmuştur.

Durum çalışması sonuçlarına göre, yazılım ürününün (KAMA – Kavramsal Modelleme Aracı) büyüklüğü, IFPUG FNA yöntemi ile ölçüldüğünde 306; MkII FNA yöntemi ile ölçüldüğünde 3,861.36 ve COSMIC ile ölçüldüğünde 697 çıkmıştır. Bu yöntemlerin ölçüm birimleri aynı olmamakla birlikte elde ettiğimiz sonuçlar mevcut dönüşüm formüllerinin ortaya koyduğu sonuçların çok ötesindedir.

Ölçümler, her takım için aynı 'yazılım gereksinimleri dokümanını temel almıştır. Buna rağmen, üç yöntem için temel işlevsel bileşenlerin sayısı IFPUG için 45, MkII için 283, COSMIC için 55 olarak bulunmuştur. İBÖ yöntemlerinin temel işlevsel bileşenler ile ilgili tanımları birbirinin aynısı olmadığı için bu sayıların da birebir aynı olması beklenemez, fakat bu düzeydeki farklılıkları da yöntemlerin ölçüm farklılığı olarak açıklamak mümkün değildir.

Bu çalışmanın sonuçları göstermiştir ki, KAMA gibi geleneksel olmayan yazılım ürünlerinin ölçüm sırasında farklı yaklaşımlara ve varsayımlara maruz kalması, yöntemlerin mevcut tanımlamaları sebebiyle çok olasıdır. Durum çalışmasına katılan takımların, aynı ürünün 'temel işlevsel bileşenlerini' ve 'temel işlevsel bileşen tiplerini' belirlerken farklı varsayımlarda bulunmaları ve farklı çıkarımlar yapmaları sebebiyle birbirinden farklı sonuçlara ulaşmışlardır. Bu nedenle işlevsel büyüklük ölçüm kılavuzları ölçüm sürecinde varsayım ve çıkarım yapmayı engelleyecek ölçüde ölçüm bileşenlerinin tanımlarını detaylandırmalıdır.

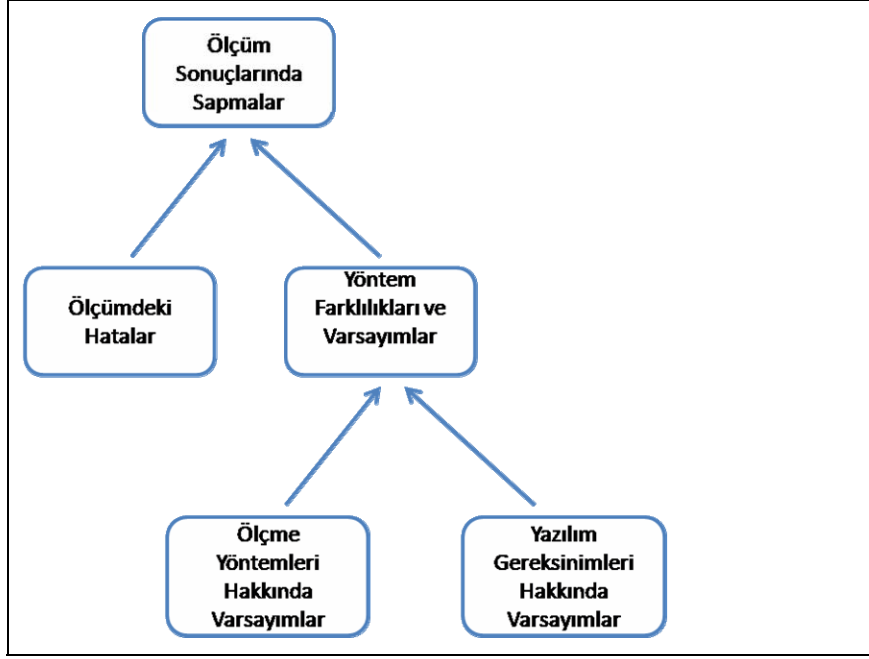
4.3.3 Kişisel varsayımların COSMIC İBÖ yönteminin uygulanmasında etkilerini belirlenmesi kontrollü deneyi

Kişilerin bilgi seviyesi, kullanıcı gereksinimlerini yorumlama biçimleri ve yöntemin öğrenildiği kaynaklara göre ölçümlerin kalitesi ve sonuçlarında farklılıklar gözlemlenmiştir. Bu gözlemleri kontrollü ve karşılaştırmalı bir ortam ve süreçte tekrarlamak amacıyla, ODTÜ Enformatik Enstitüsü bünyesinde konu ile ilgili ders alan öğrenciler üzerinde bir deney çalışması yürütülmüştür [96]. Katılımcıların ölçüm sonuçları analiz edilerek, işlevsel büyüklük ölçümü (İBÖ) sırasında sıklıkla karşılaşılan hatalar ve ölçüm sonuçlarında farklılıklara neden olan etmenler araştırılmıştır. Tespit edilen etmenlerin altında yatan temel sorunlar araştırılmış ve bu sorunlar için bazı öneriler ileri sürülmüştür.

Çalışmada COSMIC İBÖ yönteminin kullanılmasının nedenleri; yöntemin uluslararası bir ölçüm standardı olarak kabul edilmiş olması, birçok organizasyon tarafından büyüklük ölçüm yöntemi olarak kullanılıyor olması ve COSMIC için literatürde benzer bir çalışmanın yapılmamış olması şeklinde özetlenebilir. Çalışmanın benzer amaçlarla gerçekleştirilen diğer çalışmadan farkı, endüstri bünyesinde gerçekleştirilmiş bir durum çalışması yerine kontrollü bir ortamda yürütülmüş bir deney olmasıdır. Aynı zamanda bilindiği kadarıyla bu çalışmalar bu konuda görece yeni bir İBÖ yöntemi olan COSMIC ile gerçekleştirilmiş ilk çalışmalardır.

Çalışma kapsamında, sırasıyla, katılımcılara İBÖ ve COSMIC üzerine eğitimler verilmiştir. Daha sonra, eğitimin başarısını doğrulamak ve katılımcıların deneyim kazanmalarını sağlamak amacıyla bir pilot proje üzerinde benzeri bir ölçüm çalışması yapılmıştır. Bu çalışma sonunda katılımcıların ölçümleri üzerinde tartışılmış ve gerekli geri dönüşler yapılmıştır. Daha sonra katılımcılara, özel olarak hazırlanmış bir proje tanımı, gereksinimler ve varlık ilişki diyagramı verilmiş, bu bilgilerle gereksinimlerin ölçülmesi istenmiştir.

Ölçüm sonuçları, konunun uzmanlarınca önceden hazırlanmış bir çözüm anahtarıyla karşılaştırılarak detaylı bir şekilde değerlendirilmiştir. Bu değerlendirmeler sırasında gözlenen hatalar, ortak noktalarına göre gruplanmıştır. Son olarak, tespit edilen bu noktalarda soruna neden olan temel nedenler tartışılmış ve bu alanlarda gerçekleştirilebilecek bazı iyileştirmeler öne sürülmüştür. Çalışmanın sonuçları yukarıda bahsedildiği gibi üç aşamada sunulmuştur. İlk aşamada, ölçümlerin çözüm anahtarı ile karşılaştırılmasıyla elde edilen, yalın istatistiksel veriler sunulmuştur. Çalışma sonunda varılan sonuca göre; farklı kişilerce gerçekleştirilen ölçüm sonuçları arasında gözlenen farklılıkların, iki temel boyutu bulunmaktadır. İlki, ölçüm yönteminin kurallarının uygulanmasındaki hatalar, ikincisi ise, bu kurallar doğru uygulanırsa bile kişilerin kendi yorum ve varsayımlarından kaynaklanan farklardır(Şekil 3).



Şekil 3-Ölçüm sonuçlarında sapmaya yol açan etkenler

Bu çalışmada, tek bir yazılım problemi, kontrollü bir ortamda, farklı kişilerce, COSMIC İBÖ yöntemi kullanılarak ölçülmüştür. Ölçüm sonuçlarında farklılıklara neden olan unsurlar, farklı projelerin ölçülmesinden kaynaklanabilecek veri kirliliğinden etkilenmeden tespit edilmiştir. Çalışma sonucunda, yöntemin kendisinde ve uygulanmasında bazı iyileştirmeler önerilmiş ve bu iyileştirmelerle bahsedilen unsurlardan kaynaklanan sapmaların önüne geçmenin mümkün olduğu ileri sürülmüştür.

4.3.4 COSMIC işlevsel büyüklük ölçüm sonuçlarının güvenilirliği durum çalışması

Bu çalışmanın [97] amacı COSMIC işlevsel büyüklük ölçümü sırasında sıklıkla yapılan hataları, bu hataların nedenlerini ortaya koyarak ölçüm sonuçlarının doğruluğunun artırılmasına katkıda bulunmaktır. Bunlara ek olarak pilot çalışmaların ölçüm sonuçlarının doğruluğuna olan etkisi de çoklu durum çalışması kapsamında araştırılmıştır.

Durum çalışmasında sırasıyla pilot proje (Film Yönetim Aracı) ve 12 farklı endüstri projesi 5 farklı katılımcı tarafından COSMIC v3.0 kullanılarak ölçülmüştür. Sinema Yönetim Aracının gereksinimleri, ölçüm sürecinin temel ve sorunlara açık yönlerini ortaya koyacak şekilde tasarlanmıştır. Çalışmada COSMIC İBÖ yönteminin kullanılmasının nedeni, yöntemin ISO tarafından uluslararası bir ölçüm standart olarak kabul edilmiş olması, birçok organizasyon tarafından büyüklük ölçüm yöntemi olarak kullanılıyor olması ve COSMIC için literatürde benzer bir çalışmanın yapılmamış olmasıdır. Çalışmada kullanılan örnek olaylar ölçümü yapan katılımcılar bazında gruplandırılarak, sonuçların doğruluk ve hata oranları hesaplanmış, sıklıkla yapılan hatalar belirlenerek bu hataların nedenleri araştırılmıştır.

Pilot çalışmanın sonuçları incelendiğinde katılımcıların COSMIC işlevsel büyüklük ölçümü ile ilgili çok temel kavramlarda hatalar yaptığı gözlemlenmiştir. Pilot proje ölçümlerinin ortalama Doğruluk ve Hata oranları sırasıyla %21.99 ve % 219.61 olarak hesaplanmıştır. Endüstri projelerinin Doğruluk ve Hata oranları ise %78.68 ve %22.82 olarak hesaplanmıştır. Endüstri projelerinin doğruluk oranındaki bu gözle görülür artış başarılı bir yazılım yönetimi için yeterli değildir. Bu nedenle, yazılımın büyüklüğünün güvenilirliğini artırmak amacıyla bazı önleyici ve doğrulayıcı önlemler alınmalıdır.

İşlevsel büyüklük ölçümünün güvenilirliği önemli ölçüde ölçümü gerçekleştirecek kişilerin bilgi ve tecrübe seviyelerine bağlı olarak değişmektedir. Bu nedenle ölçüm yapmadan önce, ölçülecek yazılımın yapısına uygun olarak tasarlanmış pilot uygulamalar üzerinde çalışarak; hem ölçüm sürecine hem de yazılımın uygulama alanından kaynaklanan özgün durumlara aşina olmak, ölçüm sonuçlarının güvenilirliğini artıracaktır.

Özellikle iş uygulama yazılımları için, COSMIC İBÖ eğitimleri sırasında detaylı olarak aşağıda yer alan makale ve teknik raporda anlatılmış olan hata tiplerinin üzerinde durulması ve ölçüm sonuçlarını doğrularken bu hata tiplerine odaklanması işlevsel büyüklük ölçüm sonuçlarının güvenilirliğini artıracaktır. Buna ek olarak iş uygulamaları için COSMIC resmi kaynaklarında verilen kılavuz ve örnekler yöntemi daha anlaşılır kılmaktadır.

4.4 İşlevsel Büyüklük Ölçümü İçin Araç Desteği

Proje kapsamında, işlevsel büyüklük ölçümünü ve ölçüm sonuçlarının çeşitli proje yönetim süreçlerinde kullanımını destekleyecek bir otomasyon çözümüne yönelik araştırmalar bu kısımda anlatılmıştır.

4.4.1 Yazılım İşlevsel büyüklük ölçüm sürecini destekleyen araç incelemesi

Geliştirilen bütünleşik yazılım kestirimi için bir aracının sahip olması beklenen özelliklerinin belirlenmesine katkı sağlamak amacıyla, işlevsel büyüklük ölçümü sürecini ve büyüklüğün işgücü ve maliyet kestirimi gibi süreçlerde kullanımını destekleyen araçlar incelenmiştir [101]. Konu ile ilgili yapılan en önemli çalışmalar Mendes, Abran, Borque [98],[99] ve Stambolian, Abran tarafından gerçekleştirilmiştir. Mendes ve Abran, Borque tarafından yapılan çalışmalarda, 1996 yılında pazarda bulunan elliye aşkın araç incelenerek, araçların menşei, teknik ve işlevsel özellikleri ve desteklediği etkinlikler göz önüne alınarak, kullanım kategorileri oluşturulmuş ve mevcut araçlar bu kategorilere göre sınıflandırılmıştır. Daha sonra yapılan çalışmalarla bu kategoriler de gruplandırılmış, gruplar dört farklı boyut olarak yeniden tanımlanarak araçların değerlendirilmesine kolaylık sağlayacak İBÖ fonksiyon genel çatısı oluşturulmuştur. Stambolian, Abran'ın 2006 yılında yaptığı çalışmada COSMIC İBÖ yöntemi için geliştirilmiş araçlar bu çerçevede değerlendirilmiştir. Tablo 5-İBÖ araç kategorileri ve anahtar boyutları bu çalışmalar neticesinde oluşturulmuş kategori ve boyutları göstermektedir.

Tablo 5-İBÖ araç kategorileri ve anahtar boyutları

| Anahtar Boyutlar | Araç Kategorileri |
|------------------|---|
| A) Ölçüm Desteği | - Dokümantasyon - Eğitim |
| B) Ölçüm | - Veri toplama ve Hesaplama - Ölçüm için Uzman Sistem - Otomatik Ölçüm - Üst Seviye Büyüklük Ölçme |
| C) Saklama | - Yazılım Ölçüm Havuzu |
| D) Faydalanma | - Kestirim ve Tahminleme - Proje Yönetimi - Değerlendirme(kıyaslama, vb.) |

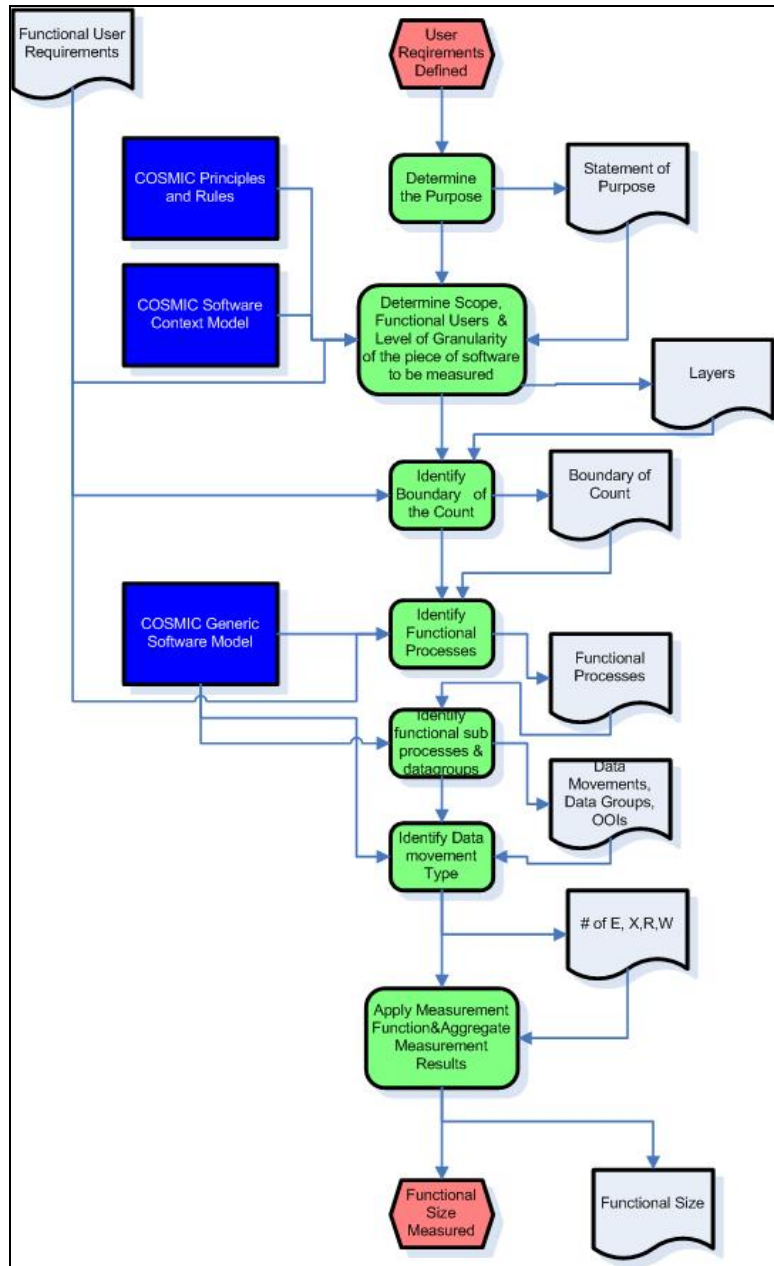
Yaptığımız çalışmada, farklı olarak araçlar İBÖ yöntemlerince belirlenmiş ölçüm süreci perspektifinden değerlendirilmiştir. Değerlendirme kapsamına kullanımı görece yaygın yada özellik kapsamı bakımından geniş olan aşağıdaki araçlar alınmıştır (Tablo 6).

Tablo 6- İncelenen İBÖ Araçları

| Araç | Erişim adresi |
|--------------|---|
| FP Workbench | http://www.charismatek.com/_public1/html/fpw_evaluation.htm |
| MeterIT | www.telmaco.com/pdmetco15e.pdf |

| | |
|--------------|---|
| EasyEstimate | http://www.ii.metu.edu.tr/~smrg/ |
| FP Modeler | http://www.functionpointmodeler.com/ |
| Siesta | http://metrieken.sogeti.nl/Home/index.jsp |
| EstimatorPal | http://www.chemuturi.com/estimatorpal.html |

Araştırmada IFPUG ve COSMIC yöntemleri için araçlar gruplanılmış, her bir yöntem için hazırlanan süreç şemalarından yola çıkarak ölçüm basamakları ve ölçüm girdilerin ve çıktıları için birer değerlendirme formu hazırlanmış, ürün özellikleri puanlandırılmıştır. Şekil 4 ve Tablo 7’de COSMIC İBÖ yöntemini ölçüm süreci ve COSMIC İBÖ yöntemini destekleyen araçlar için yapılan değerlendirme verilmiştir.



Şekil 4-COSMIC İBÖ yöntemi ölçüm prosedürü

Tablo 7-COSMIC İBÖ yöntemini destekleyen araçların değerlendirmesi

| COSMIC İBÖ yöntemini destekleyen araçların değerlendirilmesi | | | |
|--|--------|--------------|---------|
| Ölçüm adımı | SIESTA | EasyEstimate | MeterIT |
| Amaç belirle | L | P | P |
| Kapsamı, Fonsiyonel kullanıcıları, detay seviyesi belirle | P | P | P |
| Ölçüm sınırını belirle | N | P | P |
| İşlevsel Süreçleri belirle | L | L | L |
| İşlevsel alt –süreçleri ve veri gruplarını belirle | P | L | L |
| Veri hareket tiplerini belirle | L | L | L |
| Ölçüm fonksiyonunu uygula ve sonuçları topla | F | L | F |
| Ürünler | | | |
| Amaç ifadesi | AM, PS | AM, PS | AM, PS |
| Katmanlar | AM, PS | AN, PN | AN,PN |
| Ölçüm sınırı | AN,PN | AN,PN | AN,PN |
| İşlevsel süreçler | AM,PS | AM,PS | AM,PS |
| Veri hareketleri | AM,PS | AM,PS | AM,PS |
| Veri grupları | AN,PN | AM,PS | AM,PS |
| İlgi nesnelere | AN,PN | AN,PN | AM,M |
| E, X,R,W veri hareketi tipi sayıları | AG,PS | AG,PS | AG,PS |
| İşlevsel Büyüklük | AG,PS | AG,PS | AG,PS |

Tablo 8-Araç Değerlendirme puan tablosu

| Puan | Açıklama |
|------|-------------------------------------|
| F | Tamamen destekliyor |
| L | Çoğu destekleniyor |
| P | Kısmen destekleniyor |
| N | Araç Kapsamında desteklenmiyor. |
| PS | Ürün saklanıyor. |
| PN | Ürün saklanmıyor |
| AG | Ürün otomatik üretiliyor |
| AM | Ürün elle giriliyor. |
| AN | Ürün araç kapsamında ele alınmıyor. |

Çalışma sonucunda, var olan araçların odaklandığı ve desteklenmediği ürünler, adımlar ve özellikler saptanmış, geliştirilecek yeni araçlarda yöntemlerin daha iyi desteklenmesi ve proje yönetiminde kullanımının artırılması için potansiyel iyileştirmeler tespit edilmiştir.

4.4.2 Yazılım işlevsel büyüklüğü ölçüsünün yazılım proje yönetimi ve ilgili diğer alanlarda kullanımı

Bu çalışmada işlevsel büyüklük ölçümlerinin proje yönetimine hangi şekillerde katkı sağlayabileceği, araştırılmıştır [102]. Araştırmanın çıktıları, bir aracın bir ölçümü desteklemenin ötesinde ölçüm sonuçlarının kullanımı için ne tür beklentileri karşılama gerektirdiğini belirlemiştir. Bu nedenle, araştırma 'bütünleşik yazılım kestirimi aracının' sahip olması beklenen özelliklerinin belirlenmesine katkı sağlamıştır. Bu amaçla ilgili literatür incelemeleri yapılmış ve proje yönetimi ilgili en iyi uygulamaları içeren Proje Yönetimi Bilgi Kütüğü'nde yer alan proje yönetimi süreçlerinin üzerinden gidilerek işlevsel büyüklüğün bu alanlarda nasıl kullanılabileceği incelenmiştir. Çalışma, işlevsel büyüklüğün maliyet ve işgücü kestirimini yanında, birçok alanda kullanılabileceğini göstermiştir.

Çalışmada kullanılan Proje Yönetimi Bilgi Kütüğü (Project Management Body Of Knowledge - PMBOK) [103] belirli bir süreç kümesi ve bunlara ait bilgi alanlarını (BA) içeren uluslararası alanda kabul görmüş bir referans kaynaktır. Proje Yönetimi Bilgi Kütüğü'nde (PYBK) bilgi; başlatma, planlama, yürütme, kontrol ve izleme ve kapatma olmak üzere beş proje yönetim süreci grubu şeklinde organize edilmiştir. Süreçler dokuz bilgi alanında kategorize edilmiştir. Yapılan çalışmada işlevsel büyüklük ölçüleri kullanımının şu bilgi alanlarında açık katkıları olacağı öngörülmüştür:

- Proje entegrasyon yönetimi
- Proje kapsam yönetimi
- Proje zaman yönetimi
- Proje maliyet yönetimi
- Proje kalite yönetimi
- Proje risk yönetimi
- Proje tedarik yönetimi

PYBK her süreci, girdiler, çıktılar ve süreçte uygulanabilecek araç ve teknikler olmak üzere üç kısımda anlatır. Süreçler, buldukları bilgi alanlarındaki veya diğer bilgi alanlarındaki süreçler ile doğrudan veya dolaylı olarak etkileşim içindedir, dolayısıyla bir bilgi alanındaki katkılar diğer bilgi alanına da katkıda bulunmaktadır. Çalışmada odak, ISO uyumlu üç işlevsel büyüklük ölçümü (İBÖ) yönteminin (MkII FPA, IFPUG FPA, ve COSMIC İBÖ) yetenek ve özellikleri göz önüne alınarak bilgi alanlarına verilebilecek doğrudan katkılar üzerindedir.

Proje yöneticilerinin işlevsel büyüklüğün kullanımında İBÖ yöntemlerinin zorlukları ve kısıtları hakkında bilgi sahibi olmaları gerekir. İşlevsel büyüklüğün ve ilgili ölçme yönteminin kullanımında ve seçiminde, çeşitli yöntemlerin avantaj, dezavantaj ve kısıtları göz önüne alınmalıdır.

Bu çalışmada, işlevsel büyüklüğün, proje yönetimi süreçlerinin etkin bir şekilde işletilmesine ne şekilde katkı sağlayabileceği incelenmiştir. Maliyet kestirimini proje yönetimindeki kritik süreçlerden biri olması ve işlevsel büyüklüğün kestirim modellerinde girdi olması, İBÖ yöntemlerinin yanlış bir algılama ile kestirim yöntemi olarak tanınmasına sebep olmuştur. İBÖ araştırma ve pratiğinde odak kullanım noktası haline gelen maliyet kestirimini yanında işlevsel büyüklüğün proje yönetimi çerçevesinde kullanılabileceği olası diğer fırsat noktaları araştırılmıştır. Bu amaçla, Proje Yönetimi Bilgi Kütüğü'nde yer alan bilgi alanları, işlevsel büyüklüğün çeşitli kullanım alanları ile eşleştirilmiştir. Bu eşleştirmede de görüldüğü üzere, her bilgi alanı için işlevsel büyüklük farklı şekillerde kullanılabiliyor olsa da, işlevsel büyüklüğün, proje yönetimi ve kurumun diğer süreçleri çerçevesinde birçok genel ve etkin kullanım alanı vardır. Süreçleri dik olarak kesen bu genel kullanım alanları şunlardır:

- Maliyet, işgücü ve süre kestirimine girdi oluşturma
- Kapsam değişikliğini izlemeye destek sunma

- Proje erken aşamalarında büyüklük ölçümüne olanak sağlama
- Yazılım satın almalarında satın alma birimi olarak kullanma
- Performans ve kalite ölçülerinin normalleştirilmesi

Bu araştırma çalışmasında, yazılım endüstrisinde işlevsel büyüklüğün tanınması ve kullanımının, temelde büyüklüğün maliyet ve işgücü kestiriminde kullanılması ve İBÖ yöntemlerinin bu konudaki model ve tekniklere odaklanmasından kaynaklandığı görülmüştür. Çeşitli kısıt ve kaygılara rağmen, işlevsel büyüklük ölçüsünün, maliyet ve işgücü kestiriminin yanında proje yöneticilerinin birçok ihtiyacına cevap verecek şekilde kullanılabileceği gözlemlenmiştir. İşlevsel büyüklüğü ölçen veya proje yönetiminde çeşitli amaçlarla kullanan model, yöntem ve araçların geliştirilmesi ve iyileştirilmesinde, kullanışlılığın ve uygulanabilirliğin artırılması ve tüm yönetim süreçlerinde etkin kullanımın sağlanması için yukarıda bahsi geçen kullanım alanları ve bunlara ilişkin gereksinimler dikkate alınmalıdır.

4.4.3 Bütünleşik Ölçüm Araç Prototip Geliştirilmesi

Proje boyunca birleşik yöntem için geçerleme yapmak, ölçüm verisi toplamak ve getirilen önerileri sınamak yada uygulamak amacıyla prototip bir İBÖ aracı geliştirilmiştir. Araç MS Access ürünü ile uygulanmıştır. Aracın araştırmanın devamında geliştirilerek, daha fazla yazılım projesi verisi ve işlevsel büyüklük ölçümünü destekleyecek çok katmanlı bir web uygulaması haline evrilmesine başlanmıştır. Araç ile ekran görüntüleri Ek-1'de verilmiştir.

5. Sonuç

Projede geliştirilen birleşik ölçüm modeli ve yaklaşımı ile modelin kapsamındaki İBÖ yöntemleri için tek bir ölçüm gerçekleştirilerek, her bir yöntem cinsinden büyüklükler elde edilebilmesi önerilmiştir. Bu öneri ile bir yazılım projesi için büyüklük temelli kıyaslama için belirlenebilecek kit proje sayısı, kullanılan İBÖ yöntemi kümesinden herhangi biri yeterli olacağından, artabilecektir. Bu da kıyaslamalardaki kıyas kümesini büyütür, daha gerçekçi kıyaslamalara destek olabilir. Proje kapsamında yapılan ilk geçerleme çalışmaları bu öneriyi destekler niteliktedir. Birleşik yaklaşımda, özellikle ölçüm sırasında yöntemin gereği olan veri gruplarının ve tiplerinin belirlenmesi sırasında varlık-ilişki formalitesinde sağlanan yazılım kullanıcı verisi ile ölçümlerin daha doğru yapıldığı gözlemlenmiştir. Bunun yanında ölçüm yöntemlerinde uygulanabilecek formalizasyonlara yönelik çalışmalarda(4.1.2,4.1.3,4.3.1) yöntemlerde belirsiz veya yorumlara açık durumların tespit edildiği ve formal gösterim ihtiyaçları bulunduğu gösterilmiştir.

Aynı yazılım işlevsel gereksinimleri kümesinden farklı kişilerin aynı amaçla, aynı yöntemle elde edilen ölçüm sonuçlarındaki farklılıklar, İBÖ kullanım yaygılığını azaltan bir faktör olarak eleştirile gelmiştir. Bu konuda projede yapılan çalışmalar ile farklılığı yaratan faktörler durum çalışmaları ve deneylerle irdelenmiştir. Farklılıkların ne gibi eylemler ya da araştırmalar ile azaltılabileceği belirlenmiştir. COSMIC yöntemi için yapılan çalışmalar, bu yöntem için yayınlanan ilk örnekler olmuşlardır.

İşgücü-işlevsel büyüklük ilişkisi araştırmalarında, İBÖ yöntemlerinde ortak bir kavram olan işlevsel süreçler arasındaki benzerliklerin, toplam yazılım geliştirme işgücüne etkisi olduğu önerilmiştir. Bu konuda yapılan durum çalışmaları bu yönde ilerleme sağlamıştır. Bu ilişkiyi göz önüne alan bir istatistiksel işgücü-büyüklük modeli önerilmiştir. Benzeri çalışmalarda, işgücü-büyüklük ilişkisini toplam büyüklük üzerinden öte, işlevsellik tipleri ile kuran modeller ile ilişki daha başarılı şekilde kurulabilmiştir. Özetle bu konudaki proje çalışmaları ile, işgücü-büyüklük ilişkisini salt toplam büyüklük ölçüsünden öte, ölçüm kavramları ve işlevsel bileşen cinsinden değerlendirmeler ile daha doğru kurulması desteklenmiş ve gözlemlenmiştir.

Ölçüm aracına yönelik çalışmalarda, proje yönetim süreçlerinin ölçüm sonuçlarını kullanım tipleri belirlenmiştir. Varolan araçların ölçüm süreçlerine olan destekleri değerlendirilmiştir, bu konuda fazla sayıda iyileştirme fırsatı belirlenmiştir. Araçta, birleşik yaklaşımı ve proje kapsamında işgücü ilişkisini daha doğru kurmaya yönelik çalışmalarda gerekli olan detayda ölçüm verisinin saklanması gereği belirlenmiştir. Bunun yanına proje ömür döngüsü boyunca değişiklik temelli ve projeler arası etkileşimi dikkate alan bir ölçüm modelini destekleyecek özellikler (konfigürasyon, değişiklik yönetimi, portföy ölçümü vb.) belirlenmiştir Bunun yanında proje yönetim araçları ile entegrasyonu ve otomasyonu

destekleyecek esnekliğe ve genişletmeye uygun bir yazılım mimarisine gereği belirlenmiştir. Prototip geliştirme, birleşik ölçüm yapma ile kısıtlı kalmıştır.

Çalışmalar çeşitli dergi [81], konferans bildirisinde yayınlanmış [82], [89], [90], [91], [92], [93], [94], [95], [96], [97], [102] ve teknik raporlarda basılmıştır [84], [85], [86], [87], [83], [104].

Bu araştırma projesi kapsamında belirlenen araştırma problemlerine yönelik yapılan çalışmaların devamı niteliğinde çalışma ve araştırma konuları belirlenmiştir:

Daha fazla proje ve ölçüm verisi ile işgücü-işlevsel büyüklük ilişkisinin incelenmesi: Araştırmaların daha geniş veri kümesi üzerinden yapılması ile kıyaslama (benchmarking) altyapısı oluşturulması düşünülmektedir. Prototip aracın genişletilip, mimarisin uygun hale getirilmesiyle, yazılım organizasyonlarından daha fazla ölçüm ve proje bilgisi alınabilir. ISBSG gibi varolan kıyaslama proje verisi kümelerindeki verilerin kirliliği ve gerekli detayda büyüklük ölçümü sonuçları içermemesi sorunu da bu şekliyle ele alınacaktır. Bu fikir projelendirilmiş ve TÜBİTAK 1001 programı desteği ile birlikte çalışılmaya başlanmıştır.

Veri Analizi için Formal Öneriler: Projede varlık-ilişki modelleri elemanları cinsinden yapılan kavram eşleştirmelerinin, UML-Sınıf modelleri gibi farklı gösterimler için genelleştirilerek, farklı teknikler ve gösterimlerle ifade edilen İG'lerinden ölçüm yapılması sağlanabilir. Bu dönüşümler için formal gösterimler sağlanırsa araç otomasyonu mümkün kılınabilir.

Ölçüm farklılıklarını Azaltma: Aynı yazılım için farklı ölçümcülerin elde ettiği farklı sonuçlar konusunda projede tespit edilen kök sebeplerin eğitim, yöntem iyileştirmeleri kılavuz ve formal ölçüm kavramı tanımları ile kabul edilebilir seviyelere getirilebileceği düşünülmektedir. Bu konu ile ilgili kontrollü deney çalışmalarına devam edilmektedir.

6. Referanslar

- [1] Abran, A.: COSMIC FFP 2.0: An Implementation of COSMIC Functional Size Measurement Concepts. FESMA'99, Amsterdam, October 7. 1999.
- [2] Abran, A., D. St-Pierre, M. Maya, and J.M. Desharnais: Full Function Points for Embedded and Real-Time Software. UKSMA Fall Conference, London (UK), October 30-31 1998.
- [3] Abran, A.: Analysis of the Measurement Process of Function Point Analysis. A PhD Thesis Submitted to Department of Electrical Engineering and Software Engineering, École Polytechnique De Montréal, 1994.
- [4] Abran, A., Robillard, P.N: Function Points: A Study of Their Measurement Processes and Scale Transformations. Journal of Systems Software, No.25, pp.171-184, 1994.
- [5] Abran, A., Ndiaye, I., Bourque, P.: Contribution of Software Size in Effort Estimation. Research Lab. in Software Engineering, École de Technologie Supérieure, Canada, 2003.
- [6] Abran A., Gil B., Lefebvre E.: Estimation Models Based on Functional Profiles. International Workshop on Software Measurement -- IWSM/MetriKon, Kronisburg (Germany), Shaker Verlag, (2004), 195-211.
- [7] Abran, A., Panteliuc, A.: Estimation Models Based on Functional Profiles. III Taller Internacional de Calidad en Tecnologias de Informacion et de Comunicaciones, Cuba, February 15-16. 2007
- [8] Albrecht, A.J.: Measuring Application Development Productivity, Proceedings IBM Applications Development Symposium, Monterey, California, October 14-17. 1979.
- [9] Albrecht, A.J. and Gaffney, J.E.: Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, IEEE Transactions on Software Engineering, vol. SE-9, no. 6, November 1983.
- [10] Angelis L., Stamelos, I. Morisio, M.: Building a Cost Estimation Model Based on Categorical Data. 7th IEEE Int. Software Metrics Symposium (METRICS 2001), London, (April 2001).
- [11] Banker, R., R.J. Kauffman, C. Wright, D. Zweig: Automating Output Size and Reuse Metrics in a Repository Based Computer Aided Software Engineering (CASE) Environment. IEEE Transactions on Software Engineering, Vol.20, No.3, 1994.
- [12] Boehm, B.W.: Software Engineering Economics, Prentice-Hall, 1981.
- [13] Boehm, B.W., Horowitz, E., Madachy, R., Reifer, D., Bradford K.C., Steece, B., Brown, A.W., Chulani, S., Abts, C.: Software Cost Estimation with COCOMO II, Prentice Hall, New Jersey, 2000.
- [14] Briand, L.C., El Emam K., Maxwell, K., Surmann, D., Wiczorek, I.: An Assessment and Comparison of Common Software Cost Estimation Models. in Proc. of the 21st Intern. Conference on Software Engineering, ICSE 99, Los Angeles, CA, USA, 1998 313-322.
- [15] Briand, L.C., Langley, T., Wiczorek, I.: A Replicated Assessment and Comparison of Software Cost Modeling Techniques. In Proc. of the 22nd Intern. Conf. on Software engineering, ICSE 00, Limerick, Ireland 2000, 377-386.
- [16] Caldiera, G., G. Antonioli, R. Fiutem, C. Lokan: Definition and Experimental Evaluation for Object Oriented Systems. Proc. of the 5th Intern. Symposium on Software Metrics, Bethesda, November 1998.
- [17] Conte, M., T. Iorio, R. Meli, L. Santillo: E&Q: An Early & Quick Approach to Functional Size Measurement Methods. In Software Measurement European Forum (SMEF), Rome, Italy, 2004.
- [18] DeMarco, T.: Controlling Software Projects, Yourdon press, New York, 1982.
- [19] Demirors, O., Gencel, C.: A Comparison of Size Estimation Techniques Applied Early in the Life Cycle, European Software Process Improvement Conference (EurSPI 2004), Springer Verlag Springer Lecture Notes in Computer Science (LNCS), 2004.
- [20] Demirors, O., Gencel, C.: Conceptual Differences Among Functional Size Measurement Methods, Empirical Software Engineering and Measurement-ESEM, 2007.
- [21] Demirors, O., Gencel, C.: Conceptual Association of Functional Size Measurement Methods, IEEE Software (2008) (değerlendirme aşamasında)
- [22] Efe, P. A Unification Model and Tool Support for Software Functional Size Measurement Methods, A MSc Thesis submitted to Middle East Technical University, Dept. of Information Systems (2006).
- [23] Efe, P., O. Demirors, C. Gencel: A Unified Model for Functional Size Measurement Methods. Proceedings of the Joint Conference 16th International Workshop on Software Measurement (IWSM2006) - DASMA Metrik Kongress (MetriKon2006), (2006), Potsdam, Germany, 343-358.
- [24] Fenton, N.E. and S.L. Pflieger: Software Metrics: A Rigorous and Practical Approach Second Edition, International Thomson Computer Press, Boston, 1996.
- [25] Fenton, N., Software Measurement: A Necessary Scientific Basis, IEEE Transactions on Software Engineering, Vol.20, No.3, March 1994.
- [26] Fetcke T., A. Abran, R. Dumke: A Generalized Representation for Selected Functional Size Measurement Methods, In Current Trends in Software Measurement, R. Dumke Ed. and A. Abran Ed., Shaker, (2001), 1-25.
- [27] Forselius, P.: Benchmarking Software-Development Productivity. IEEE Software, Vol. 17, No. 1, (Jan./ Feb. 2000), 80-88.
- [28] Forselius, P.: Finnish Software Measurement Association (FiSMA), FSM Working Group: FiSMA Functional Size, 2004.

- [29] Gencel, C.: An Architectural Dimensions Based Software Functional Size Measurement Method, a PhD Thesis Submitted to Dept. of Information Systems, Informatics Institute, Middle East Technical University, Ankara, Turkey, 2005.
- [30] Gencel, C., O. Demirors, E. Yuceer: A Case Study on Using Functional Size Measurement Methods for Real Time Systems. Proceedings of the 15th. International Workshop on Software Measurement-IWSM 2005, September 12-14, pp.159-178, 2005.
- [31] Gencel, C., O. Demirors: Functional Size Measurement Revisited. Scheduled for publication in ACM Transactions on Software Engineering and Methodology, 2007.
- [32] Hastings, T.E., Sajeev, A.S.M.: A Vector-Based Approach to Software Size Measurement and Effort Estimation. IEEE Transactions on Software Engineering, Vol. 27, No. 4 (April 2001), 337-350.
- [33] IEEE Std. 14143.1-2000, Implementation Note for IEEE Adoption of ISO/IEC 14143-1:1998 - Information Technology- Software Measurement- Functional Size Measurement -Part 1: Definition of Concepts, 2000.
- [34] IFPUG: Counting Practices Manual - Release. 4.1, International Function Point Users Group, Westerville, OH, 1999.
- [35] ISBSG (International Software Benchmarking Standard Group) Dataset, [http:// www.isbsg.org](http://www.isbsg.org), Estimating, Benchmarking & Research Suite Release 8 & 9, International Software Benchmarking Standards Group – ISBSG, Australia, 2005.
- [36] ISBSG: Software Project Estimates - How accurate are they?, <http://www.isbsg.org>.
- [37] ISO/IEC 14143-1:1998 Information Technology - Software Measurement - Functional Size Measurement - Part 1: Definition of Concepts, 1998.
- [38] ISO/IEC 14143-1:2007 Information Technology - Software Measurement - Functional Size Measurement - Part 1: Definition of Concepts, 2007.
- [39] ISO/IEC 14143-2:2002 Information Technology - Software Measurement - Functional Size Measurement - Part 2: Conformity Evaluation of Software Size Measurement Methods to ISO/IEC 14143-1:1998, 2002.
- [40] ISO/IEC TR 14143-3:2003 Information Technology - Software Measurement - Functional Size Measurement - Part 3: Verification of Functional Size Measurement Methods, 2003.
- [41] ISO/IEC TR 14143-4:2002 Information Technology - Software Measurement - Functional Size Measurement - Part 4: Reference Model, 2002.
- [42] ISO/IEC TR 14143-5:2004 Information Technology- Software Measurement - Functional Size Measurement - Part 5: Determination of Functional Domains for Use with Functional Size Measurement, 2004.
- [43] ISO/IEC FCD 14143-6:2005 Guide for the Use of ISO/IEC 14143 and related International Standards, 2005.
- [44] ISO/IEC 19761:2003: COSMIC Full Function Points Measurement Manual v. 2.2, 2003.
- [45] ISO/IEC 20926, Software engineering - IFPUG 4.1 Unadjusted FSM Method - Counting Practices Manual, 2003.
- [46] ISO/IEC 20968, Software engineering - Mk II Function Point Analysis - Counting Practices Manual, 2002.
- [47] ISO/IEC 24570:2005: Software engineering - NESMA Functional Size Measurement Method v.2.1 - Definitions and counting guidelines for the application of Function Point Analysis, 2005.
- [48] Jeffery, D.R., Stathis, J.: Function Point Sizing: Structure, Validity and Applicability. Journal of Empirical Software Engineering, Vol.1, No.1, pp.11-30, 1996.
- [49] Jeffery, R., Ruhe, M., Wieczorek, I.: A Comparative Study of Two Software Development Cost Modeling Techniques using Multi-organizational and Company-specific Data. Information and Software Technology, vol. 42, (2000), 1009-1016.
- [50] Jones, T. C, A Short History of Function Points and Feature Points, Software Productivity Research Inc., USA, 1987.
- [51] Jørgensen, M., Molokken-Ostfold, K.: Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach, and Data Analysis Method. IEEE Transactions on Software Engineering, Vol. 30, No. 12, (December 2004), 993-1007.
- [52] Kammelar, J.: A Sizing Approach for OO-environments. 4th Intern. ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering, 2000.
- [53] Kauffman, R., R. Kumar, "Investigating Object-Based Metrics for Representing Software Output Size", Conference on Information Systems and Technology (CIST), in the INFORMS 1997 Annual Conference, San Diego, May 1997.
- [54] Kitchenham, B., Kansala, K.: Inter-item Correlations among Function Points. In Proc. of the First International Metrics Symposium, May 21-22, IEEE Computer Society, pp. 11-14, 1993.
- [55] Kitchenham, B.: The Problem with Function Points. IEEE Software, Vol. 14, Issue: 2, pp. 29-31, Mar/Apr 1997.
- [56] Kitchenham, B., Mendes, E.: Software Productivity Measurement Using Multiple Size Measures. IEEE Transactions on Software Engineering, Vol. 30, No. 12, (December 2004), 1023-1035.
- [57] Leung, H., Fan, Z.: Software Cost Estimation. Handbook of Software Engineering, Hong Kong Polytechnic University, (2002).
- [58] Lokan, C.J.: An Empirical Study of the Correlations Between Function Point Elements. Sixth IEEE International Symposium on Software Metrics, Boca Raton, Florida, November 04 - 06 1999.
- [59] Lokan, C., Wright, T., Hill, P.R., Stringer, M.: Organizational Benchmarking Using the ISBSG Data Repository. IEEE Software, Vol. 18, No. 5, (Sept./Oct. 2001), 26-32.

- [60]Lother, M. and Dumke, R.R: Points Metrics - Comparison and Analysis. In International Workshop on Software Measurement (IWSM'01), pp. 155-172, Montréal, Québec, 2001.
- [61]Maxwell, K.D.: Collecting Data for Comparability: Benchmarking Software Development Productivity. IEEE Software, Vol. 18, No. 5, (Sept./Oct. 2001), 22-25.
- [62]Matson, J.E., B.E. Barret, and J.M. Mellichamp: Software Development Cost Estimation Using Function Points. IEEE Transactions on Software Engineering, Vol. 20, No. 4, April 1994.
- [63]Meli, R.: Early and Extended Function Point: A New Method for Function Points Estimation. IFPUG-Fall Conference, Scottsdale, Arizona, USA, September 15-19 1997.
- [64]Meli, R.: Early Function Points: A New Estimation Method for Software Projects. ESCOM 97, Berlin, 1997.
- [65]Meli, R., A. Abran, , V.T. Ho, and S. Oligny: On the Applicability of COSMIC-FFP for Measuring Software Throughout Its Life Cycle. Escom-Scope, 2000.
- [66]Menzies, T., Chen, Z., Hihn, J., Lum, K.: Selecting Best Practices for Effort Estimation. IEEE Transactions on Software Engineering, Vol. 32, No. 11, (November 2006), 883-895.
- [67]MK II Function Point Analysis Counting Practices Manual Version 1.3.1, UKSMA: United Kingdom Software Metrics Association, 1998.
- [68]Morasca, S., Russo, G.: An Empirical Study of Software Productivity. In Proc. of the 25th Intern. Computer Software and Applications Conf. on Invigorating Software Development (2001), 317-322.
- [69]NESMA: Netherlands Software Metrics Association, Definitions and Counting Guidelines for the Application of Function Point Analysis, Version 2.0, 1997.
- [70]Pastor, O., S.M. Abrahão, J.C. Molina, and I. Torres: A FPA-like Measure for Object Oriented Systems from Conceptual Models. 11th International Workshop on Software Measurement - IWSM'01, Montréal, Canada, Shaker Verlag, pp. 51-69, 2001.
- [71]Premraj, R., Shepperd, M.J., Kitchenham, B., Forselius, P.: An Empirical Analysis of Software Productivity over Time. 11th IEEE International Symposium on Software Metrics (Metrics 2005), IEEE Computer Society, (2005), 37.
- [72]Reifer, D.J.: Asset-R: A Function Point Sizing Tool for Scientific and Real-time Systems. Journal of Systems and Software, Vol.11, No.3, pp.159-171, 1990.
- [73]Rollo, T.: Sizing e-commerce. Proc. of the ACOSM 2000 – Australian Conference on Software Measurement, Sydney, 2000.
- [74]Rule, G.: A Comparison of the Mark II and IFPUG Variants of Function Point Analysis. [Online], Available from: <http://www.gifpa.co.uk/library/Papers/Rule/MK2IFPUG.html>, 1999.
- [75]Symons, C.: Function Point Analysis: Difficulties and Improvements. IEEE Transactions on Software Engineering, Vol. 14, No. 1, January, 1988.
- [76]Symons, C.: Come Back Function Point Analysis (Modernized) – All is Forgiven!, Proc. of the 4th European Conference on Software Measurement and ICT Control, FESMA-DASMA 2001, Germany, 2001, pp. 413-426.
- [77]Teoglou, G.: Measuring OO Software with Predictive Object Points. Shaker Publ., ISBN 90-423-0075-2, 1999.
- [78]The Common Software Measurement International Consortium (COSMIC), Measurement Manual, Version 2.2, 2003.
- [79]The Common Software Measurement International Consortium (COSMIC), COSMIC Method Version 3.0, Measurement Manual, 2007.
- [80]Whitmire, S.A.: 3D Function Points: Scientific and Real-time Extensions to Function Points. Pacific Northwest Software Quality Conference, 1992.
- [81]Onur Demirsors, Cigdem Gencel, "Conceptual Association of Functional Size Measurement Methods," IEEE Software, vol. 26, no. 3, pp. 71-78, May/June 2009
- [82]Baris Özkan, Onur Demirörs: Formalization Studies in Functional Size Measurement: How Do They Help? IWSM/Mensura 2009: 197-211
- [83]Turetken, O., Mapping IFPUG & COSMIC FSM Terminologies to ER Concepts , METU II. Technical Report METU/II-TR-2008-22. Middle East Technical University, Informatics Institute.
- [84]Ardic, A. (2008). İşlevsel Büyüklük Ölçme Yöntemleri için Bütünleşik Yöntem Uygulaması: Hazır Yazılı Çevrimiçi Bankacılık Uygulaması Üzerine Durum Çalışması', Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [85]Şentürk, O. (2008). 'İşlevsel Büyüklük Ölçme Yöntemleri için Bütünleşik Yöntem üzerine Durum Çalışması: Kurumsal Çözümler Ağ Uygulamaları', Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [86]Ergüden, E. (2008). İşlevsel Büyüklük Ölçme Yöntemleri için Bütünleşik Yöntem Uygulaması: Ağ Uygulamaları üzerine Durum Çalışması', Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [87]Aslan, U. (2009), Veri İşleme Ağırlıklı Projelerin Fonksiyonel Büyüklük Ölçümünde Birleşik Fonksiyonel Büyüklük Kestirim Modelinin Kullanılması Üzerine Örnekli Çalışma, Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [88]Santillo, L., Abran, A., (2006), "Software Reuse Evaluation Based on Functional Similarity in COSMIC-FFP Size Components", in Software Measurement European Forum –SMEF'2006, May 10-12, 2006, Rome, Italy, pp. 167-176.
- [89]Top, O. O., Tunallilar, S., and Demirsors, O. 2008. Evaluation of the Effect of Functional Similarities on Development Effort. In Proceedings of the 2008 34th Euromicro Conference Software Engineering and Advanced Applications (September 03 - 05, 2008).

- [90] Top, Ö., İşlevsel benzerliklerin işlevsel büyüklük ve yazılım geliştirme işgücü arasındaki ilişkiye olan etkisi, 2009, Yüksek Lisans Tezi, Enformatik Enstitüsü, ODTÜ.
- [91] Özden Özcan Top, Onur Demirörs, Oktay Türetken: Making functional similarity count for more reliable effort prediction models. ISCIS 2009.
- [92] Tunalılar, S., Demirörs O., Effect of Functional Similarity for Establishing Relation between Effort and Functional Size, SPACE 2008.
- [93] Tunalılar, S., Demirörs O: A Comparison of Neural Network Model and Regression Model Approaches Based on Sub-functional Components. IWSM/Mensura 2009: 272-284
- [94] Türetken, O., Demirörs, O., Gencil. C. (2008). "The Effect of Entity Generalization on Software Functional Sizing: A Case Study". PROFES 2008, 105–116.
- [95] Türetken O., Top O., Özkan B., Demirörs O, The Impact of Individual Assumptions on Functional Size Measurement. IWSM/Metrikon/Mensura 2008: 155-169
- [96] Ungan E., Top O., Özkan B., Demirörs O: An Experimental Study on the Reliability of COSMIC Measurement Results. IWSM/Mensura 2009: 321-336
- [97] Özden Özcan Top, Onur Demirörs, Baris Özkan: Reliability of COSMIC Functional Size Measurement Results: A Multiple Case Study on Industry Cases. EUROMICRO-SEAA 2009: 327-334
- [98] Mendes, O., Abran, A., Bourque, P., (1996) FP Tool Classification Framework and Market Survey, SEMRL, IFPUG Fall Conference, Dallas, International Function Point Users Group.
- [99] Mendes, O., Abran, A., Bourque, P., (1996) Function Point Tool Market Survey, Montréal, Université du Québec à Montréal – LRGL.
- [100] Stambollian, A.; Abran, A., Survey of Automation Tools Supporting COSMIV-FFP - ISO 19761, IWSM-Metrikom 2006, 435-454 .
- [101] Özkan B., Evaluation Of Functional Size Measurement Tools From A Process Perspective, Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [102] Ozkan, B., Türetken, O. & Demirörs, O., "Software Functional Size: For Cost Estimation and More", EuroSPI 2008, 59–69.
- [103] A Guide to the Project Management Body of Knowledge (PMBOK Guides). Project Management Institute, 2004.
- [104] Özcan Top, Ö., Demirörs, O., Özkan B., (2009) "Reliability of COSMIC Functional Size Measurement Results", Teknik Rapor, Enformatik Enstitüsü, ODTÜ

EK-I: PROTOTİP ARAÇ EKРАН GÖRÜNTÜLERİ

PROJECT ENTITIES

Project and Data Information

PROJECT 1 Movie Manager

Entities Entity Of Interests Logical Files Object Of Interests

| ENTITY NAME | DESCRIPTION | ENTITY TYPE | # OF Attributes |
|-------------|-------------|-------------|-----------------|
| MOVIE | | Core | 4 |
| PERSON | | Core | 4 |
| DIRECTOR | | Associative | 2 |
| CAST | | Associative | 3 |
| WRITER | | Associative | 3 |
| PRODUCER | | Associative | 3 |
| MOVIE_GENRE | | Attributive | 2 |
| * | | Core | |

Entity ATTRIBUTES

| Attribute Name | DESCRIPTION | Primary Key? | Foreign Key? |
|----------------|-------------|-------------------------------------|--------------------------|
| MOV_ID | | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| TITLE | | <input type="checkbox"/> | <input type="checkbox"/> |
| YEAR | | <input type="checkbox"/> | <input type="checkbox"/> |
| PROD_COMPANY | | <input type="checkbox"/> | <input type="checkbox"/> |
| * | | <input type="checkbox"/> | <input type="checkbox"/> |

Enter Entity Relationships

Generate EOIs Generate LFs Generate OOIs Transactions Quit

Record: 1 of 2 No Filter Search

Form View

Varlık Tanımlama

PROJECT ENTITIES FRM_ENTITY

Entity Relationships [Navigation Buttons]

Entity Name: MOVIE DESCRIPTION

Entity Type: Core

of Attributes: 4

FROM 'MOVIE' TO ...

| REL_ID | FROM ... | TO ... | Relationship Type | Entity Dependency |
|---------|----------|-------------|-------------------|---------------------|
| 1 | MOVIE | DIRECTOR | 1 : (N) | B is dependent on A |
| 2 | MOVIE | CAST | 1 : (N) | B is dependent on A |
| 4 | MOVIE | WRITER | 1 : (N) | B is dependent on A |
| 5 | MOVIE | PRODUCER | 1 : (N) | B is dependent on A |
| 9 | MOVIE | MOVIE_GENRE | 1 : (N) | B is dependent on A |
| * (New) | MOVIE | | | |

FROM ... TO 'MOVIE' **This form is 'READ ONLY'

| REL_ID | FROM ... | TO ... | Relationship Type | Entity Dependency |
|---------|----------|--------|-------------------|-------------------|
| * (New) | | MOVIE | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Close

Record: 1 of 7 Filtered Search

Form View

İlişki Tanımlama

UFSM-Movie Manager v01-2

Home Create External Data Database Tools Acrobat

View Paste Copy Format Painter View Clipboard Font Rich Text Records Sort & Filter Find

PROJECT ENTITIES TRANSACTIONS

PROJECT: **Movie Manager**

TRANSACTIONS

| ID | TRANSACTION | TYPE (IFPUG) | DESCRIPTION |
|----|---------------------|-----------------------|-------------|
| 1 | Add Person | EI - External Input | |
| 2 | List Persons | EQ - External Inquiry | |
| 8 | Retrieve Person | EQ - External Inquiry | |
| 9 | Update Person | EI - External Input | |
| 10 | Add Movie | EI - External Input | |
| 11 | Query Movie | EQ - External Inquiry | |
| 12 | List Movie Details | EQ - External Inquiry | |
| 13 | Query Person | EQ - External Inquiry | |
| 14 | List Person Detail: | EQ - External Inquiry | |
| 15 | List Movies | EQ - External Inquiry | |
| 16 | Delete Movie | EI - External Input | |
| * | (New) | | |

TRANSACTION PARTS

Input Processing Output

| Input EOI | # of DETs Inp | DESCRIPTION |
|-----------|---------------|-------------------------------------|
| PERSON | 4 | Name, Date of Birth, Place of Birth |
| * | | |

IFPUG Report COSMIC Report Close

Form View

İşlevsel Süreç (Transaction) Tanımlama

COSMIC Report

PROJECT Movie Manager

| ID | FUNCTIONAL PROCESS | DESCRIPTION | Entry | Read | Write | eXit | Size (COSMIC) |
|---------------|---------------------|-------------|-----------|-----------|-----------|-----------|---------------|
| 1 | Add Person | | 1 | | 1 | 1 | |
| 2 | List Persons | | 1 | 1 | | 1 | |
| 8 | Retrieve Person | | 1 | 1 | | 1 | |
| 9 | Update Person | | 1 | | 1 | 1 | |
| 10 | Add Movie | | 5 | 1 | 5 | 2 | |
| 11 | Query Movie | | 1 | 1 | | 1 | |
| 12 | List Movie Details | | 1 | 6 | | 6 | |
| 13 | Query Person | | 1 | 1 | | 1 | |
| 14 | List Person Details | | 1 | 6 | | 6 | |
| 15 | List Movies | | 1 | 1 | | 1 | |
| 16 | Delete Movie | | 1 | 1 | 5 | 1 | |
| TOTALS | | | 15 | 19 | 12 | 22 | |

COSMIC Ölçüm Sonucu Raporu

IFPUG REPORT

PROJECT **Movie Manager**

1

DATA FUNCTIONS

| ID | NAME | DESCRIPTION | LF TYPE | # of RETs | # of DETs | COMPLEXITY | SIZE (IFPUG FP) |
|----|--------|-------------|----------------|-----------|-----------|------------|-----------------|
| 2 | MOVIE | - | ILF - Internal | 5 | 11 | Low | 7 |
| 3 | PERSON | - | ILF - Internal | 5 | 10 | Low | 7 |

OF DATA FUNCTIONS: 2

TOTAL SIZE of TRANSACTIONAL FUNCTIONS: 14

TRANSACTIONAL FUNCTIONS

| ID | NAME | TYP | DESCRIPTION | # of FTR | # of DETs | COMPLX | FP |
|----|---------------------|------|-------------|----------|-----------|---------|----|
| 1 | Add Person | EI - | | 1 | 4 | Low | 3 |
| 2 | List Persons | EQ - | | 1 | 2 | Low | 3 |
| 8 | Retrieve Person | EQ - | | 1 | 4 | Low | 3 |
| 9 | Update Person | EI - | | 1 | 4 | Low | 3 |
| 10 | Add Movie | EI - | | 2 | 12 | Average | 4 |
| 11 | Query Movie | EQ - | | 1 | 2 | Low | 3 |
| 12 | List Movie Details | EQ - | | 2 | 12 | Average | 4 |
| 13 | Query Person | EQ - | | 1 | 1 | Low | 3 |
| 14 | List Person Details | EQ - | | 2 | 12 | Average | 4 |
| 15 | List Movies | EQ - | | 1 | 2 | Low | 3 |
| 16 | Delete Movie | EI - | | 1 | 11 | Low | 3 |

OF TRANSACTIONAL FUNCTIONS

11

TOTAL SIZE of TRANSACTIONAL FUNCTIONS: 36

TOTAL UNADJUSTED FUNCTION POINT: 50

IFPUG Ölçüm Sonucu Raporu

TÜBİTAK
PROJE ÖZET BİLGİ FORMU

| |
|--|
| Proje No: 107E010 |
| Proje Başlığı: Bütünleşik Yazılım Kestirim ve Araç Seti |
| Proje Yürütücüsü ve Araştırmacılar: Yürütücü: Dr. Onur Demirörs Araştırmacı/Bursiyerler: Dr. Oktay Türetken Özden Özcan Top Barış Özkan Erdir Ungan |
| Projenin Yürütüldüğü Kuruluş ve Adresi: Orta Doğu Teknik Üniversitesi Enformatik Enstitüsü, İnönü Bulvarı, 06531, Ankara, Türkiye. |
| Destekleyen Kuruluş(ların) Adı ve Adresi: |
| Projenin Başlangıç ve Bitiş Tarihleri: 01.07.2007-01.01.2010 |
| Öz (en çok 70 kelime) Projesi kapsamında, İşlevsel Büyüklük Ölçümü(İBÖ) ile ilgili farklı İBÖ yöntemlerinin ve farklı yöntemlerle elde edilmiş ölçüm sonuçlarının karşılaştırılabilirliği, İBÖ işlevsel büyüklük ve işgücü arasındaki ilişki, ölçüm sonuçlarında bireyler arasındaki farklı sonuçlar ile İBÖ yöntemleri için araç desteği problemlerine yönelik öneriler getirilmiştir. Bu bağlamda, farklı İBÖ yöntemleri arasındaki benzerlik ve farklılıklar temel alan bir model ile birleşik bir ölçüm yaklaşımı ve bu yaklaşımla yapılan ölçümleri destekleyebilecek bir prototip araç geliştirilmiştir. Örnek çalışmalarla yaklaşım ve prototip araç geçerlenmiştir. İşgücü-işlevsel büyüklük ilişkisine yönelik, yazılımın sunduğu işlevsellikler arasındaki benzerliklerin ve farklı yazılım işlevsel bileşen tiplerinin büyüklük işgücü ilişkisine etkisi incelenmiştir. Bunun yanında bireysel ölçümlerle elde edilen sonuçlardaki farklılıklar, durum çalışmaları ile irdelenmiş, farklılıkları azaltma yönünde İBÖ yöntemlerinde ve birleşik ölçüm yaklaşımında çalışma konuları ve fırsatları belirlenmiştir. |
| Anahtar Kelimeler: Yazılım Büyüklük Ölçümü, İşgücü Kestirimi |

Fikri Ürün Bildirim Formu Sunuldu mu?

Evet

Gerekli Değil

Fikri Ürün Bildirim Formu'nun tesliminden sonra 3 ay içerisinde patent başvurusu yapılmalıdır.

Projeden Yapılan Yayınlar:

- [1] Onur Demirors, Cigdem Gencel, "Conceptual Association of Functional Size Measurement Methods," IEEE Software, vol. 26, no. 3, pp. 71-78, May/June 2009
- [2] Baris Özkan, Onur Demirörs: Formalization Studies in Functional Size Measurement: How Do They Help? IWSM/Mensura 2009: 197-211
- [3] Turetken, O. (2008), Mapping IFPUG & COSMIC FSM Terminologies to ER Concepts , Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [4] Ardic, A. (2008). İşlevsel Büyüklük Ölçme Yöntemleri için Bütünleşik Yöntem Uygulaması: Hazır Yazılı Çevrimiçi Bankacılık Uygulaması Üzerine Durum Çalışması', Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [5] Şentürk, O. (2008). 'İşlevsel Büyüklük Ölçme Yöntemleri için Bütünleşik Yöntem üzerine Durum Çalışması: Kurumsal Çözümler Ağ Uygulamaları', Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [6] Ergüden, E. (2008). İşlevsel Büyüklük Ölçme Yöntemleri için Bütünleşik Yöntem Uygulaması: Ağ Uygulamaları üzerine Durum Çalışması', Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [7] Aslan, U. (2009), Veri İşleme Ağırlıklı Projelerin Fonksiyonel Büyüklük Ölçümünde Birleşik Fonksiyonel Büyüklük Kestirim Modelinin Kullanılması Üzerine Örnekli Çalışma, , Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [8] Top, O. O., Tunalılar, S., and Demirors, O., Evaluation of the Effect of Functional Similarities on Development Effort. EUROMICRO-SEAA 2009: 419-426
- [9] Top,Ö, İşlevsel benzerliklerin işlevsel büyüklük ve yazılım geliştirme işgücü arasındaki ilişkiye olan etkisi,2009, Yüksek Lisans Tezi,Enformatik Enstitüsü,ODTÜ.
- [10] Top, O. O, Demirörs O., Turetken O.: Making functional similarity count for more reliable effort prediction models. ISCIS 2009: 504-512.
- [11]Tunalılar, S., Demirors O., Effect of Functional Similarity for Establishing Relation between Effort and Functional Size,SPACE 2008.
- [12]Tunalılar, S., Demirors O: A Comparison of Neural Network Model and Regression Model Approaches Based on Sub-functional Components. IWSM/Mensura 2009: 272-284
- [13]Turetken, O., Demirors, O., Gencel. C. (2008). "The Effect of Entity Generalization on Software Functional Sizing: A Case Study". PROFES 2008, 105–116.
- [14]Turetken O., Top O., Özkan B., Demirörs O,The Impact of Individual Assumptions on Functional Size Measurement. IWSM/Metrikon/Mensura 2008: 155-169
- [15]Ungan E., Top O., Özkan B., Demirörs O: An Experimental Study on the Reliability of COSMIC Measurement Results. IWSM/Mensura 2009: 321-336
- [16]Özden Özcan Top, Onur Demirörs, Baris Özkan: Reliability of COSMIC Functional Size Measurement Results: A Multiple Case Study on Industry Cases. EUROMICRO-SEAA 2009: 327-334
- [17]Özkan B.,Evaluation Of Functional Size Measurement Tools From A Process Perspective, Teknik Rapor, ODTÜ Enformatik Enstitüsü.
- [18]Ozkan, B.,Turetken, O. & Demirors, O., "Software Functional Size: For Cost Estimation and More",EuroSPI 2008, 59–69.