

Türkiye'deki Yazılım Test Uygulamaları Anketi

Vahid Garousi^{1,2}, Ahmet Coşkunçay¹, Aysu Betin Can¹, Onur Demirörs¹

¹Informatics Institute
Middle East Technical University
Orta Doğu Teknik Üniversitesi
Ankara, Turkey

²Software Quality Engineering Research Group (SoftQual)
Department of Electrical and Computer Engineering, Schulich School of Engineering
University of Calgary, Calgary, Alberta, Canada

¹{vahid, cahmet, betincan, demirors}@metu.edu.tr
²vgarousi@ucalgary.ca

Özet. Bağlam: Yazılım testi, yazılım geliştirme yaşam döngüsünde önemli bir faaliyettir. Her bir yazılım takımı ve şirketinde gerçekleştirilen yazılım test uygulamalarının türleri ve olgunlukları geniş bir yelpazeye sahiptir. Bu yelpazeyi oluşturan test uygulamalarının türleri hakkında üst seviye bir bakış açısı betimlemek için değişik büyüklük ve bölgelerde (örneğin, Kanada, İsveç, ve Finlandiya) çeşitli anketler gerçekleştirilmiştir. Oldukça aktif olan Türkiye yazılım endüstrisinde de bu uygulamaların durumunu karakterize etmek ve anlamak önem taşımaktadır.

Amaç: Amacımız, Türk yazılım endüstrisindeki test uygulama türlerine üst seviye bir bakış açısını karakterize etmek ve kavrayabilmektir. Bu çalışmada da test uygulamalarının özellikle şu açıları ile ilgilenmekteyiz: test türleri/seviyeleri, test teknikleri, test otomasyonu ve test araçları, test ölçütleri, test yönetimi, ve aynı zamanda Türk test mühendislerinin karşılaştıkları zorluklar.

Yöntem: Yukarıda amaca ulaşmak için, Kanada bağlamındaki geçmiş deneyimimizi ve Software Engineering Body of Knowledge (SWEBOK)'ın kullanımını temel alan, 9'u yazılım testine odaklı 54 soruluk çevrimiçi bir anketi sistematik olarak tasarladık. Ankete, Türk yazılım endüstrisinden 163 yazılım mühendisi katıldı. Bu bildiride, yazılım testine odaklanan soruların sonuçlarını raporluyor ve sentezliyoruz. Mümkün olduğu ölçüde, anketimizin eğilim ve sonuçlarını Turkish Testing Board tarafından 2013'te gerçekleştirilen yeni bir anketin sonuçları ve Kanada yazılım endüstrisinde 2010'da gerçekleştirilen bir diğer benzer anket ile karşılaştırıyoruz.

Bulgular: Anket sonuçları Türkiye'deki yazılım test uygulamaları hakkında önemli ve ilginç bulguları ortaya koymaktadır. Bulgular arasında şunlar yer almaktadır: (1) İşlevsel (fonksiyonel)/sistem testi, Kullanıcı kabul testi, ve Entegrasyon testi en geniş kullanılan üç test yaklaşımıdır. En az kullanılan teknik ise Stres testidir. (2) Türk testçilerinin %10'undan daha azı formal test durumu tasarım yaklaşımlarından birini kullanırken, Kanada'da bu oran %15 ve %27 arası olarak biraz daha iyi durumdadır. (3) hem Kanada'da hem de Türkiye'de, manuel ve otomatik test yaklaşımları için genel olarak geniş bir yelpaze mevcuttur. Bu durum, farklı katılımcıların birbirlerinden çok farklı uygulamalara sahip olduğunu göstermektedir. Örnek olarak, bazıları otomatik testi yoğun olarak uygularken, diğerleri manuel testi tercih etmektedir. (4) Genel olarak Türk yazılım endüstrisinde kod kapsama ölçütlerinin popülerliği düşüktür. (5) Türkiye'de

Kod satırı (LOC) başına düşen (ortalama) hata sayısı, ve başarılı kullanıcı kabul testlerinin sayısı gibi test ve kalite ölçütleri yaygın olarak kullanılmamaktadır. (6) Çoğu şirkette, geliştiriciler testçilere göre 1:2'den, 1:5 ve üstü oranlarla sayıca üstündür. (7) Test-sonra Geliştirme yaklaşımları (yani, geliştirmeden sonra test etme) hala Test Gündümlü (test önce) Geliştirmeden çok daha popülerdir.

Sonuç: Anketimizin sonuçları hem Türkiye hem de dünyada test profesyonellerini ilgilendirecektir. Aynı zamanda, yazılım test endüstrisindeki son eğilimleri gözlemlemek ve güçlü ve zayıf alanları belirlemek bu alanda endüstri-akademi işbirliklerinin önünü açarak araştırmacıların yararına olacaktır.

Anahtar Kelimeler. Anket, yazılım testi, endüstri uygulamaları, Türkiye

1 Giriş

Yazılım testi, yazılım geliştirme yaşam döngüsünde önemli ve maliyetli bir aktivitedir. Dahası, yetersiz yazılım testi genellikle önemli risklere ve sonuçlara yol açar. Örneğin, American National Institute of Standards and Technology (NIST)'nin 2002 raporu [20], Birleşik Devletler'de yazılım test altyapısının olmamasının tek başına negatif etkisinin yıllık 62 Milyar Dolar olduğunu raporlar. Diğer ülkelerde de benzer zorlukların olduğunu söylemek mümkündür.

Türkiye canlı bir yazılım endüstrisine sahiptir. 2011 itibarıyla, Türkiye'de yaklaşık 1,600 yazılım geliştirme firması bulunmaktadır [21]. Bu endüstrideki uygulamaların durumunu karakterize etmek ve anlamak önemlidir. Dünya çapında, bilişim teknolojileri sektörü kaliteli ürünleri zamanında ve bütçe dâhilinde teslim etmekte devasa sorunlarla yüzleşmektedir. Standish Group konu hakkında düzenli olarak CHAOS isimli bir rapor serisini yayınlayan, bilişim teknolojileri proje ve değer performans analizinde çok bilinen bir pazar araştırması firmasıdır. Çok geniş bir bilişim teknolojileri proje havuzunu inceleyen 2001 CHAOS raporuna [22] göre, bu projelerin sadece %28'i başarılıydı (zamanında teslimat, istenen özellikler ve işlevler ile birlikte). Projelerin %23'ü başarısızlığa uğradı (yani, tamamlanamadan iptal edildi veya teslim edilmesine rağmen hiç kullanılmadı), ve %49'u zorlandı (yani, geç, bütçenin üstünde ve/veya gerekli özellik ve işlevlerin daha azıyla). En son 2009 CHAOS raporuna göre, 2009'da durum daha da kötüye gitmiştir [23]: bilişim teknolojileri projelerinin sadece %32'si başarılı iken, %44'ü başarısız olmuş ve %24'ü zorlanmıştır. Gerçekten de, yazılım testi ve kalite güvencesi yazılım projelerinin başarı veya başarısızlığını belirleyen önemli bir kilit taşıdır.

InformationWeek'in Nisan 2002'deki bir anketi, 800 BT çalışanının %97'sinin yazılım hatalarıyla ilgili problemleri olduğunu, ve onda dokuzunun bu problemlerin sonucu olarak yüksek maliyetler, gelir kaybı, veya her ikisini de yaşadığını raporlamaktadır [24]. Ayrıca, sürpriz bir şekilde, yazılım testi yazılım geliştirme toplam maliyetinin yaklaşık ortalama %80'ine mal olmaktadır [20]. Yazılım endüstrisinin bu görece acımasız manzarası yazılım mühendisleri olarak yöntem ve uygulamalarımızı sorgulamamıza yol açmaktadır. Bu çalışma, yeni yazılım geliştirme yöntemlerinde testin artan kritiklikteki rolü göz önüne alınarak, özellikle yazılım testi yöntem ve uygulamalarına odaklanmaktadır.

2004'teki daha önceki bir çalışmada [7], Geras ve arkadaşları, Kanada'nın Alberta eyaletinde uygulamadaki yazılım testi ve yazılım kalite güvence tekniklerinin bölgesel bir incelemesinin sonuçlarını betimlemişlerdir. Bu ilk çalışmanın beş yıl sonrasında, bu anket 2009'da tekrarlandı ve 2004'ten 2009'a eğilimlerdeki değişimler analiz edildi ve sonuçları Journal of Systems and Software (JSS)'de bir bildiri olarak yayınlandı [14]. Bu konuda daha büyük ülke çapında (Kanada) bir manzara elde etmek için, 2010'da güncellenmiş soru listesi ile birlikte [18]'de yayınlanan yeni bir anket gerçekleştirdik. Türkiye'deki test uygulamalarını karakterize etmek ve onları Kanada'dakiler ile karşılaştırmak için, anketi 2013'te Türkiye'de tekrar ettik. Bu bildiri 2013'te Türkiye çapındaki en son anketimizin tasarım, uygulama ve sonuçlarını raporluyoruz.

Bu bildirin devamı şu şekilde yapılandırılmıştır. İlgili çalışmaların bir incelemesi 2. Kısım'da sunulmuştur. Anketin tasarımı ve uygulamasını 3. Kısım'da açıklanmaktadır. 4. Kısım, anket bulgularını ve analizlerimizi sunmaktadır. 5. Kısım bulguları ve edinilen dersleri özetlemektedir. Son olarak, 6. Kısım'da sonuçları gelecek araştırmalar için alanlar önerileri sunmaktadır.

2 İlgili Çalışmalar

Yazılım test uygulamaları konusunda değişik ülke ve büyüklüklerde yüksek sayıda anket yapılmıştır. Yaptığımız literatür araştırması Tablo 1'de (tarihlerine göre sıralanmış) özetlenmiş ve ardından kısaca tartışılmıştır [1-19].

Bu anketlerin çoğunun son on yılda gerçekleştirilmiş olduğunu görmek (2002'den beri) bu türden anketlere ihtiyacın yazılım test endüstrisi daha olgun hale geldikçe arttığını göstermesi açısından ilginçtir. Listedeki 19 anket içinde, İsveçli yazılım topluluğu beş bildiri ile yazılım test uygulamaları hakkında anket gerçekleştirme ve raporlama açısından en aktiftir. Amerikalı araştırmacılar 1983 ve 1988'de ilk iki anket sonucunu raporlamıştır.

Tablo 1. Yazılım test uygulamaları konusundaki anketlerin bir özeti

Bildiri Referansı	Büyükklük/bölge	Yıl	Katılımcı sayısı	Hedef/Odak alanı
[1]	ABD	1983	207	Test prosedürleri
[2]	Washington, ABD	1988	Raporlanmadı	Test metodolojileri, test süreç modelleri, profesyonelliğin büyümesi
[3]	Hindistan	2002	Raporlanmadı	Test standartları/prosedürleri, donanım platformu, test uygulaması, eğitim, ölçütler, test sertifikasyonu
[4, 5]	İsveç	2003	11 İsveç şirketi	Test süreçleri
[6]	İsveç	2003	91	Genel yazılım testi (ve yeniden kullanım)

[7]	Alberta eyaleti, Kanada	2004	60	Test uygulamaları, test türleri/seviyeleri, teknikleri, otomasyon, ölçütleri, yönetimi, eğitimi
[8]	Avustralya	2004	65	Test teknikleri, araçları, ölçütleri, standartları
[9]	Finlandiya	2005	18 uzman	Testte araştırma istikametlerinin sıralanması
[10]	Avustralya	2006	35	Doğrulama ve geçerli kılma bağlamı, doğrulama ve geçerli kılma kullanımı karar süreci, enformasyon ve maliyet etkinlik vaka çalışması
[11]	İsveç	2006	17	Birim testi
[12]	İsveç	2006	12 yazılım organizasyonu	Test olgunluğu
[13]	Birleşik Krallık	2007	Bir yazılım firması	Entegrasyon testi, gereksinim testi, test kapsama, test otomasyon, test senaryo tasarımı
[14]	Alberta eyaleti, Kanada	2009	53	Test uygulamaları, test türleri/seviyeleri, teknikleri, otomasyon, araçlar, ölçütler, yönetim, eğitim
[15]	Çevrimiçi testçi topluluğu	2009	121	Test Güdümlü Geliştirme'nin benimsenme oranı ve Test Güdümlü Geliştirme ile ilişkili teknikler
[16]	İsveç	2010	32	Regresyon testi
[17]	Avrupa	2010	83	Yazılım testinin çağdaş görünümü
[18]	Kanada	2010	246	Test uygulamaları, test türleri/seviyeleri, teknikleri, otomasyon, araçlar, ölçütler, yönetim, eğitim, akademi ile araştırma ve etkileşim
[19]	Türkiye	2013	?	Test teknikleri, test yeterliği, test amacı, çıkış kriteri, bulut testi, mobil test, olgunluk ve standardizasyon, keşif testi, yetenek kümesi
Mevcut anket	Türkiye	2013	163	Test uygulamaları, test türleri/seviyeleri, teknikleri, otomasyon, araçlar, ölçütler, yönetim

[1, 2]'deki çalışmalar konu hakkındaki en erken çalışmalar olarak değerlendirilmektedir. [1]'deki çalışma ABD Genel Muhasebe Dairesi'nin ABD Hükümeti

Genel Hizmetler İdaresine sunduğu bir kamu raporudur ve başlığı: “Greater Emphasis on Testing needed to Make Computer Software more Reliable and Less Costly”dur. Bu raporda, ABD hükümetinin 207 çalışanı incelenmiş ve detaylı bir rapor üretilmiştir. Bulgular arasında ajansların her zaman test politikalarını ve gereksinimlerini mecburi etmedikleri bulunmaktadır. Örneğin, programcı değişikliği küçük olarak değerlendirdiği için bir maaş cetveli modifikasyonunda gerekli olan birim ve sistem testleri atlanmıştır.

Gelperin ve Hetzel [2] yazılım testinin büyüme ve endüstriye nüfuz etmesini raporlamıştır. Anketleri Haziran 1987’de Washington, DC’de gerçekleştirilen 4th International Conference on Software Testing’de endüstri katılımcıları tarafından doldurulmuştur. Çalışmanın ilginç gözlemlerinden biri şu ifadedir: “Test, diğer her şey gibi, ya az yapılmıştır ya da aşırı yapılmıştır”. En çok raporlanan uygulama test sırasında bulunan hataların kaydedilmesi (katılımcıların %73’ü). En az ortak uygulama ise test sırasında kod kapsamının (örneğin, çalıştırılmış çalıştırılabilir kaynak satır sayısı) ölçümüydü (sadece %5 bunu normal bir uygulama olarak görüyordu). Katılımcıların yüksek bir yüzdesi kurumlarında testin sistematik ve organize bir aktivite olduğunu hissediyordu (%91 evet cevabı verdi). Anketin [2] sonuçlarının son derece yanlış olduğu anlaşılmalıdır. Teknik konferanslara katılan kişiler test farkındalığı ve bilgisinin uç noktalarını temsil etmektedir. Bu yüzden, anket genel endüstri uygulamalarını değil, yazılım geliştirme topluluğunda hususların farkında olan ve onları iyileştirmek isteyen küçük bir parçanın uygulamalarını ölçmektedir. Bazı gözlemciler genel endüstri uygulamalarının anketin ortaya koyduğu kesitten çok daha kötü olduğuna inanmaktadır. Bu görünüm daha önceki [1] 1983 tarihli ABD Genel Muhasebe Dairesi’nin federal ajanslarda yazılım testi hakkındaki çalışmasının istatistikleri ile karşılaştırılarak desteklenebilir.

Bir diğer anket raporu [3] Quality Assurance Institute tarafından yazılım testi hakkındaki 2002 yılı Hindistan’daki uluslararası konferansında tamamlanan anketi temel almaktadır. Yukarıda bahsedilen ankete benzer şekilde [2], bu anketin sonuçları da yanlıdır, çünkü konferans katılımcıları muhtemelen testin değerini bilen yazılım test profesyonelleridir.

Test süreç uygulamalarını incelemek için 2003’te nitel bir anket gerçekleştirilmiştir [4, 5]. Yedi İsveç şirketinin cevaplarını analiz ettikten sonra, yazarlar şirketlerin test sürecinin değerine karşı çeşitli tutumları olduğunu buldular. Büyük kurumlar yazılı geliştirme sürecini anahtar bir varlık olarak vurgularken, küçük organizasyonlar daha çok kıdemli uygulayıcılara bel bağlamaktadır. Gelişim yaklaşımı olarak, artırılmış ve günlük derleme en popüler iki yaklaşım olarak raporlanmıştır. Son olarak, yazarlar iki tip test otomasyonu belirlemişlerdir: kayıtlı veri otomasyonu ve komutlu otomasyon. Sonuca göre, işlevsel olmayan özelliklere odaklanan ürünler öncelikli olarak kayıtlı veri kullanılarak test edilmektedir. Tam tersine, komutlu otomasyon tekniği çoğunlukla işlevselliğe odaklanan ürünlerin testi için kullanılmaktadır.

Torkar ve Mankefors [6] kod yeniden kullanımı bağlamında test aktivitelerini çalışmışlardır. Açık kaynak kodlu projelerden 91 İsveç kökenli katılımcı ve üç şirket ankete katkı vermiştir. Anket sonuçları, ankete katılan geliştiricilerin büyük çoğunluğunun yeniden kullanılan kodu test etmediklerini ve bilimsel topluluğun kabul ettiği diğer test metodolojilerinin kullanılmadığını göstermektedir. Kod kapsama analizi ve

istatistiksel analizle birleşim halinde daha otomatik bir yaklaşıma ihtiyaç ortaya çıkmıştır.

Ng ve arkadaşları [8] 2003 ve 2004 yıllarında Avustralya'da yazılım test uygulamalarını çalışmıştır. 65 endüstri katılımcısı anketi tamamlamıştır. Çalışma yazılım test metodoloji ve tekniklerinin uygulanmasında en belirgin bariyerin uygulayıcıların tecrübe yoksunluğu olduğunu bulmuştur. Bu bulgu, formal test metodolojileri ve tekniklerinin kullanımı için uygun eğitimi almamış çok miktarda yazılım test çalışanının olabileceğini önermektedir. Bu, endüstrinin gerçek talebini karşılamak için yazılım test profesyonellerinin eğitiminde bir eksikliğin veya tekniklerin kendilerindeki eksikliklerin göstergesi olabilir. Otomatik test araçlarının kullanımındaki bariyerler arasında maliyet ilk sırada yer almıştır.

Ng ve arkadaşları [8]'un ilk tahminine göre devlet ve kamu ticari olmayan kurumları, kamu tarafından finanse edildikleri için, yapısal test metodolojilerini benimsemeleri çok olasıdır. Ancak görüldü ki, özel kurumların yaklaşık %70'i yapısal test metodolojilerini benimserken, devlet ve kamu kurumları önemli ölçüde daha düşük bir yüzde bildirmişlerdir. Kurumların yaklaşık %75'i geliştirme bütçelerinin %40'ından daha azını yazılım test aktivitelerine ayırmakta ve yazılım testi aktivitelerinde yardımcı olmaları için dış testçiler kiralayan kurumlar arasında önemli bir memnuniyet seviyesi vardır. Kullanılan en popüler araç türleri test çalıştırmayı destekleyenlerdir (44 kurumun 35'i), onu regresyon testi (33 kurum) takip ederken, bulgu analizi ve raporlama araçları üçüncü sıradadır (27 kurum).

Taipale ve arkadaşları [9] yazılım test araştırmalarındaki hususları değerlendirmek üzere endüstri ve akademiden 18 uzmanı davet etmiştir. Bu anketteki ilk üç sıradaki araştırma hususu şunlardır: (1) test süreç iyileştirme, (2) test otomasyonu ve test araçları, ve (3) standardizasyon. Yazarlar bir adım daha ileriye atarak gelecek araştırmalar için hipotezler önermişlerdir. Yazarlara göre, yazılım geliştirme ve test süreçleri arasındaki bilgi akışındaki problemler geliştirme ve test işlerini arttırabilir. Ancak, yazarlar bu anket sonuçlarının uygulanabilirliğinin az olduğunu itiraf etmektedir, yalnızca benzer ortamlarda uygulanabilirler.

İki Avustralyalı araştırmacı tarafından gerçekleştirilen, eşzamanlı programlar için doğrulama ve geçerli kılma hakkındaki bir başka anketteki oranlar [10] kayda değerdir. Bu çalışmada IBM developerWorks multi-threaded Java programming forum'dan dünya çapında seçilen 35 katılımcı vardır [10]. Yukarıdaki anketteki çoğu katılımcı sistem testi gerçekleştirirken, bu anketteki katılımcıların çoğunun birim test gerçekleştirdiğini gözlemlemek ilginçtir.

Runeson [11] bir yazılım süreç iyileştirme ağındaki (SPIN) odak grup tartışmaları temelinde birim testi uygulamalarını incelemiştir. Amaçları yazılım uygulayıcıları için birim testin anlamını incelemektir. Anket sonuçlarının gösterdiğine göre, katılımcılar birim testin kapsamı hakkında uzlaşırken, test ortamının ayrı bir yazılım sistemi olup olmaması hakkında ayrı düşmektedirler. Dahası, yöneticiler ya da kalite güvence takımları birim testleri yalnızca geliştirici hususları olarak bırakmış ve test stratejileri ve uygulamalarına düşük önem vermişlerdir. Test kapsama bakımından, geliştiricilerin nadiren yapısal kapsamı ölçtüklerini görmek heves kırıcıdır.

Grindal ve arkadaşları [12] yazılım organizasyonlarının test olgunluğu hakkında bir mülakat çalışması bildirmişlerdir. On iki Avrupa kurumundan test yöneticileri ele

alınmıştır. Veri göstermektedir ki, düzgün yapısal test stratejilerinin yokluğu nedeniyle test olgunluğu düşük olsa da, bu organizasyonlar ticari olarak başarılıdır. Mülakat edilen birçok yöneticinin düşük test olgunluğu ile ilgili sorunlarını ifade etmesi, problem ile ilgili farkındalıklarının olduğunu göstermektedir. Yazar son olarak yöneticilere ürün kalitesinin ekonomik tabanlı yönetimi için bir ölçüt programına yoğunlaşmaları önerisini sunmuştur.

Martin ve arkadaşları [13] küçük bir yazılım evinden Birleşik Krallık temelli bir vaka çalışması raporlamıştır. Bulgularını göstermek için, yazarlar bu firmadaki bir entegrasyon testi örneğini detaylarıyla betimlemişlerdir. Örnek vasıtasıyla; test kapsama, gereksinim testi, test senaryo tasarımı gibi test ilgilerini şekillendiren bazı kurumsal gerçekleri ve kısıtları belirlemişlerdir. Son olarak, yazarlar test araştırmaları ile uygulamaları arasında bir bağlantısızlık olduğu sonucuna varmışlar ve bu ikisi arasındaki ilişkinin ele alınması ihtiyacına dikkat çekmişlerdir.

Test GÜdümlü Geliştirme hakkındaki yakın tarihli bir anket 2008'de gerçekleştirilmiş ve sonuçları [15]'te yayınlanmıştır. Anket Ekstrem Programlama ve Test GÜdümlü Geliştirme e-mail listelerinde duyurulmuş ve 121 katılımcı bulmuştur. Amaç, çevik literatüründe savunulan süre modeliyle karşılaştırıldığında çevik geliştiricilerin gerçek uygulamalarını incelemektir. Anketin ana bulgularından bazıları şunlardır. Ekstrem Programlama ve Test GÜdümlü Geliştirme toplulukları içinde; muayene, regresyon testi, yaşam döngüsü sonu testi gibi diğer test ve doğrulama tekniklerinde önemli bir odaklanma vardır. Bu topluluklarda Test GÜdümlü Geliştirmeye karşı açık bir yanlılık varsa da, Test GÜdümlü Geliştirme çevik geliştiricilerin net bir şekilde benimsediği tek geçerli kılma tekniği değildir. Belki de bu topluluk içinde daha çok teknik odak olduğu için, geliştirici Test GÜdümlü Geliştirme, kabul Test GÜdümlü Geliştirmeden daha yaygındır [25]. Her iki Test GÜdümlü Geliştirme türü için de (geliştirici ve kabul), tekniği öğrenmenin en etkin yolları tecrübeli bir kişi ile eşleşme ve akıl hocalığı olarak değerlendirilmiştir. Eğitim açık ara ile üçüncü gelmekte ve başka bir öğrenci ile eşleşme, okuma ve çevrimiçi forum tartışmaları gibi diğer teknikler tarafından takip edilmektedir. Anket verisi [15], birçok grup kabul Test GÜdümlü Geliştirme [25] kullandığını öne sürse de, hala gereksinim belirtimlerinin doküman veya modeller ile yakalanmasının onlar için daha yaygın olduğu savına kanıt sunmaktadır.

Regresyon testi hem akademi hem de endüstriden önemli bir ilgi görmüştür. Yakın tarihli bir anket 2009'da bu konuyu raporlamıştır [16]. Bu nitel anketi eşsiz kılan, yazarların önce 15 endüstri katılımcısı ile odak grup toplantıları gerçekleştirip, sonra sonucu 32 katılımcılı bir çevrimiçi anket ile geçerli kılmalarıdır. Regresyon testinin tanımı hakkında, formal bir toplantıda katılımcıların çoğu hemfikir olmuşlardır. Ancak, regresyon testi uygulaması değişik bağlamlarda kapsam ve önkoşulları farklılaşarak değişiklik göstermiştir. Öte yandan, bu çalışmada belirtilen çoğu zorluk regresyon testine özgü değil, diğer test türlerindeki genel zorluklardı. Yazarlar bu zorluklar arasından, test otomasyonu ve dost test ortamı desteğinin regresyon testi uygulamalarının etkinliğinde önemli etkiye sahip olduğunu belirtmişlerdir.

Causevic ve arkadaşları [17] yazılım test sürecinin algısına odaklanan bir endüstriyel anketi bildirmişlerdir. Analizden önce yazarlar katılımcıları dört farklı kategoriye bölmüşlerdir: güvenlik kritiklik, çeviklik, geliştiricinin dağılımı, ve uygulama alanı.

Bulguları, tercih edilen ve gerçek test uygulamaları arasında dikkate değer çelişkileri açığa çıkarmaktadır. Uygulayıcıların memnuniyet seviyesi hakkındaki nicel analizin kayda değer sonuçlarından biri Test GÜdümlü Geliştirme ile ilgilidir. Beş problem tanımı içinde, tercih edilen ile mevcut uygulama arasındaki farkın en kayda değer olduğu Test GÜdümlü Geliştirme dikkat çekmiştir.

Bu alandaki mevcut anketlerin nicel ve nitel soruların bir karışımını içerdiğinden bahsetmeliyiz. Nicel sorularda, katılımcılar tarafından kati nicel cevaplar (örneğin, sayı veya yüzdeler) raporlanmıştır. Diğer yandan, nitel anket sorularında, katılımcılara nitel doğal sorular yöneltilmiştir, örneğin, “yazılımınızı test ederken yaşadığınız zorluklar nelerdir?”. Genellikle, nitel anket bulgularının analiz, sentez ve raporlanması nicel anketlere göre daha zordur. Yukarıda bahsedilen anket havuzu içinde, üç anketin yukarıda bahsedilen nitel anket tekniklerine [4, 5, 16] güçlü bir vurgusu bulunmaktadır.

2004 ve 2009’da biz ve meslektaşlarımız tarafından Alberta çapında anket ile uygulamadaki yazılım testi ve kalite güvence tekniklerini inceleyen iki çalışma [7, 14] gerçekleştirilmiştir. Daha sonra 2010’da, Kanada çapında test uygulamalarının bir anketini gerçekleştirdik ve sonuçlar [18]’de raporlanmıştır.

Son fakat aynı derecede önemli olarak, Türkiye’de yazılım test profesyonellerini temsil eden ve destekleyen organ olan Turkish Testing Board (TTB), 2011’den beri Türkiye’de yazılım kalitesinin durumu hakkında yıllık bir anket gerçekleştirmekte ve 2013’teki son raporları [19]’da bulunabilmektedir. Bu bildiri boyunca mümkün olduğu ölçüde, anketimizin eğilim ve bulgularını TTB anketi ve ayrıca 2010’daki Kanada çapındaki anket ile karşılaştıracacağız.

3 Anket Hedef, Tasarım ve Uygulaması

3.1 Hedef ve Sorular

Bu anket çalışmasında kullandığımız yaklaşım Hedef, Soru, Ölçüt (GQM) metodolojisidir [26]. Çalışmanın hedefi Türkiye’deki yazılım mühendisliği uygulamalarını karakterize etmek; profesyonel testçiler tarafından kullanılan en son yazılım mühendisliği teknik, araç ve ölçütleri ve karşılaşılan zorlukları ortaya koymak; yazılım test endüstrisindeki son eğilimleri gözlemlemek ve güçlü ve zayıf alanları belirlemek ve akademi-endüstri işbirliğini teşvik etmek için hem Türkiye hem de dünyadaki test profesyonelleri ve araştırmacılarına faydalı olabilmektir.

Bu hedefe ulaşmak için, Kanada bağlamındaki geçmiş deneyimimizi temel alarak ve Software Engineering Body of Knowledge (SWEBOK)’ı kullanarak sistematik bir şekilde 54 soruluk bir çevrimiçi anket tasarladık. Soruların 9’u yazılım testi odaklıdır ve bu bildiriye yazılım testine odaklanan bulguları raporlamaktayız.

Çalışmada ilk olarak yazılım testi alanındaki en son konuları kapsarken aynı zamanda soru sayısında ekonomik olabilmek için benzer geçmiş anketler gözden geçirildi [1-19] ve taslak bir soru kümesi tasarlandı. Anket tasarımında, endüstriyel partnerlerden geribildirim almak, bazı çalışmalarda, örneğin [6], takip edilmiş bir yaklaşımdır. Bu çalışmada hazırlanan soru kümesi de birkaç endüstri uygulayıcısına dikkatli bir gözden geçirme için iletildi. Buradaki amaç, anketimizde kullanılan ter-

minolojinin hedef kitlenin makul bir oranı için tanıdık olmasını sağlamaktır. Maalesef, akademi ve endüstride kullanılan yazılım testi terminolojileri bazen farklı hatta yanıltıcı olabilmektedir, örneğin, sistem testi ve işlevsel test farklı alanlarda aynı veya farklı anlamlarda kullanılmaktadır. Endüstriyel uygulayıcılardan alınan kullanışlı geribildirimler anket soruları kümesinin son haline getirilmesinde kullanılmıştır.

Anketimizin soru kümesini tasarlarken kullandığımız kriterler; soruların endüstri açısından anlamlı olması ve en kullanışlı bilgileri elde etmektir. Teste odaklanan 9 sorunun tam listesi Tablo 2’de gösterilmiştir. Kısa tutmak için, Tablo 2’deki anket sorularının çoktan seçmeli şıklarını göstermiyoruz ancak bunlar bildiri içinde 4. Kısım’da tartışılacaktır. Anketin tam hali (soruların tam listesini ve sunulan çoktan seçmeli seçenekleri) çevrimiçi sunulmuştur [32].

Tablo 2. Soru Listesi

	Sorular (ve Ölçütler)
Katılımcıların Profili	<ul style="list-style-type: none"> • Mevcut pozisyonunuz/pozisyonlarınız ne(ler)dir? (birden fazla seçeneği seçebilirsiniz) • Bilgi Teknolojileri ve yazılım geliştirme endüstrisinde kaç yıllık iş tecrübesine sahipsiniz? • En yüksek akademik dereceniz hangisidir? • Üniversite derece(leri)nizi hangi alanda aldınız? (birden fazla seçeneği seçebilirsiniz) • Lütfen çalıştığımız şehri seçiniz?
Firmaların Profili	<ul style="list-style-type: none"> • Firmanız tarafından geliştirilen ürünlerin hedef sektörü nedir? • Firmanızın büyüklüğü nedir (çalışan sayısı)?
Teknik sorular	<ol style="list-style-type: none"> 1. Test Türleri/Seviyeleri: Aşağıdaki test aktivite türlerinin her birini hangi sıklıkta gerçekleştiriyorsunuz? 2. Mevcut ya da en son yazılım projenizde, test durumlarını (prosedürlerini) üretmek için ekibiniz hangi test tekniğini kullandı? 3. Test Otomasyonu: Genel olarak geçmiş projelerinizin tamamında, ne kadar manuel, ne kadar otomatik test yaptınız? 4. Test Ölçümleri: Test aktivitelerinizde aşağıdaki kod kapsama (coverage) ölçütlerinden hangilerini ölçüyorsunuz? 5. Projelerinizde aşağıdaki diğer test ve kalite ölçütlerinden hangilerini belirgin bir şekilde ölçüyorsunuz? 6. Test yönetimi: Lütfen mevcut veya en son projenizde testçilerin geliştiricilere oranını (testçi:geliştirici) belirtiniz. 7. Test aktivitelerinin bittiğine/tamamlandığına karar vermek için projelerinizde hangi kriterler kullanılır? 8. Tür ve Yazılım geliştirme yaşam döngüsünde testin fazlandırması açısından, test aktivitelerinizin türü nedir? 9. Yazılım testi sırasında kurumunuzda yürütülen uygulamalar

3.2 Anket Uygulama

Anket, SurveyMonkey¹ isimli bir çevrimiçi anket sağlama hizmeti üzerinde tasarlanmış ve sağlanmış ve 2013 yazında üç ay boyunca davetlilerin erişimine açıktı. Anketin araştırma etik onayı Mayıs 2013'te Orta Doğu Teknik Üniversitesi, Uygulamalı Etik Araştırma Merkezi'nden alınmış ve ardından anket uygulanmıştır. Katılımcılardan çevrimiçi olarak anketi tamamlamaları istenmiştir. Katılımcılar herhangi bir zamanda cevaplarını geri çekebilirler ve etik ilkelerine göre, araştırmacılar ankette sadece özet ve toplu bilgileri yayınlamayı kabul etmişlerdir.

Mümkün olduğunca çok cevap elde edebilmeyi sağlamak için, Türk yazılım şirketlerindeki partner/bağlantı ağıma e-mail davetleri gönderildi. Merkez ofis olup olmamasına bakılmadan, Türkiye'de yazılım geliştirme ofisi olan şirketler de bu gruba dâhil edildi. Geliştirme ofisleri (yazılım geliştirme ve test görevlerinin gerçekleştirildiği) çalışmamızın ilgi odağı olduğu için, şirketin Türkiye'de bir merkez ofisi olup olmaması ankete katılım için önemli bir faktör değildi. Aynı kriter önceki üç ankette [7, 14, 18] de uygulanmıştı. Anket verimiz Türkiye'den 163 katılımcının cevaplarını içermektedir.

4 Anket Sonuçları ve Bulguları

Bu kısımda, anket sonuçlarını raporlayacak ve bulgularını analiz edeceğiz:

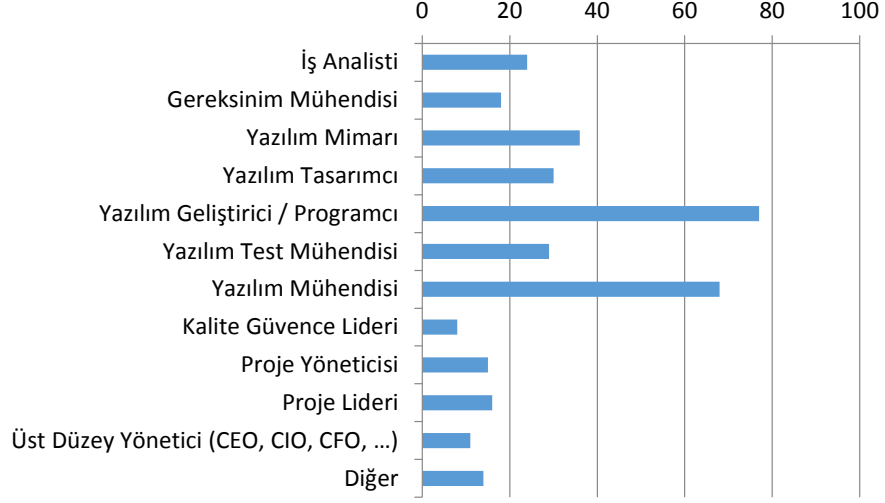
- Katılımcıların Profili (Kısım 4.1)
- Firmaların Profili (Kısım 4.2)
- Teknik sorular (Kısım 4.3)
- Çapraz Faktör Korelasyon Analizi (Kısım 4.4)

4.1 Katılımcıların Profilleri

Katılımcı Pozisyonları. Bu ankette katılımcıların pozisyonları Şekil 1'de gösterilmiştir. Bu çoktan seçmeli bir soru olduğu için, rol çakışmasının olabileceği göz önünde bulundurulmalıdır, örneğin, bir kişi aynı anda hem geliştirici hem de testçi olabilir. Şekilde gösterildiği üzere, katılımcıların çoğu “yazılım geliştirici” ve “yazılım mühendisidir”. Bu kısımda yer alan tüm figürlerdeki sayıları gösteren eksenler katılımcı sayısını belirtmektedir (aksi belirtilmediği takdirde).

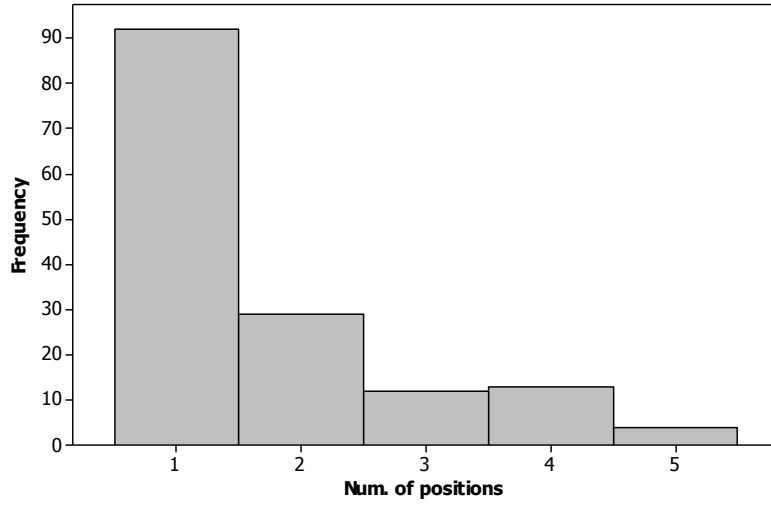
Şunu belirtmeliyiz ki, bilişim kalitesi çalışmalarında ortaya konulduğu şekilde (örneğin Garvin [27]'in çalışması), farklı pozisyonlardaki kişiler farklı hususların önemini farklı görür ve derecelendirirler ve genellikle yazılım testi ve ilgili süreçlerle ilgili değişken bakış açılarına sahiptirler.

¹ www.surveymonkey.com



Şekil 1. Katılımcı Pozisyonları

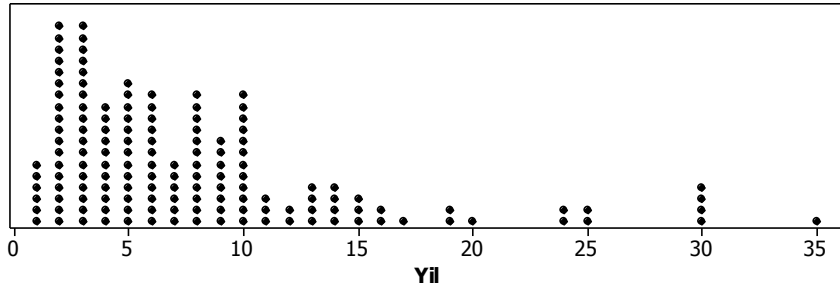
Bu soru çoktan seçmeli bir soru olduğu için, bir katılımcının birden fazla pozisyonu seçtiği durumların frekansını ölçtük. Sonuçlar Şekil 2’de gösterilmektedir. Katılımcıların çoğu sadece tek pozisyonda olduklarını bildirmiştir. Sürpriz bir şekilde, dört katılımcı aynı anda beş role atandığını bildirmiştir.



Şekil 2. Her katılımcının sahip oldukları pozisyon sayısı histogramı

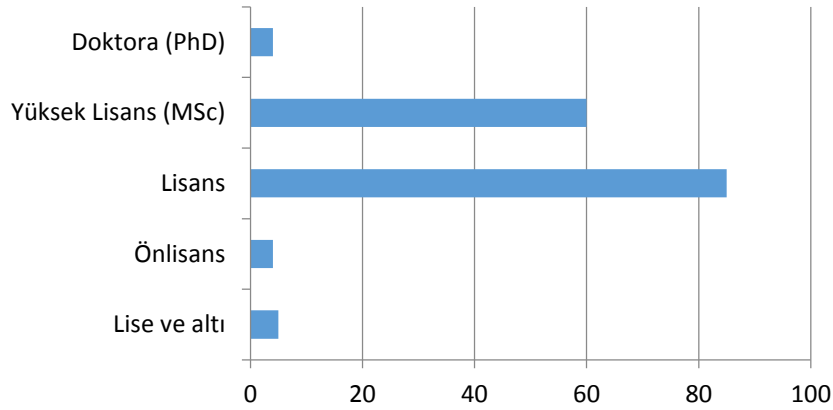
Yazılım Geliştirmedeki İş Tecrübesi (Soru 2 ve 3). Şekil 3, bir bağımsız değerler grafiği olarak, katılımcıların yazılım endüstrisindeki iş tecrübesinin dağılımını yıl bazında göstermektedir. Ortalama değer 8 yıldır.

Şekil 3'te görsel olarak da görebileceğimiz gibi, grafiğin çoğu sola doğru çarpıktır, bu katılımcı havuzunun “genç” eğilimli olduğunu ifade eder. Katılımcıların yaklaşık %40 ve %70'i, yazılım endüstrisinde sırasıyla 5 ve 10 yılın altında tecrübeye sahiptir. Katılımcıların sadece %19'u 10 yıl üstünde bir tecrübeye sahiptir. Çeşitli uzunluktaki işi tecrübesine sahip katılımcı popülasyonundaki çeşitliliğin, daha geniş bir hedef kitle tabanından değerli girdiler olarak ankete yardımcı olacağı için oluşan pozitif etkiyi görüyoruz.



Şekil 3. Katılımcıların bilgi teknolojileri/yazılım geliştirme ve yazılım testi/kalite güvence 'deki iş tecrübesi

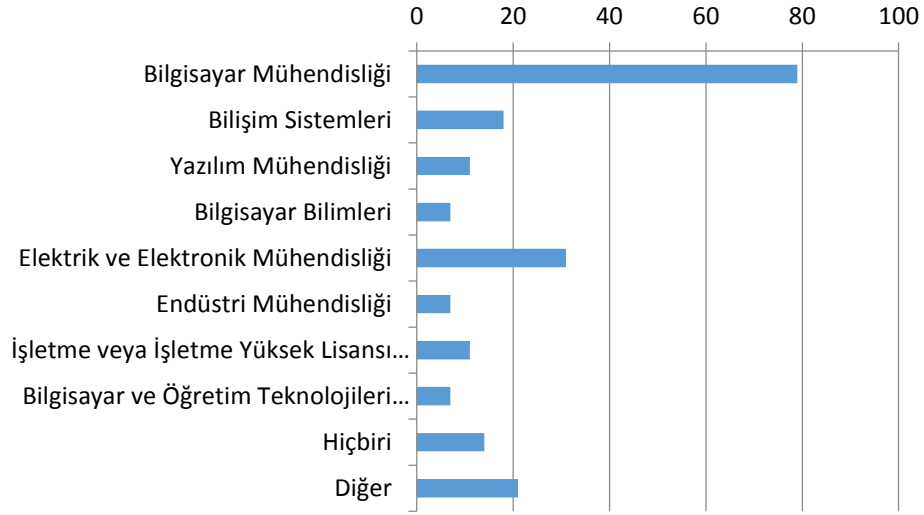
En Yüksek Akademik Dereceler (Soru 5). Katılımcıların eğitim geçmişlerini anlamak için, katılımcılara en yüksek akademik dereceleri sorulmuştur. Şekil 4'te görüldüğü gibi, sonuç göstermektedir ki katılımcıların %52.14 ve %36.8'i sırasıyla Lisans ve Yüksek Lisans derecesine sahiptir. 5, 4, ve 4 katılımcı ise sırasıyla Lise ve altı, Önlisans ve Doktora (PhD) derecelerine sahiptir.



Şekil 4. Katılımcıların en yüksek akademik dereceleri

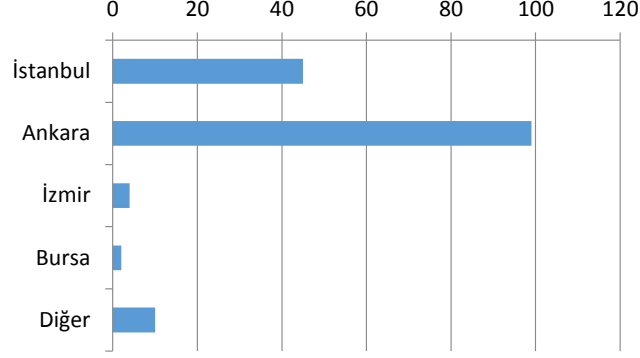
Akademik Branş (Soru 6). Önceki sorunun devam sorusu olarak, katılımcıların eğitimsel yetenek kümesini anlamak için, katılımcılardan akademik branşlarını (örneğin, yazılım mühendisliği ve bilgisayar mühendisliği) bildirmeleri istenmiştir. Sonuçlar Şekil 5’te görülmektedir.

İlk üç sırada Bilgisayar Mühendisliği (katılımcıların %48.4’i), Elektrik ve Elektronik Mühendisliği (%19.0) ve “Diğer” (%11.0) yer almaktadır. Diğer seçeneği Matematik, Makina, İktisat ve Fizik branşlarını içermiştir. Bu altyapılardan gelen birçok testçi ile kişisel görüşmelerimize göre, bu testçiler takımları için çok değerlidirler çünkü kendi iş alanlarında zengin bir bilgiye sahiptirler ve test sürecinde iş-alanı-türü hataların (örneğin, bankacılık uygulamaları) bulunmasında yararlı olabilirler.



Şekil 5. Katılımcıların Akademik Branşları

Katılımcıların Bölgesel Dağılımı (Soru 9). Daha önceki Kanada çapındaki anketimize [18] benzer olarak, anketimizde her katılımcının ikamet ettiği şehri kayıt etmek için bir soru mevcuttu. Hedefimiz mümkün olduğunca Türkiye’de birçok şehre ulaşmaktı ve veri setinde, bir yazılım endüstrisi olan her bölgenin temsil edilmesini sağlamaktı. Katılımcıların bölgesel dağılımı Şekil 6’da gösterilmiştir. Şaşırtıcı olmayan bir şekilde, Ankara ve sonrasında İstanbul katılımcıların en çok yerleştiği yerlerdi. Yazarların yerleşim yeri (Ankara) dolayısıyla ve bağlantılarımızın çoğunun yerel olması nedeniyle, katılımcıların %60’ı Ankara’dandır. Diğer şehirler Afyon, Balıkesir, Eskişehir, Gebze, Kocaeli ve Samsun’u içermektedir.

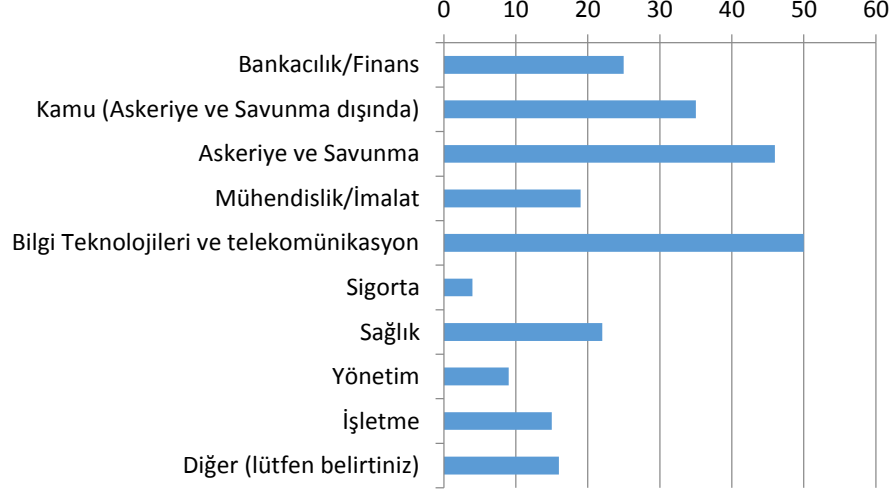


Şekil 6. Katılımcıların Bölgesel Dağılımı

4.2 Firmaların Profili

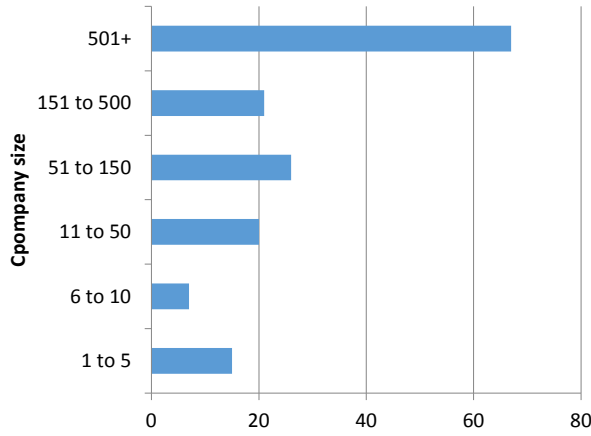
Firmaların geliştirdiği ürünlerin hedef sektörü. Bir sonraki soru, yazılım organizasyonları tarafından geliştirilen ürünlerin türleri hakkındaydı (Şekil 7). Önceki anketlerimizde [14, 18] aldığımız cevaplar temelinde, on muhtemel cevap ankette sunulmuştur. Çoğu katılımcının, şirketlerinin ürettiği ürünlerin hedef sektörünü bilgi teknolojileri ve telekomünikasyon olarak kategorize ettikleri görüyoruz. İkinci ve üçüncü sırada askeriye ve savunma, ve kamu (askeriye ve savunma dışında) yer almaktadır. Katılımcılar tarafından bildirilen diğer kategorisinde gıda, e-ticaret, otomotiv, tekstil, oyun ve oyun teknolojileri yer almaktadır.

Yukarıdaki organizasyon türlerinin her birinin ürettiği yazılım sistemlerinin kalitesi önemli olsa da, sıklıkla uyguladıkları test aktivitelerinde ufak farklar bulunmaktadır, örneğin, askeriye ve savunma endüstrisi için üretilen yazılımların diğer kategorilerle kıyaslandığında daha sıkı kalite gereksinimleri vardır.



Şekil 7. Anket yapılan yazılım organizasyonlarının geliştirdikleri ürün türleri

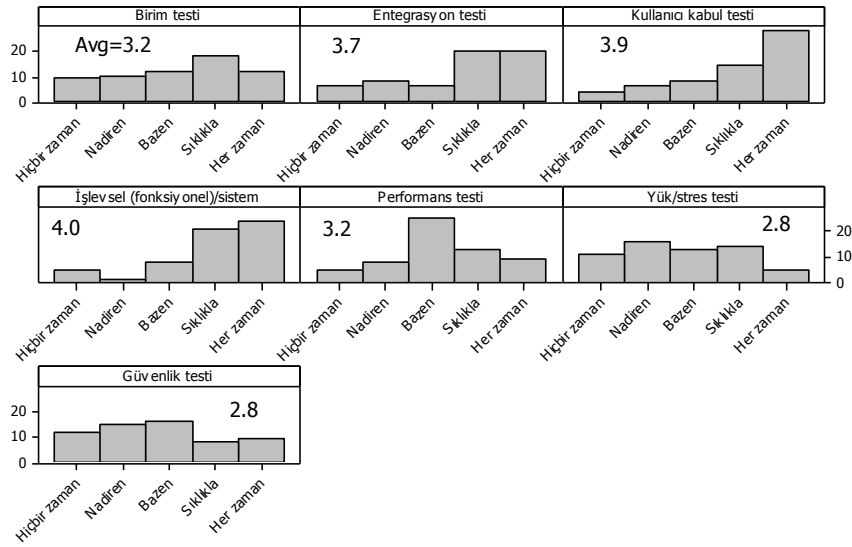
Firmaların büyüklüğü. Anketimizde, test aktivite türleri ile çalışan sayılarını ilişkilendirebilmek için şirket büyüklüklerini bilmek de önemliydi. Şekil 8’de çalışan sayısı bakımından şirket büyüklüklerinin dağılımı gösterilmiştir. 500’den fazla çalışana sahip şirketlerden 60’dan fazla katılımcının anketimizi tamamladığını görmek ilginçtir. Diğer çalışan sayısı aralıklarından katılımcıların iyi bir karışımı anket havuzumuzda mevcuttur. Bu analizimizin, şirket büyüklüğü açısından daha büyük bir girdi yelpazesini kapsama olanak tanyacaktır.



Şekil 8. Firma büyüklüğü

4.3 Teknik Sorular

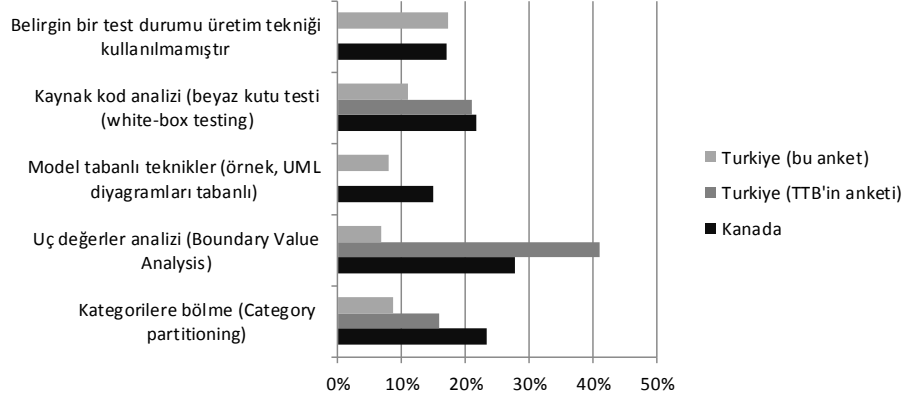
Test Türleri/Seviyeleri (Soru 1). İlk teknik soru test aktivite türleri hakkındaydı. Cevaplar 1 (hiçbir zaman)...5 (her zaman) arası bir Likert ölçeği kullanılarak toplandı. Oyların ortalama değerleri Şekil 9'da gösterilmiştir. İşlevsel (fonksiyonel)/sistem testi, Kullanıcı kabul testi, ve Entegrasyon testi 5 üzerinde sırasıyla 4, 3.9, ve 5 ortalamalarla en sık kullanılan üç test yaklaşımıdır. Stres testi en az kullanılan tekniktir.



Şekil 9. Test türleri/seviyeleri

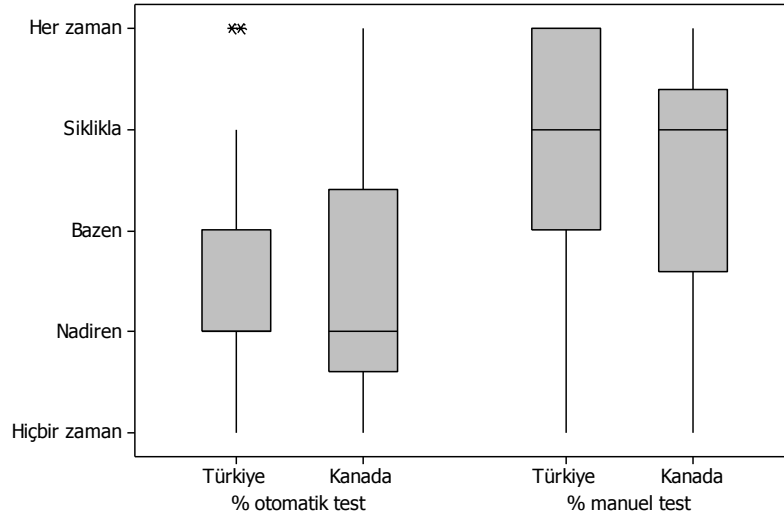
Test prosedürlerini üretmek için kullanılan teknikler (Soru 2). Test durumu tasarımı yazılım testinde önemli bir aktivitedir. Yakın tarihli Kanada çapındaki anketimize [18] benzer olarak, ankette test durumlarını tasarlamak için kullanılan teknikleri sorduk. İki ülkenin sonuçları Şekil 10'da gösterilmiş ve karşılaştırılmıştır. Yatay eksenindeki değerler katılımcı havuzundaki yüzdelerdir. Anketimize göre, Türk testçilerin %10'undan azı dört formal yaklaşımdan herhangi birini kullanırken, Kanada'da bu sayıların biraz daha iyi olduğunu (%15 ve %27 arasında) gözlemleyebiliyoruz. Tesadüfen, neredeyse eşit olarak, her iki ülkeden katılımcıların yaklaşık %17'si projelerinde belirgin bir test durumu üretim tekniği kullanmadıklarından bahsetmişlerdir.

TTB anketi de Şekil 10'da yer verilen test durumu tasarım yaklaşımlarından üçü hakkında ölçümler sunmaktadır. İlginç bir şekilde, üç yaklaşım için TTB anketinde yer alan oranlar bizim anketimizdekilerden yüksektir. Bunun nedeni belki de, TTB anketi hedef kitesinin, TTB organizasyonu üyesi olan ve formal test durumu tasarım yaklaşımlarını kullanıyor olma ihtimali daha yüksek ihtisaslaşmış test uzmanları olmasıdır.



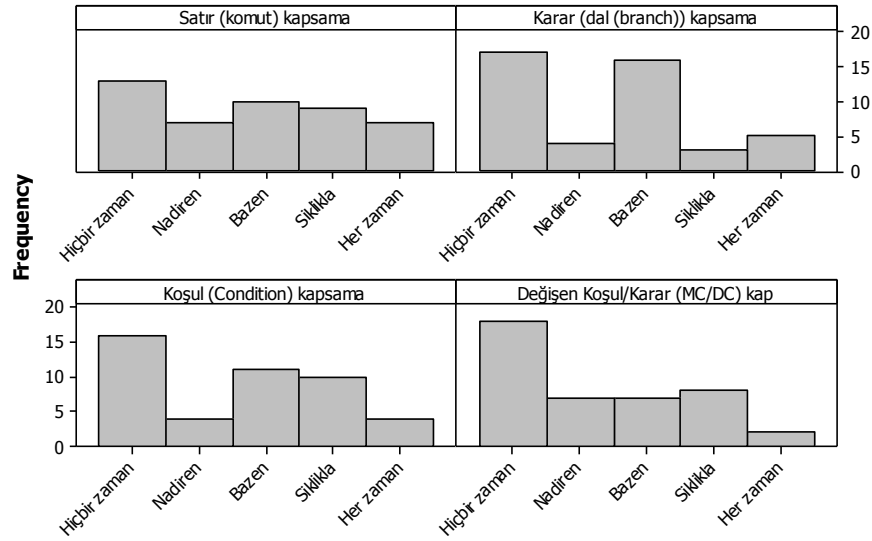
Şekil 10. Test prosedürlerini üretmek için kullanılan teknikler

Manuel Test ve Test Otomasyonu (Soru 3). Bu soruda manuel ve otomatik testlerin gerçekleştirilme frekansını ölçülmesi amaçlanmıştır. Kanada anketinde de bu veriler mevcut olduğundan, her iki ülkenin sonuçlarını Şekil 11’de karşılaştırmaktayız. Her iki ülkede de görüyoruz ki, genel olarak her iki test yaklaşımının kullanımında geniş bir yelpaze (hiçbir zaman ile her zaman arasında bir yerlerde) vardır. Bu durum, bu bağlamda farklı katılımcıların çok farklı uygulamaları olduğunu belirtir. Örneğin, bazıları yoğun olarak test otomasyonu uygularken, diğerleri manuel testi tercih etmektedir.



Şekil 11. Manuel Test ve Test Otomasyonu

Test kod kapsama (coverage) ölçütleri (Soru 4). Kod kapsama (coverage) ölçütleri hem test durumu tasarımı hem de test etkinliği ölçümü aşamalarında önemli araçlardır. Şekil 12'deki dört histogram, sıklıkla kullanılan dört metriğin kullanımıyla ilgili anket sonuçlarını resmetmektedir. Dört histogramın tamamının genel olarak sola doğru yatık olması, Türk yazılım endüstrisinde kapsam ölçütlerinin genel olarak popülerliğinin düşük olduğu anlamına gelmektedir.

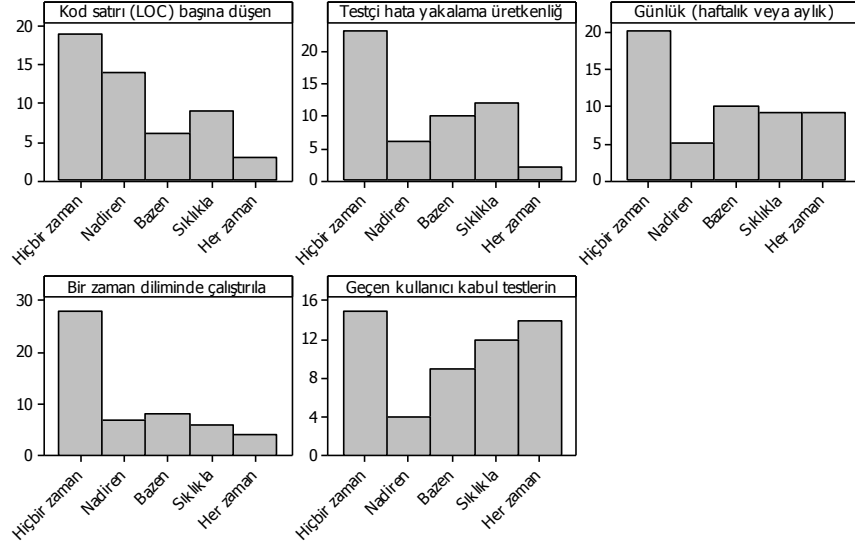


Şekil 12. Kod kapsama (coverage) ölçütleri

Diğer test ve kalite ölçütleri (Soru 5). Testçiler tarafından çoğunlukla kullanılan test ve kalite ölçüt türleri vardır [18]. Bu ölçütlerden bazılarının kullanımını inceledik ve sonuçları Şekil 13'de sunmaktayız. Seçilen beş ölçüt aşağıda verilmiştir:

- Kod satırı (LOC) başına düşen (ortalama) hata sayısı
- Testçi hata yakalama üretkenliği (her bir testçi tarafından günlük bulunan hatalar)
- Günlük (haftalık veya aylık) bulunan toplam hata sayısı
- Bir zaman diliminde çalıştırılan test durumu sayısı
- Geçen kullanıcı kabul testlerinin sayısı

Yine, histogramların sola doğru yatık olması, bu ölçütlerin genel olarak düşük kullanımını göstermektedir.

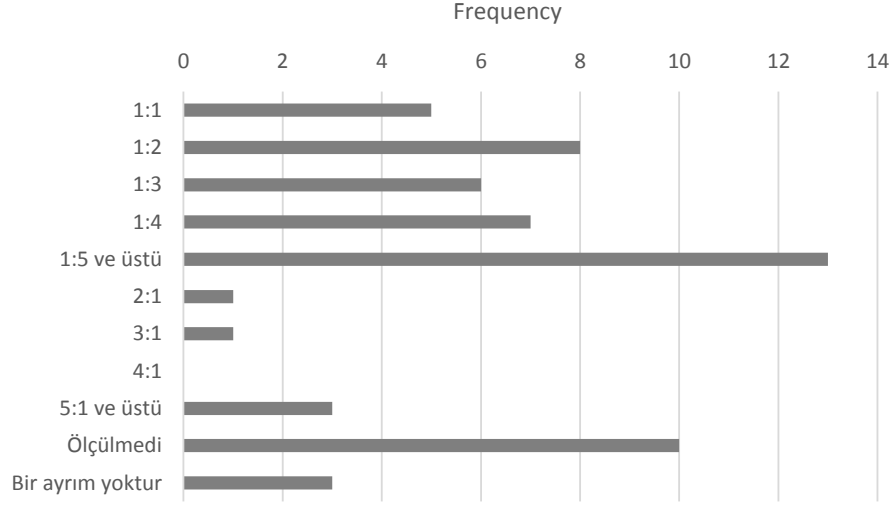


Şekil 13. Diğer test ve kalite ölçütleri

Testçilerin geliştiricilere oranı (Soru 6). Bu soru değişik takımlarda testçilerin geliştiricilere oranını bulmak için tasarlanmıştır. Sonuçlar Şekil 14’te gösterilmiştir. Çoğu şirkette, 1:2 ile 1:5 ve üstü arası değişen oranlarla testçiler geliştiricilerden sayıca azdır. Şekle göre, testçi:geliştirici oranı 1:5’ten 1:2’ye çıktıkça frekans düşmektedir. Ayrıca, bazı şirketler ya bu iki rol arasında bir ayırım yapmamakta ya da bu metriği ölçmemektedir.

Testçi geliştirici oranı (testçi:geliştirici) günümüzde yazılım endüstrisinde aktif olarak tartışılan bir konudur. James Whittaker, bir test mühendisliği yöneticisi olarak iki öncü yazılım şirketini testçi:geliştirici oranı açısından karşılaştırmıştır. Seçilen iki büyük öncü teknoloji şirketinin oranlarının 1:1 ve 1:3 arasında değiştiğine dikkat çekmektedir. Yazısını şu şekilde bağlamaktadır: “Test yöneticileri optimum noktayı bulmaya çalışmalıdırlar” [28]. Iberle ve Bartlett’in bir şirketin bu problemi nasıl ele alması gerektiği, örneğin uygun testçi geliştirici oranını kestirmek ile ilgili çevrimiçi bir bildirisi vardır [29]. Belli bir şirket veya proje için gerçekçi bir testçi geliştirici oranı elde eden ilginç bir model ve yaklaşım sunarlar. Bizim sonuçlarımızda, görünen o ki yöneticiler tarafından seçilen “optimum” 1:2 ile 1:5 arasındadır.

Testçi geliştirici oranı hususunda başka çalışmalar da vardır. Cusumano ve Yoffie [30] 1998’de tarayıcı geliştirmede Microsoft ve Netscape rekabetini bildirmiştir. Netscape’in daha az testçi çalıştırdığını ve sonra bu kararın geliştirme sürecinin diğer yönlerine etkilerini (örneğin, yayım süresi) incelediklerini bildirmişlerdir.

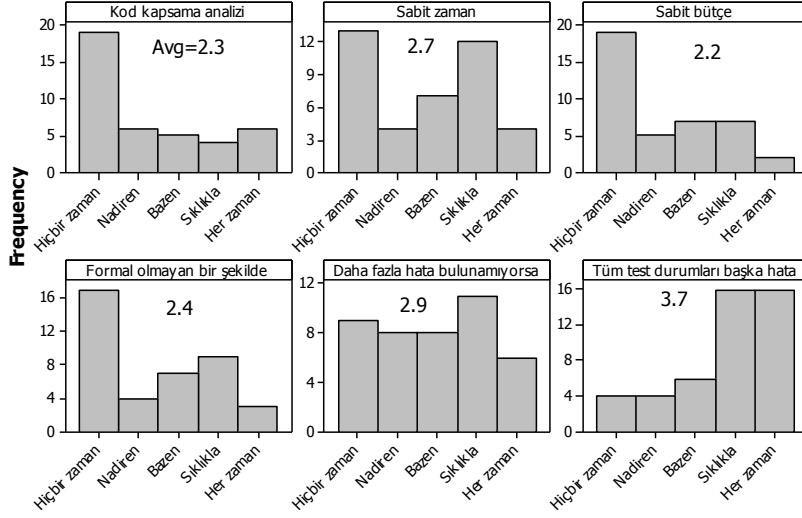


Şekil 14. Testçilerin geliştiricilere oranı

Test aktivitelerinin bittiğine karar vermek için kullanılan kriterler (Soru 7). Yazılım testi literatüründe, testi sonlandırmak için birçok kriter önerilmiştir. Bu soru için aşağıdaki altı kriteri listeledik. Bunun yanı sıra katılımcılar kendi kriterlerini de ekleyebilmektedirler.

- Kod kapsama analizi (örnek, %x komut kapsama oranına ulaştığınız zaman)
- Sabit zaman
- Sabit bütçe
- Formal olmayan bir şekilde
- Daha fazla hata bulunamıyorsa (güvenilirlik büyüme/doyma modelleri)
- Tüm test durumları başka hata bulmadan çalıştırılıyorsa

Şekil 15 sonuçları göstermektedir. Oyların ortalama değerleri de Şekil 15'te, 1 "hiçbir zaman" ve 5 "her zaman" anlamına gelecek şekilde gösterilmiştir. Test aktivitelerini sonlandırmak için en çok kullanılan kriter son kriter gibi görünmektedir. Türk endüstrisindeki düşük kabullerini gösterecek şekilde, kod kapsama ölçütleri pek popüler gözükmemektedir.



Şekil 15. Test aktivitelerinin bitiğine karar vermek için kullanılan kriterler

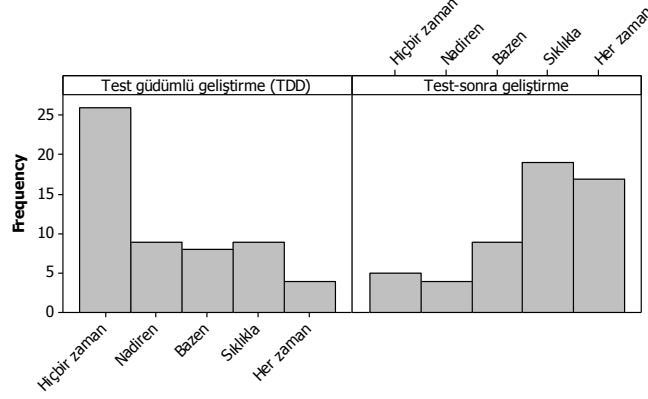
Turkish Testing Board (TTB)'un 2013 anket raporu [19] da test çıkış kriteri ile ilgili bir soru içermektedir. Bu rapordaki seçeneklerden ikisi, Şekil 15'deki seçeneklerimiz ile aynıdır; bunlar zaman ve bütçe kısıtlarıdır. TTB raporundaki ölçüler yüzde olarak verilmiştir. Örneğin, katılımcıların %11'i test çıkış kriterlerinin bütçe kısıtları olduğunu bildirmiştir. Sonuçlarımızı TTB anketininkilerle karşılaştırabilmek için, Likert verimizi yüzdelerle değerlere çevirdik. Şekil 15'teki sabit bütçe veri kümesinin ortalama değeri 2.2'dir, yani bunun yüzdelerle değeri $(2.2-1)/5=\%22$ olur. Benzer şekilde, zaman kısıtı serisinin yüzdelerle değeri $(2.7-1)/5=\%34$ olur. Sonuçlarımızın TTB anketi sonuçları ile kıyaslaması Tablo 3'tedir. Her iki ankette de, zaman kısıtları bütçe kısıtlarına göre daha çok kullanılan bir kriter olarak görülmektedir.

Tablo 3. Test aktivitelerinin bitiğine karar vermek için kullanılan kriterler: sonuçlarımızın TTB anketi [19] sonuçları ile karşılaştırılması

	Anketimiz	TTB anketi [19]
Zaman kısıtları	%34	%50.8
Bütçe kısıtları	%22	%11

Yazılım geliştirme yaşam döngüsündeki test aktivitelerinin türü (Soru 8). Katılımcılara yazılım geliştirme yaşam döngüsü sırasındaki test eforlarının türleri ve hangi fazlarda gerçekleştiği sorulmuştur. Sonuçlar Şekil 16'da yer almaktadır. Test-sonra Geliştirme yaklaşımlarının (geliştirmeden sonra test etme) hala Test Güdümlü (önce test) Geliştirmeden çok daha popüler olduğunu görebiliyoruz. Bu sonuç

göstermektedir ki, geleneksel test-sonra geliştirme yaklaşımı Türk şirketleri arasında hala baskın bir yöntemdir.

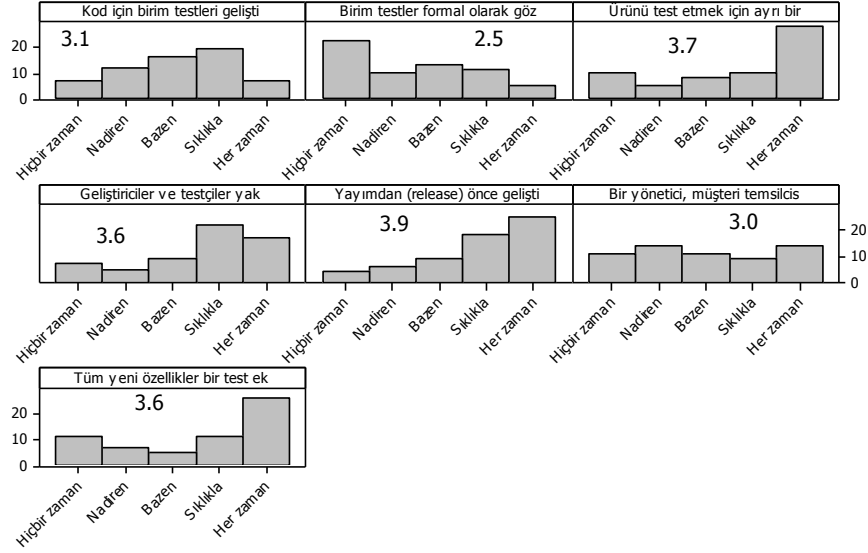


Şekil 16. Yazılım geliştirme yaşam döngüsündeki test aktivitelerinin türü

Test sırasında yürütülen uygulamalar (Soru 9). Son ama bir o kadar önemli olarak, anket sorularımızın test kategorisinde, katılımcılara test sırasında yürütülen uygulamalar hakkında sorular sorduk. Şekil 17 cevapları ve oyların ortalama değerleri göstermektedir. Bu soru için aşağıdaki yedi seçeneği sunduk.

- Kod için birim testleri geliştiriyoruz.
- Birim testler formal olarak gözden geçiriliyor.
- Ürünü test etmek için ayrı bir ekibimiz var.
- Geliştiriciler ve testçiler yakın olarak birlikte çalışmaktadır.
- Yayım (release) önce geliştirici ürünü test etmektedir.
- Bir yönetici, müşteri temsilcisi veya destek çalışanı ürünün test edilmesine yardımcı olmaktadır.
- Tüm yeni özellikler bir test ekibi tarafından bağımsız olarak test edilmektedir.

Listedeki en çok ve en az popüler aktiviteler sırasıyla “Yayımdan (release) önce geliştirici ürünü test etmektedir”, ve “Birim testler formal olarak gözden geçiriliyor” olarak ortaya çıktı.



Şekil 17. Test sırasında yürütülen uygulamalar

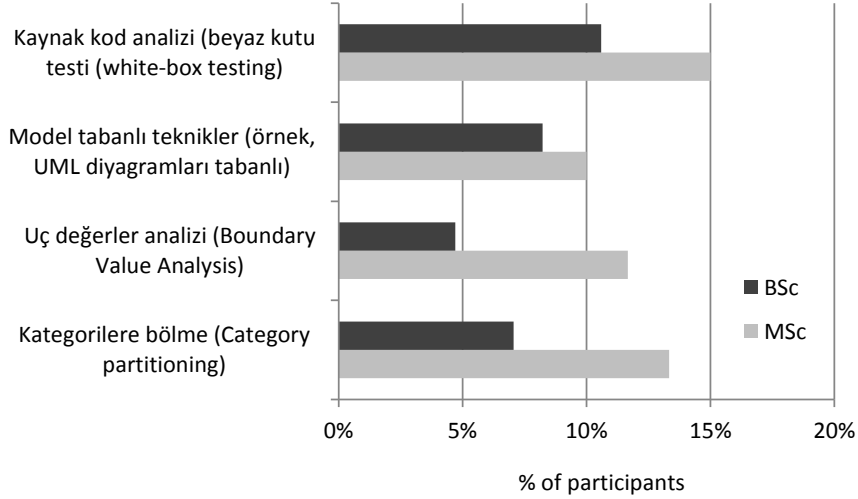
4.4 Çapraz Faktör Korelasyon Analizi

Bireysel anket sorularının sonuçlarını analiz ettikten sonra, değişik faktörler arasında herhangi bir ilişki olup olmadığını değerlendirmeyi, yani çapraz faktör analizi gerçekleştirmeyi hedefledik. Mevcut sorular ve veriler temelinde, daha sonra analizlerini sunduğumuz, aşağıdaki hipotezleri belirledik:

- Test durumu tasarımı için kullanılan test tekniklerinde katılımcıların akademik derecesinin herhangi bir etkisi var mı? Ya da, daha yüksek dereceye sahip mühendisler daha uygun test durumu tasarım teknikleri mi kullanıyorlar?
- Katılımcıların iş tecrübesi miktarı ve test otomasyonu kullanımı: yıl olarak daha deneyimli olan mühendisler, test otomasyonunu kullanmaya daha çok mu yatkın?
- Katılımcıların iş tecrübesi ve kod kapsama (coverage) ölçütlerinin kullanımı: yıl olarak daha deneyimli olan mühendisler, kod kapsama (coverage) ölçütlerini kullanmaya daha çok mu yatkın?
- Şirket profili ve test durumu tasarım teknikleri: kullanılan test durumu tasarım tekniklerinin türü üzerinde şirket profilinin bir etkisi var mı?
- Şirket profili ve test kod kapsama (coverage): Şirket profilinin kod kapsama (coverage) ölçütlerinin kullanılma seviyesine bir etkisi var mı?
- Şirket profili, ve manuel test ve test otomasyonu: Şirket profilinin test otomasyonunun kullanım seviyesi üstünde bir etkisi var mı?

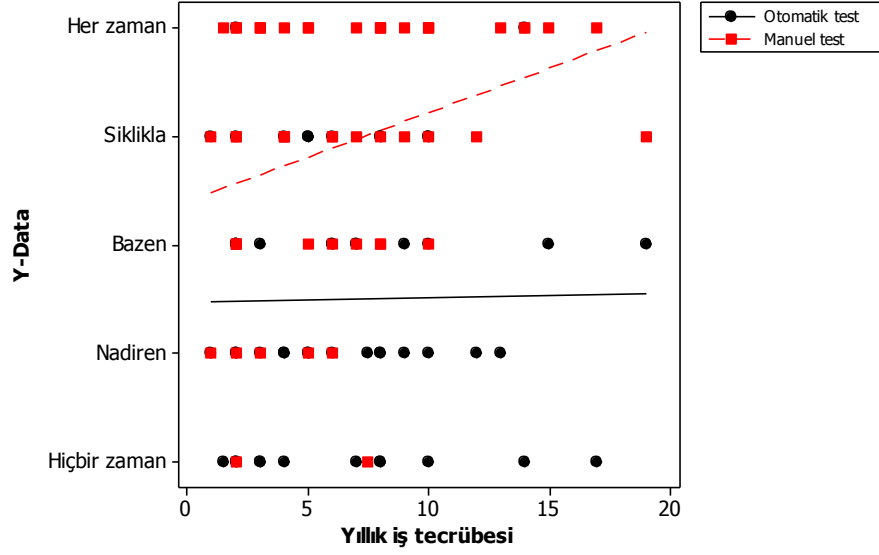
Akademik derece ve test durumu tasarımı için kullanılan teknikler. Yukarıda tartışıldığı gibi, daha yüksek dereceli mühendislerin daha olgun test durumu tasarımı teknikleri kullanacağı hipotezi kurduk. Bu hipotezi değerlendirmek için, iki farklı katılımcı kümesi (MSc ve BSc sahibi olanlar) için test durumu tasarımı tekniklerinin türleri Şekil 18’de gösterilmiştir. Diğer üç derece için (Lise ve altı, Önlisans ve Doktora), güvenilir bir veri analizi sağlamak için yeterli verimiz yoktu.

Yüksek lisans dereceli katılımcıların %15’i beyaz kutu (white box) testi kullandığını bildirirken, lisans derecelilerin sadece %11’inin bu teknikleri kullandığını görmekteyiz. İki havuz arasındaki fark %2 (model-tabanlı test tasarımı) ile %7 (uç değerler analizinde) arasında değişmektedir. Böylece, daha yüksek akademik derecenin daha olgun test durumu tasarımı teknikleri kullanımına hafif bir etkisi vardır diyebiliriz. Bu beklenen bir durumdur, çünkü daha yüksek akademik eğitimde mühendisler daha fazla test kavramı ile karşı karşıya kalırlar.



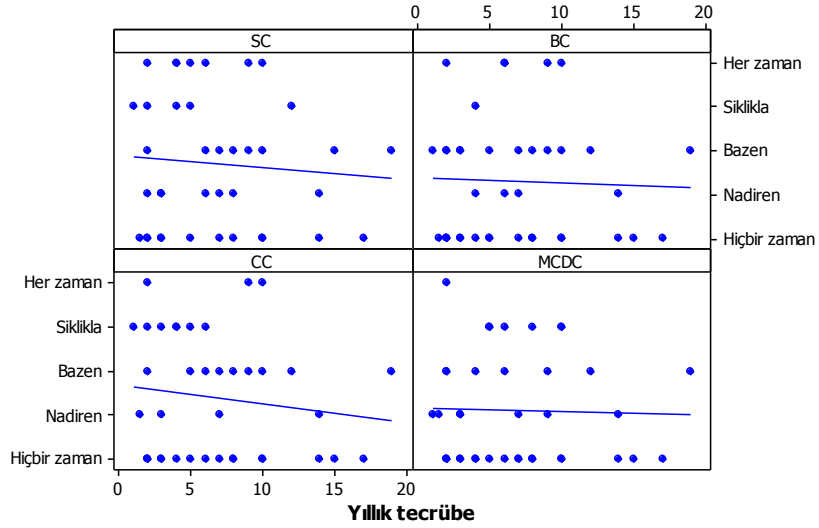
Şekil 18. Akademik derece ve test durumu tasarımı için kullanılan teknikler

Yıllık iş tecrübesi ve Test Otomasyonu. Yıllar boyunca projelerinde manuel testin zorlukları ve maliyetini tecrübe ettikleri için, daha fazla iş tecrübesine sahip mühendisleri test otomasyonunu kullanmaya daha meyilli olmaları beklenir. Bu hipotezi değerlendirmek için, katılımcılar tarafından bildirilen iş tecrübesi ve manuel veya otomatik test tekniklerinin kullanım frekansı verilerini ilişkilendirdik. Şekil 19 sonuçları bir serpm diyagramı (x-y) ile göstermektedir. Regresyon çizgileri de gösterilmektedir. Gözlemleyebileceğimiz gibi, yıllık tecrübe arttıkça, manuel test kullanımını test otomasyonu kullanımına göre biraz daha dik bir eğilimde artmaktadır. Ankara’da birçok yazılım mühendisi ile yaptığımız gayri resmi görüşmelere göre, bu eğilim çeşitli nedenlere bağlı olabilir. Örneğin, (1) belki de tecrübeli mühendisler bile henüz etkin test otomasyonu yaklaşımları ile karşılaşmamış/onları kullanmamış ve böylece projelerine uyarlamamıştır, (2) test otomasyonunun ön maliyeti nedeniyle, manuel test hala birçok takım için çekici ve tercih edilebilir olabilir.



Şekil 19. Yıllık iş tecrübesi ve Test Otomasyonu

Yıllık tecrübe ve Test kod kapsama (coverage). Bir sonraki çapraz faktör korelasyon analizi, daha tecrübeli mühendislerin kod kapsama (coverage) ölçütlerini kullanmaya daha yatkın olup olmadıklarını değerlendirmek için yapıldı. Önceki analizlere benzer olarak, veriyi yıllık tecrübe ve dört kapsama metriğinin her birinin kullanımına ayırdık: (1) SC: Satır (komut) kapsama, (2) BC: Karar (branch) kapsama, (3) CC: Koşul (Condition) kapsama, ve (4) MC/DC: Değişen Koşul/Karar kapsama. Şekil 20 sonuçları bir serpm diyagramı (x-y) ile göstermektedir. Regresyon çizgileri de gösterilmektedir. Gözlemlediğimiz kadarıyla, yıl sayısı arttıkça, sürpriz bir şekilde, kapsama ölçütlerinin kullanımı yavaşça azalmaktadır. Ankara'daki birçok yazılım mühendisi ile yaptığımız gayri resmi görüşmelere göre, bu eğilimin arkasında çeşitli nedenler olabilir. Örneğin, tecrübeli test mühendisleri eğitimini kapsama ölçütlerinin test araçlarında ortaya çıkmasından yıllar önce tamamlamış olabilir, ve böylece bu ölçütlerin kullanımı hakkında aşinalığı olmayabilir.



Şekil 20. Yıllık tecrübe ve Test kod kapsama (coverage)

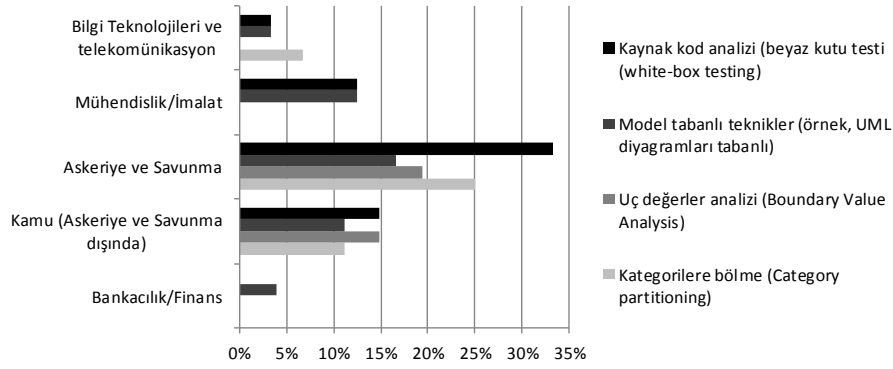
Şekil 20'deli veri seti için Pearson korelasyon ölçüşünü de hesapladık. Sonuçlar Tablo 4'de gösterilmektedir. Yıllık tecrübe ve dört kapsama metriğinin kullanımı arasındaki marjinal negatif korelasyon ölçüsü dikkat çekicidir. Kapsama ölçütlerinin kullanımı arasında pozitif bir ilişki olduğunu gözlemlemek ilginçtir, buna göre örneğin bir mühendis Karar Kapsama (BC) kullanıyorsa, Koşul Kapsama (CC) da kullanması çok olasıdır (Pearson korelasyon=0.71).

Tablo 4. Yıllık tecrübe ve test kod kapsama (coverage) kullanımı arasındaki korelasyon.²

	Yıl	SC	BC	CC
	-0.08 (Pearson korelasyon)			
SC	0.60 (p-value)			
BC	-0.04 0.80	0.56 0.00		
CC	-0.14 0.39	0.54 0.00	0.71 0.00	
MCDC	-0.02 0.86	0.36 0.016	0.52 0.00	0.74 0.00

² SC: Satir (komut) kapsama
 BC: Karar (branch) kapsama
 CC: Koşul (Condition) kapsama
 MC/DC: Değişen Koşul/Karar kapsama

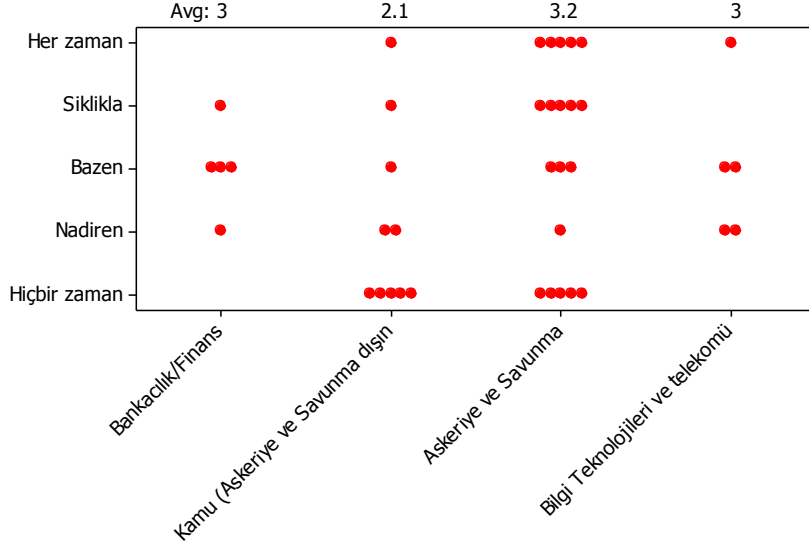
Şirket Profili ve test durumu tasarım teknikleri. Bir sonraki hipotez, şirket profilinin şirkette kullanılan test durumu tasarım tekniklerinin türü üstünde etkisi olup olmadığıdır. Kısım 4.2’de firmaların ürettiği ürünlerin hedef endüstri türleri le ilgili bir soru olduğunu hatırlayalım. Anket verisini, endüstri türleri ve test durumu tasarım tekniklerinin her birine göre böldük. Şekil 21 sonuçları göstermektedir. Yatay (x) eksen her bir test durumu tasarım tekniğinin kullanımını bildiren, endüstri türlerinden katılımcı yüzdelerini göstermektedir. Örneğin, Askeriye ve Savunma endüstrisi için ürün geliştiren katılımcıların %33 ve %25’i beyaz kutu (White box) testi ve kategorilere bölme kullanmaktadır. Diğer dört endüstri türü (sigorta, sağlık, yönetim, ve işletme) Şekil 21’de dâhil edilmemiştir çünkü diğer dört endüstri türünde bir istatistikî temsil için yeterli



Şekil 21. Şirket Profili ve test durumu tasarım teknikleri

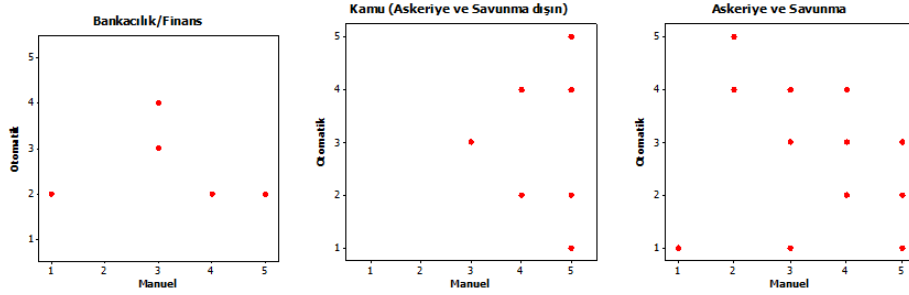
Şirket Profili ve Test kod kapsama (coverage). Bir sonraki hipotez şirket profilinin projelerde kullanılan test kod kapsama (coverage) türüne bir etkisi olup olmadığıdır. Bir bireysel değer grafiği şeklinde sonuçlar Şekil 22’de yer almaktadır. Her serinin ortalama değeri de gösterilmiştir (Likert ölçeğini hatırlayın: 1=Hiçbir zaman ... 5=Her zaman).

Askeriye ve Savunma sektörü için ürün geliştiren katılımcıların, diğer üç sektörle karşılaştırıldığında, daha fazla kapsama metriği kullandıklarını bildirdiklerini görebiliyoruz.



Şekil 22. Şirket Profili ve Test kod kapsama (coverage)

Şirket Profili, ve Manuel Test ve Test Otomasyonu. Sıradaki hipotez, şirket profilinin manuel veya otomatik test kullanım frekansına etkisinin olup olmadığıdır. Şekil 23 cevapları serpmme diyagramı olarak göstermektedir. Endüstri türlerinden sadece üçünün sunulmak için yeterli geçerli verileri mevcuttur. Her grafikte yer alan cevaplarda geniş bir sapma olduğunu gözlemleyebiliyoruz., ancak genel bir ifade ile Askeriye ve Savunma endüstrisinin yüksek bir test otomasyonu uygulamasına sahip olduğunu söyleyebiliriz.



Şekil 23. Şirket Profili, ve Manuel Test ve Test Otomasyonu

5 Tartışma

Bulgularımızın bir özeti Kısım 5.1’de tartışılmaktadır. Kısım 5.2 öğrenilen dersleri ele almaktadır. Çalışmamızın geçerliliğine olası tehditler ve bunları azaltmak ya da hafifletmek için attığımız adımlar Kısım 5.3’te ele alınmıştır.

5.1 Bulguların Özeti

Anketimiz Türk şirketlerinde çalışan 163 yazılım mühendisinden cevaplar toplamıştır. Demografik verilerimize göre, yazılım geliştirilen Türkiye’deki başlıca şehirlerden (örneğin, Ankara, İstanbul, İzmir, Eskişehir, Gebze, Kocaeli ve Samsun) yazılım şirketleri anketimize katılmışlardır.

Şirketler tarafından geliştirilen ürünlerin hedef endüstri türleri bankacılık/finans’tan, askeriye ve savunma, kamu (askeriye ve savunma dışında) ve mühendislik/imalat’a kadar çeşitlenmiştir. Anket verimizde farklı proje ve şirket profillerinden (4.1 ve 4.2 no’lu kısımlar) iyi bir karışım vardır ve bu durum bulgularımızın belli geliştirme şirketi türlerinden bağımsız olmasına yardımcı olmuştur.

Anket Türk yazılım endüstrisinde ilginç anlayış ve eğilimleri açığa çıkarmıştır. Vurgulanan bazı bulgular aşağıda listelenmiştir:

- Soru 1-İşlevsel (fonksiyonel)/sistem testi, Kullanıcı kabul testi, ve Entegrasyon testi 5 üstünden sırasıyla 4, 3.9, ve 3.7 ortalamaları ile en geniş kullanılan üç test yaklaşımıdır. Stres testi en az kullanılan tekniktir.
- Soru 2-Türk testçilerin %10’undan azı formal yaklaşımlardan herhangi birini kullanırken, Kanada’da bu oran biraz daha iyi durumdadır (%15 ve %27 arasında). Formal yaklaşımlara rağbetin neden düşük olduğunu izleyen çalışmalarda soruşturmalısı gereklidir.
- Soru 3-Manuel test ve test otomasyonu: hem Kanada hem de Türkiye’de, genel olarak her iki test yaklaşımının kullanımında geniş bir yelpaze (hiçbir zaman ile her zaman arasında bir yerlerde) vardır. Bu durum, farklı katılımcıların çok farklı uygulamaları olduğunu belirtir, örneğin, bazıları yoğun olarak test otomasyonu uygularken, diğerleri manuel testi tercih etmektedir.
- Soru 4-Genel olarak, Türk yazılım endüstrisinde sorulan dört test kapsama metriğinin (satır, karar, koşul ve MC/DC) tamamının popülerliği düşüktür.
- Soru 5-Kod satırı (LOC) başına düşen (ortalama) hata sayısı, ve başarılı kullanıcı kabul testlerinin sayısı gibi test ve kalite ölçütleri Türkiye’de geniş bir kullanıma sahip değildir.
- Soru 6-Çoğu şirkette, 1:2 ile 1:5 ve üstü arası değişen oranlarla testçiler geliştiricilerden sayıca azdır. Şekilde görüldüğü üzere, testçi:geliştirici oranı 1:5’ten 1:2’ye çıktıkça frekans düşmektedir. Ayrıca, bazı şirketler ya bu iki rol arasında bir ayrım yapmamakta ya da bu metriği ölçmemektedir.
- Soru 7-Test aktivitelerinin bittiğine karar vermek için kullanılan kriterler arasında en geniş kullanılan tüm test durumları başka hata bulmadan çalıştırılıyorsa kriteridir.

- Soru 8- Test-sonra Geliştirme yaklaşımlarının (geliştirmeden sonra test etme) hala Test Güdümlü (önce test) Geliştirmeden çok daha popülerdir.
- Soru 9-Test sırasında yürütülen uygulamalar arasında, en çok ve en az popüler aktiviteler sırasıyla “Yayımdan (release) önce geliştirici ürünü test etmektedir”, ve “Birim testler formal olarak gözden geçiriliyor”dur.

Çapraz faktör analizi de (Kısım 4.4) aşağıdaki diğer ilginç bulguları açığa çıkarmıştır.

- Yıl arttıkça, manuel test kullanımındaki artış eğilimi, test otomasyonunun kullanımındaki artış eğiliminden daha diktir.
- Askeriye ve Savunma sektörü için ürün geliştiren katılımcılar, diğer üç sektörle karşılaştırıldığında, biraz daha fazla kapsama metriği kullandıklarını bildirmişlerdir.

5.2 Geçerliliğe Tehditler

Dış geçerliliğe tehditlerden birisi katılımcıların demografik dağılımında yatmaktadır. Kısım 3.2’de bildirildiği gibi, katılımcılar çoğunlukla araştırmacıların Türk yazılım şirketlerindeki partner ve bağlantı ağı üzerinden davet edilmişlerdir. Bağlantı ağımızın dışındaki şirketler muhtemelen anket popülasyonu içinde düzgün temsil edilememişlerdir. Bu geçerlilik hususunu hafifletmek için, e-posta listemizi bütün başlıca Türk şirketlerini içerecek şekilde genişletmeye çalışıldı; çevrimiçi sosyal ve profesyonel ağlarda (örneğin, LinkedIn ve Twitter) mesajlar oluşturuldu; ve tüm davetlilerin çevrimiçi anketi doldurabilmesi için 1.5 ay ayırıldı.

Yapısal (construct) geçerlilik açısından, sorun bu ankette gerçekten var olan test uygulamalarını ölçüp ölçmediğimizdir. Bu çalışmada birçok anket çalışmasında da uygulanan bir teknik kullanıldı —her soru için oyları sayıldı ve istatistiki çıkarımlar yapıldı. Bu şekildeki oylamaya dayalı sonuçların, belli bir ölçüde, Türk uygulayıcıların çoğunluğunun görüşlerini yansıttığına inanılmaktadır.

Çalışmamızın sonuç geçerliliği açısından şunlar söylenebilmektedir. Test uygulaması probleminin yalnızca teknik bir husus hakkında değil daha çok ekonomik ve psikolojik olduğu sonucuna nitel olarak varmaya çalıştık. Bu durum [37]’den esinlenilmiştir. Anketimizde, tarafsız kalabilmek için her test bakışında (test türü, teknik, otomasyon, vb.) istatistiki bulgulara başvurulmuştur.

Son olarak, bu çalışmada herhangi bir nedensel ilişki oluşturma amaçlanmamıştır, bu yüzden iç geçerlilik tartışması bu kısımda dâhil edilmemiştir.

6 Sonuç ve Gelecek Çalışmalar

Bu anket sayesinde, yazarlar, Türk endüstrisinin gelişmiş test uygulama ve yaklaşımlarını kullanmayı iyileştirmek için uzun bir yolu olmasına rağmen, son zamanlarda testin önemine dikkat ettiğini gözlemlemiştir.

Yazılım test uygulamalarındaki son eğilimleri karşılaştırabilmek için gelecekte, başka bölge ve ülkelerde benzer çalışmaların yapılması ihtiyacı vardır. Türkiye’deki test uygulamalarının olgunluğunu değerlendirmek de yararlı bir çalışma olacaktır.,

Böyle bir çalışma, Test Maturity Model Integration (TMMI) [31] gibi standartlara paralel gerçekleştirilebilir.

Teşekkür. Vahid Garousi, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK)'ın Konuk veya Akademik İzinli (Sabbatical) Bilim İnsanı Destekleme Programı (#2221) tarafından desteklenmektedir. Anketimize katılan tüm yazılım mühendislerine gönülden teşekkür ederiz.

Kaynaklar

1. US Government Accounting Office, "Greater Emphasis on Testing needed to Make Computer Software more Reliable and Less Costly," report # GAO/IMTEC-64.2, 1983.
2. D. Gelperin ve B. Hetzel, "The Growth of Software Testing," *Communications of the ACM*, vol. 31, pp. 687-695, 1988.
3. Quality Assurance Institute, "Status of Software Testing," www.geocities.com/mtarrani/StatusOfSoftwareTesting.doc, Mar. 2002 [cited Oct. 2009].
4. C. Andersson ve P. Runeson, "Verification and validation in industry - a qualitative survey on the state of practice," in *Proceedings of the International Symposium on Empirical Software Engineering*, 2002 pp. 37-47.
5. P. Runeson, C. Andersson, ve M. Host, "Test processes in software product evolution - a qualitative survey on the state of practice," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 15, pp. 41-59, 2003.
6. R. Torkar ve S. Mankefors, "A survey on testing and reuse," in *Proceedings of IEEE International Conference on Software: Science, Technology and Engineering*, 2003, pp. 164-173
7. A. M. Geras, M. R. Smith, ve J. Miller, "A Survey of Software Testing Practices in Alberta," *Canadian Journal of Electrical and Computer Engineering*, vol. 29, pp. 183-191, 2004.
8. S. P. Ng, T. Murnane, K. Reed, D. Grant, ve T. Y. Chen, "A Preliminary Survey on Software Testing Practices in Australia," in *Proceedings of Australian Software Engineering Conference*, 2004, pp. 116-125.
9. O. Taipale, K. Smolander, ve H. Kälviäinen, "Finding and Ranking Research Directions for Software Testing," in *European Conference on Software Process Improvement*, ed, 2005, pp. 39-48.
10. M. A. Wojcicki ve P. Strooper, "A State-of-practice Questionnaire on Verification and Validation for Concurrent Programs," in *Proc. of workshop on Parallel and Distributed Systems: Testing and Debugging 2006*, pp. 1-10.
11. P. Runeson, "A Survey of Unit Testing Practices," *IEEE Software*, vol. 23, pp. 22-29, 2006
12. M. Grindal, J. Offutt, ve J. Mellin, "On the testing maturity of software producing organizations," in *Testing: Academia & Industry Conference-Practice And Research Techniques*, 2006.
13. D. Martin, J. Rooksby, M. Rouncefield, ve I. Sommerville, "'Good' Organisational Reasons for 'Bad' Software Testing: An Ethnographic Study of Testing in a Small Software Company," in *International Conference on Software Engineering*, 2007, pp. 602-611.
14. V. Garousi ve T. Varma, "A Replicated Survey of Software Testing Practices in the Canadian Province of Alberta: What has Changed from 2004 to 2009?," *Journal of Systems and Software*, vol. 83, pp. 2251-2262, 2010.

15. S. Ambler, "Test Driven Development (TDD): Reality Over Rhetoric," *Dr. Dobb's Journal*, <http://www.ddj.com/architect/212902568?cid=Ambysoft>, Jan. 2009 [cited: Oct. 2009].
16. E. Engström ve P. Runeson, "A Qualitative Survey of Regression Testing Practices," in *International Conference on Product-Focused Software Process Improvement*, 2010, pp. 3-16.
17. A. Causevic, D. Sundmark, ve S. Punnekkat, "An industrial survey on contemporary aspects of software testing," in *International Conference on Software Testing Verification and Validation*, 2010, pp. 393-401.
18. V. Garousi ve J. Zhi, "A Survey of Software Testing Practices in Canada," *Journal of Systems and Software*, vol. 86, pp. 1354-1376, May 2013.
19. Turkish Testing Board (TTB), "Turkish Software Quality Report 2013-2014," http://www.testistanbul.org/TurkeySoftwareQualityReport_2013_2014.pdf, 2013.
20. G. Tassej, "The Economic Impacts of Inadequate Infrastructure for Software Testing," *Planning Report 02-3*. Prepared by RTI for the National Institute of Standards and Technology (NIST), <http://www.nist.gov/director/prog-ofc/report02-3.pdf>, 2002.
21. G. Tirpançeker, "Türkiye Yazılım Sektörü ve Yazılımın Yarattığı Katma Değerler," http://www.sde.org.tr/userfiles/file/Gulara_Tirpanceker_SDE_2011Aral%C4%B1k-2.pdf, Türkiye'de yazılım sektörü çalıştayı, Stratejik Düşünce Enstitüsü, Ankara, 2011.
22. The Standish Group, "Extreme CHAOS," http://www.standishgroup.com/sample_research/showfile.php?File=extreme_chaos.pdf, 2001 [cited: Oct. 2009].
23. The Standish Group, "CHAOS Manifesto," https://secure.standishgroup.com/newsroom/chaos_manifesto.php, Oct. 2009 [cited: Oct. 2009].
24. M. Hayes, "Quality First," *InformationWeek*, vol. 889, pp. 38-54, 2002.
25. L. Koskela, *Test Driven: TDD and Acceptance TDD for Java Developers*: Manning Publications, 2007.
26. V. R. Basili, "Software modeling and measurement: the Goal/Question/Metric paradigm," *Technical Report*, University of Maryland at College Park 1992.
27. D. A. Garvin, *Managing quality: the strategic and competitive edge*: Free Press, 1988.
28. J. Whittaker, "Google vs. Microsoft, and the Dev:Test Ratio Debate," http://blogs.msdn.com/james_whittaker/archive/2008/12/09/google-v-microsoft-and-the-dev-test-ratio-debate.aspx, Dec. 2008 [cited: Oct. 2009].
29. K. Iberle ve S. Bartlett, "Estimating Tester to Developer Ratios (or Not)," in *Pacific Northwest Software Quality Conference*, http://www.stickyminds.com/s.asp?F=S6174_ART_2, 2001.
30. M. A. Cusumano ve D. B. Yoffie, "Software Development in Internet Time," *IEEE Computer*, vol. 32, pp. 60-69, 1999.
31. TMMi Foundation, "TMMi Reference Model v2.0," <http://www.tmmifoundation.org/downloads/tmmi/TMMi%20Framework.pdf>, Last accessed: Feb. 2010.
32. Survey questions, <http://goo.gl/i1FUd>, Last accessed: July 2013.