

# Tümleşik VoIP Sistemlerinde Test Stratejileri

Miraç Emektar<sup>1</sup>, Ömer Nabi Akdeniz<sup>1</sup>, Oğuzhan Yavuz<sup>1</sup>

<sup>1</sup> Netaş Telekomünikasyon A.Ş., İstanbul, Türkiye  
{emektar, oakdeniz, oyavuz}@netas.com.tr

**Özet.** Bu çalışmada haberleşme sistemlerinin (VoIP, PSTN) yazılım tasarım süreçlerinde test stratejisi ve tekniklerine ilişkin Netaş'ın sahip olduğu bilgi birikimi ve tecrübe paylaşılmıştır. Netaş'ın yazılım Ar-Ge sorumluluğunu üstlendiği VoIP sistemleri oldukça karmaşık yazılıma sahiptir. Burada yapılan geliştirmeler mevcut yazılım üzerine yeni eklemeler şeklinde tasarlanmaktadır. Bundan dolayı tasarlanan yeni yazılımın ve tüm sistemin doğru test stratejisiyle test edilmesi elzemdir. TÜBİTAK TEYDEB tarafından desteklenmiş olan “Yüksek Kapasiteli Yeni Nesil Merkezi Santral Tasarımı” isimli projenin test uygulaması anlatılmıştır.

**Anahtar Kelimeler:** Yazılım Süreçleri, Test Stratejisi, Test Teknikleri, VoIP Sistemleri, Test Planlama

## 1 Giriş

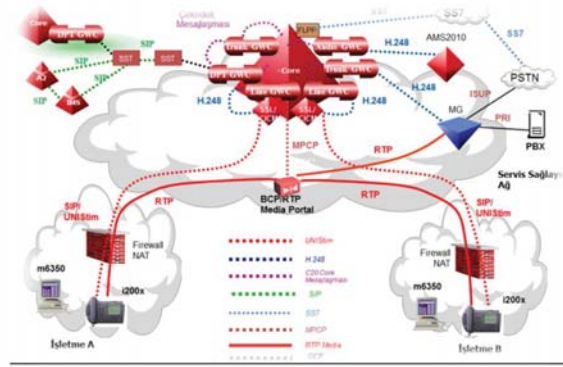
Bir haberleşme sisteminde sağlıklı haberleşme, donanımsal alt yapı ve bu yapının üzerinde çalışan yazılım ile sağlanmaktadır. Bununla birlikte bu sistemler üzerindeki yazılım donanımdan bağımsız olarak tasarlanmaktadır. Ayrı bir ürün olarak ortaya konan bu yazılımın belli haberleşme standartlarını ve kalitesini sağlaması gerekmektedir. Haberleşme sektöründe yazılım tasarım mevcut yazılım üzerine yeni eklemeler yapılarak zuhur etmektedir. Bu yeni tasarlanan yazılım bloklarının tasarlanma amaçlarını yerine getirmesinin yanında mevcut sistemin çalışmasını olumsuz etkilememesi gerekmektedir.

Yazılım Ar-Ge faaliyetlerinin en az tasarım kadar önemli olan fazı test evresidir. Bu evrede yeterli yapılmayan testler, müşteri sahasından problem olarak geri dönmektedir. Bu durumda oluşan bu problemlerin çözümü oldukça maliyetlidir. Ayrıca ürünün yazılımının kalitesi ve Ar-Ge firmasına olan güven çok ciddi yara almaktadır.

Netaş'ın Ar-Ge sorumluluğunu yaptığı tümleşik VoIP sistemleri [1], Çekirdek (Core), Ağ Geçidi (Gateway, GW), Ağ Geçidi Denetleyicisi (Gateway Controller, GWC), Oturum Trank Sunucusu (Session Server Trunk, SST), İşletim Yönetim Birimi (OAM&P) gibi bileşenlerden oluşmaktadır. Bununla birlikte bu sistem dünyadaki neredeyse tüm haberleşme standartlarını desteklemektedir. Bu sebeplerden dolayı 41 milyon satır koddan oluşan bu yazılım oldukça karmaşıktır [1].

Şekil 1'de görülen Çekirdek (Core), aramanın kurulması, yönlendirilmesi ve sonlandırılması esnasındaki sinyalleşme gibi arama ve servislerle ilgili tüm kontrolleri üstlenir. Ayrıca hizmet verdiği abonelerin tüm kayıtlarını, faturalandırma detayları

gibi bilgileri oluşturur. Bunlara ek olarak üzerinde çok çeşitli ve zengin servisleri barındırır. Ağ geçidi denetleyicisi (GWC) Çekirdek ile ağ geçitleri arasındaki bağlantıyı sağlamaktadır. Bir GWC 64 adet ağ geçidini destekleyebilmektedir. Çekirdek ile bağlantı kopması durumunda kendisine bağlı ağ geçitler arasında aramanın kurulumunu yapabilmektedir. Ayrıca hatların durum güncellemeleri GWC tarafından Çekirdek'e sağlanmaktadır. Oturum Trank Sunucusu Session Initiation Protocol (SIP) [3] kullanarak santrali IP altyapıya bağlayan, diğer santraller ile bağlantıyı sağlayan birimdir. SST, Çekirdek'e GWC üzerinden bağlanırken sisteme özel bir protokol kullanılır.



Şekil 1. Netaş Tümleşik Çözüm Mimarisi ve Protokoller

Bu karmaşık yazılım için Netaş'ın kullandığı test stratejisi bu çalışmada anlatılmaktadır.

Bu yayın dört bölümden oluşmaktadır. Bölüm 2'de tümleşik VoIP sistemlerinde kullanılan test stratejisi anlatılmıştır. Bölüm 3'de "Yüksek Kapasiteli Yeni Nesil Merkezi Santral Tasarımı" projesinde test stratejinin uygulanmasına değinilmiştir. Son bölümde ise genel değerlendirmelere ve sonuçlara yer verilmiştir.

## 2 Yeni Nesil VoIP Sistemlerde Test Stratejisi

Netaş'ın ArGe faaliyetlerinden sorumlu olduğu yeni nesil VoIP santralinin karmaşıklığı oldukça yüksek olan bir sistemdir. Bu sistem üzerinde yapılan geliştirmeler var olan yapıyı bozmadan sisteme yüklenmesini gerektirmektedir. Bu oldukça riskli ArGe faaliyetinin en önemli kısmı testler ve bu testlerin belirlenmesinde kullanılan test planı ve stratejileridir. Ayrıca bu strateji doğrultusunda yapılan testler sonucunda %99.9999 oranında başarı sağlanmadan tasarlanan geliştirme sahaya verilmemektedir. Bu karmaşık sistemde tasarımın başarısını ölçmek için Şekil 2'de verildiği üzere birim test (BT, Unit Test; UT), özellik doğrulama (ÖD; Feature Verification, FV) ve sistem doğrulama (SD, System Verification; SV) olmak üzere üç evrede test işlemi uygulanmaktadır. Bunlara ek olarak, yapılan projenin büyüklüğüne ve karmaşıklığına

göre şelale (waterfall) [3] veya çevik (agile)[4] proje yönetim teknikleri uygulanmaktadır.



Şekil 2. Birim test, sistem ve özellik doğrulama

Netaş'ın test strateji doğrultusunda her bir test tipi için farklı ekipler oluşturulmuştur. İlk test tipi olan BT'de amaç tasarlanan birimin doğruluğunun kontrol edilmesidir. Burada, tasarımın performansını ölçmek, BT için ana amaç değildir. BT senaryoları test uzmanlarınca değil yazılım tasarımcıları tarafından yazılır ve uygulanır. Burada, tasarımın istenilen özellikleri yapabilirliği ve kodda izlediği yollar incelenir. Burada yazılan test senaryoları Beyaz Kutu (White Box) [5] test tekniği kullanılarak oluşturulur. Beyaz kutu tekniği, tasarlanan yapının kod mantığı üzerindeki bilgiye bağlıdır ve akış denetimleri, koşullar vb. elemanlar sınanır. Oluşturulan test senaryoları, onay sürecinin ardından, kalite takibi amacıyla HP'nin Kalite Merkezi (Quality Center) adındaki araca aktarılır.

Yapılan BT ardından tasarım, ÖD adımı için genel yapıya hâkim, ağ, yazılım ve sistem bilgisine sahip test mühendislerinden oluşan ekibe aktarılır. Bu ekip ÖD adımı, yazılım talepleri ve yazılan kod doğrultusunda uygulanacak test senaryolarını ve test planını belirler. Bu adıma, Gri Kutu [6] ve Siyah Kutu [7] (Gray Box ve Black Box) test teknikleri örnek verilebilir. Gri kutu testlerinde, test tasarımı sırasında sadece testini yaptığımız fonksiyonun yazılımsal anlamda içeriğinin bilinmesini gerektirir. Yani büyük resme bakmayıp, sadece yazılımın bir parçası hakkında bilgi sahibi olmayı gerektirir.

Kara Kutu test uygulaması yazılımın tipine bakmaksızın seçtiği test yöntemini uygulayan yazılım test tekniklerinden oluşur. Kara Kutu test tekniğine girdi olarak uygulanabilir bir yapı taşı (Component) ve bir varsayımla işe başlanır [7]. Her program girdisinden sonra oluşan çıktı kontrol edilir, hata meydana gelmiş ise bu teknik aracılığı ile belirlenir. ÖD takımı tarafından hazırlanan test senaryoları ve test planı, yazılım mimarları ve Netaş'ın uluslararası iş ortakları ile yapılan görüşmeler neticesinde onaylatılarak, test işlemi sürecine başlanılır. Hazırlanan dokümanlar kalite merkezi aracında ve ortak paylaşım dizininde konumlandırılır.

ÖD ekibi, yapılan test aktivitelerini;

- *Kaynak Testi (Sourcing Test)*: Yeni sürüm için hazırlanan yazılımın test edildiği aktivite çeşididir. Burada önceki sürümlere uygulanacak yükseltme testleri de yer almaktadır.
- *Yama Testi (Patching Test)*: Önceki sürümlere yama olarak hazırlanan yazılımların testleridir. Müşteri tarafından yaşanan bir soruna veya müşterinin acil taleplerine cevap verebilmek için hazırlanan yamalar, kaynak testlerindeki yükseltme testlerine karşın tablo transfer testlerini daha yoğun içerir.

iki ana başlık altında ele alır. Testin nasıl yapılacağı “ÖD Test Plan” dokümanında yer alır. Burada aktivitede ne yapılmak istendiği ve neden yapılmak istendiğine dair bilgiler yer almaz. “Proje Genel Bakış” başlığında kısaca konu özetlense de aktivitenin ne yaptığını anlatan Fonksiyonel Tanım, (FT, Functional Description) dokümanıdır. Ayrıca müşteri talepleri özellik gereksinim spesifikasyon (ÖGS, Feature Requirements Specification) dokümanında toplanır. Bu taleplere karşılık özellik teknik spesifikasyon (ÖTS, Feature Technical Specification; FTS) dokümanı hazırlanır ve hangi taleplerin tasarıma uygunluğu gerekçeleri ile birlikte bu dokümanda, müşteriye sunmak üzere yer alır. ÖTS doğrudan olarak müşteriye ile paylaşılacağı için buradaki her cümle ÖD adımı test edilmelidir.

Bunlarla birlikte ÖD test plan dokümanında, yapılan aktivitenin nereleri etkileyeceği, nereler üzerinde tasarım yapıldığı ve nerelerde kod değişikliğine gidildiği bilgileri tasarım ve test grupları tarafından saptanarak “Etkilenen Ağ” (Impacted Network) başlığı altında toplanır.

Haberleşme alanında, teknoloji bir öncekinin üzerine eklenerek ilerlediğinden, ürün için geliştirilmiş birden fazla platform mevcuttur. Xa-Core, Compact, aTCA mimarileri telekom alanında geliştirilen platformlara birer örnektir. Tüm bu sürüm ve platform farklılıkları, farklı yazılım ve kapasitelere sahip yapılar oluşturmuştur. Aktivitenin, müşterinin konumu, mevcut platformu ve hangi yapılarda sürüme ekleneceği saptanarak test ortamları belirlenir. Testi yapılan aktivitenin desteklediği platformlar, ÖD dokümanında “Platform Verificaiton” başlığı altında belirtilir. Değişik bileşenlerden oluşan VoIP sistemleri için özelleşmiş tasarım grupları vardır. İlgili aktivitede hangi grupların çalıştığı, görev paylaşım ve çalışma zamanının bilgileri “Focus Area” başlığı altında anlatılır. Bu bilgiler, ÖD grubundaki test mühendislerinin yol haritalarını çıkarılmasında önemli bir yeri vardır.

Sahada karşılaşılabilecek durumların senaryolarını ve müşteri kabul testlerini içeren, tasarımın geliştirme detaylarından ve yazılan koddan bağımsız, test adımı SD mühendisleri tarafından oluşturulur. Bu tip testler kara kutu olarak bilinmektedir. Yeni tasarımın genel sistemde neleri etkilediği bu adımda detaylı olarak incelenir. Müşteriye sunulacak hale getirilmiş olan tasarımın testi bu adımda yapıldığından, bulunabilecek her hata veya eksiklik müşteri gözüyle incelenir. Bu adım diğer testleri gerçekleştiren ve kodun etkilerini bilen mühendislerden ayrı bir grup tarafından koşulur. Tespit edilen eksiklikler müşterinin proje iptaline ve anlaşma feshine kadar gidecek durumları önceden engellemek adına büyük önem taşır.

Bunlara ek olarak buradaki test senaryoları tümleşik VoIP santralının değişik bileşenlerinde PROTEL[8], C++ ve Java gibi çeşitli dillerde yazılmaktadır.

Gereksinimler doğrultusunda hazırlanan test senaryoları uygun test alanları altında gruplandırılır. Bir önceki sürümün, geliştirilen yeni özelliklerle birlikte, çalıştığını kontrol etmek için “backwards” adı verilen test senaryoları uygulanmaktadır. Yeni tasarlanan sistemin limitlerini belirlemek için “Capacity & Stress” test senaryoları kullanılmaktadır. Ayrıca tasarım gereksinimleri bu test senaryoları sonucunda belirlenebilmektedir. Haberleşme sistemlerinde standartlara uyumluluk oldukça önemlidir. Bunun kontrolü “Compliance” test senaryoları altında yapılmaktadır.

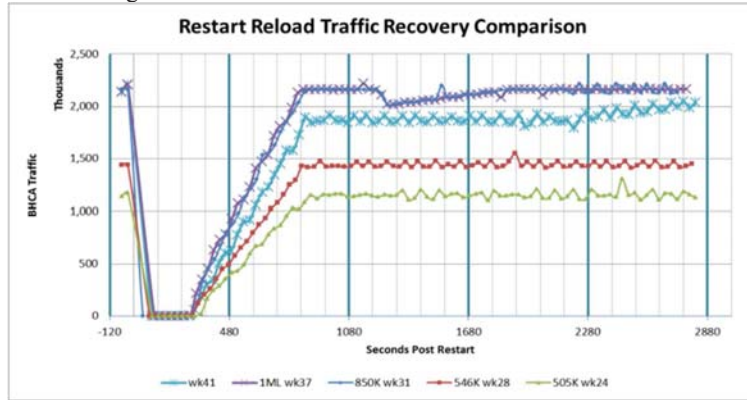
Sistemin istenildiği şekilde çalışmasını sağlayan biçimlendirme ve tanımlamaları belirlemek için “Configuration & Provisioning” test senaryoları sisteme uygulanmak-

tadır. Müşteri dokümanlarındaki bilgileri doğruluğunu tespiti için “Documentation” testleri gerçekleştirilmektedir.

Genellikle tasarımın yapması beklenen işlemleri test edilmektedir. Hâlbuki sistemin en önemli davranışı beklenmeyen durumlarda gösterdiğiidir. Bunları belirlemek için “Error Path & Negative Scenarios” test senaryoları kullanılmaktadır. Buradaki temel amaç, sistemin çalışmasını bozan “trap” olarak bilinen ve çözüm maliyeti oldukça yüksek olan bozucu etkilerin belirlenmesine çalışmaktır. Bununla birlikte ön görülen hatalar karşısında sistem hata bilgisini ilgili birime ve müşteriye iletmesi gerekmektedir. Bu durum karşılaşılan hatanın çözümü için oldukça önemlidir ve kritik bir uygulamadır. Hata karşısında doğru hata kodlarının gönderilmesi “Fault Management” test senaryoları ile kontrol edilmektedir.

Tasarım sonucunda elde edilen yazılım beklenenleri gerçekleştirdiği operasyonel testler ile belirlenmektedir. Bu testler, ÖD’nin üzerinde yoğunlaştığı bir alandır. Başka bir deyişle test deyince ilk akla gelen test senaryosu bu tip senaryolardır. Bununla birlikte sistemin bir önceki sürüme veya bir sonraki sürümlere güncellenmesinde mevcut geliştirilen yazılımın etkisi “Upgrade & Backup” senaryoları kontrol edilir. Ayrıca tasarımın var olan sürümdeki modüller arasındaki etkileşimler ve uyumlu çalışması “Release FIT” senaryoları ile test edilir. Bu test senaryoları sürümün güvenilirliğinin belirlenmesi açısından oldukça önemlidir.

Belli senaryolar ve yük altında sistemin performansı (arama kurulma zamanı, başarısız arama sayısı...), performans testleri ile kontrol edilir. Elde edilen sonuçlar bir önceki sürümlerin sonuçları ile karşılaştırılır. Şekil 3’te tümleşik VoIP sisteminin yeniden başlatma sürelerine ilişkin performans değerleri gösterilmiştir. Ayrıca bu testler için sistem performans ölçüm grubu mevcuttur. Bunlarla birlikte yeni sürümle güncellenen ürünün mevcut altyapıdaki diğer ürünlerle olan çalışması “Interoperability & Interworking” altında test edilmektedir.



Şekil 3. Arama trafiği altındaki sistemin, yüksek kapasiteli santral tasarım projesinin haftalara göre, yeniden başlatılma performans değerleri grafiği.

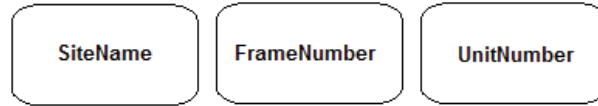
### 3 Yüksek Kapasiteli Santral Tasarım Projesi Test Stratejisi

Bu bölümde “Yüksek Kapasiteli Yeni Nesil Merkezi Santral Tasarımı” projesinde izlenen test yöntemleri ve edinilen deneyimlere binaen uygulanan metotlar anlatılmaktadır.

Kapasite artırımı esnasında, sistem performansı, test takımının dikkat etmesi gereken en önemli parametredir. Sistem performansının karşılaştırmalı veriler ile ölçülmesi yöntemi klasik fakat etkili bir yöntem olduğundan, test mühendisleri bu yöntemden de Şekil 3’teki gibi faydalanmışlardır. Sistem performansları, performans değerleri elde edilerek, bunların değerlendirilmesiyle ölçülmektedir. VoIP sistem bileşenlerinin tasarıma başlamadan önce performans değerleri elde edilmiştir.

Bunlara ek olarak, yapılan tasarımın test edilmesi için, konfigürasyon için gerekli XML ve SOS işletim sisteminin konfigürasyon dosyalarını otomatik hazırlayan araçlar geliştirilmiştir. Bash scripting ile Linux ortamında geliştirilen bu araçlar sayesinde gerekli konfigürasyon dosyaları hatasız ve çok kısa sürede tamamlanabilmektedir.

Ayrıca, test mühendisleri yapılan testlerin sonuçlarını ve etkiledikleri alanları daha ayrıntılı gözlemleyebilmek adına sistem hata kayıtlarından faydalanmaktadır. Kapasite artırımıyla birlikte gelen büyüme, santrallerin hata kodlarını okumayı ve o kodlarla ilgili elemanların tespitini zorlaştırmıştır. Bu karmaşıklık test mühendislerinin hata anındaki kodları okurken, her seferinde 5-6 adımdan oluşan bir dizi işlemleri yapmasına sebep olur. Bu işlemleri her defasında yapmak, önemli bir zaman sarfiyatı ve iş yüküdür. Test mühendisleri, Şekil 4’te gösterilen artan hat grup isim bilgilerini daha kolay takip edebilecekleri bir metodu, hazırladıkları tüm konfigürasyon araçlarında uygulamışlardır.



Şekil 4. Hat grup ismi bilgi formatı

Şekil 4’te verilen hat grup isim bilgisini daha anlaşılır yapan metotta ise, taraf ismi yerine GXXX şeklinde G ve 3 rakamdan oluşan bir format tercih edilerek ilgili LGRP’nin hangi ağ geçidi denetleyicisinin altında bulunduğu belirtilmiştir. Böylelikle 1milyon aktif son kullanıcı konfigürasyonuna sahip tümleşik VoIP santral kayıtlarını okumada büyük kolaylık sağlanmıştır. Hat grup isimlerine ait taraf ismi bilgisi kısmı olarak bilgi verse de, arıza yerini net olarak söylememektedir. Çerçeve ve birim numara bilgilerindeki toplam 4 karakter kullanılarak, eleman tespitinin başka kontrollere gerek kalmadan direkt kayıtlar üzerinde okunarak yapılması sağlanmıştır. Örneğin, ismi G116 01 64 şekilde olan bir hat grubun, ağ geçidi denetleyicisi 116’nın altındaki 164. ağ geçidi olarak anlaşılması sağlanmıştır. Bir ağ geçidi denetleyicisi altına en fazla 254 ağ geçidi tanımlanabilmektedir. Sağdan itibaren ilk 3 rakam hangi ağ geçidi kontrolü olduğunu anlatmış en soldaki rakam da test hat grupları kullanılmıştır.

Kapasite artırımında, kullanıcıları ulaşılabilir hale getirmek kadar, aynı anda yapılabilen arama sayısını artırmakta büyük önem taşımaktadır. Bu kapsamda “Capacity

& Stress” testleri sistemde uygulanmıştır. Çizelge 1’de, bulunan hata senaryolarının hangi alanlara ait olduğunu görebilirsiniz. En çok hatanın tespit edildiği alanlar, en riskli alan olarak kabul edilmemektedir. Hata yolları ve negatif senaryo testlerindeki hata sonuçları sistemin alt seviyelerine veya belleklerine ait hatalar olabileceğinden bu alanda bulunan hataların önceden tespiti büyük önem taşımaktadır. Operasyon alanında, elde edilen sistemin gereksinimleri ne oranda gerçekleştirdiği test edilmektedir. Müşterinin kabul testleri, operasyon testleri ile benzer olabilmektedir. Bu alanda yapılan testlerin kalitesi, ürün tesliminden sonra ki süreçte müşteri memnuniyeti ile doğru orantılı olmaktadır. Provisioning testleri, girilen kodların sistem konfigürasyonları üzerindeki etkisini ölçmektedir. Sistemin istenilen özelliklere sahip olabilmesi, öncelikle konfigürasyonların eksiksiz yapılmış olmasına bağlıdır. Hataların bu test adımıyla tespit edilmemesi, müşterinin kabul testlerine bile başlayamaması anlamına gelir. Bu durum test mühendisleri için büyük saygınlık kaybı anlamına gelir.

Müşterilere, VoIP santral kalite taahhütlerini sunabilmek ve sistem kapasitesindeki etkiyi saptayabilmek için, ÖD ve SD adımlarındaki testlerin arama trafiği altında gerçekleşmesi ve trafik altında alınmış performans değerlerine ihtiyaç vardır. Yapılan son kullanıcı tanımlamalarını kullanılabilir hale getirmek ve son kullanıcıların arama yapabilmesini sağlamak için her son kullanıcının bir hat grubuna bağlanması gerekmektedir.

Netaş ÖD grubu test mühendisleri, test durumlarını sağlamak ve otomatik arama trafiği oluşturmak adına Şekil 5’te gösterilen “Hurricane” adı verilen Linux tabanlı simülasyon araçlarını kullanmaktadır. Bu simülasyon aracıyla yapılan son kullanıcı tanımlamaları çalışır hale getirilerek, kullanıcıların bir birilerini aramaları sağlanmaktadır. Fakat VoIP santralin kapasite artırımı gibi bir aktivitesini test edebilmek ve bu yönde zorlama testlerini gerçekleştirebilmek için, çok sayıda ağ geçidini simüle edebilecek “Hurricane” sunuculara ihtiyaç duyulmaktadır. Kullanılan benzetim araçlarını kurmak, üzerlerinden arama trafiği başlatmak ve gerektiğinde kapatabilmek adına bash script’ler ile otomasyon araçları geliştirilmiştir. ÖD grubu bu araçları geliştirerek büyük zaman kayıplarından kurtulmuş ve mevcut zamanı daha verimli kullanabilmiştir.

Projede, büyüklüğü ve etkilediği alanların çok olması sebebiyle, çevik yöntem tercih edilmiştir. Testi yapılan ürün her hafta güncellenmiş ve mevcut etkileşim testlerinin tekrarlanması gerekli hale gelmiştir. Bu kapsamda test sürecinin sonunda “Regression” adı verilen dönüş testleri ile onaylanan test senaryoları tekrar koşulmuştur. Böylelikle testten sonra girilen kodların bozduğu alanlar tespit edilmiş, gerekli düzeltmeler yapılarak “Regression” testleri sayesinde kodda istenilen kalite yakalanmıştır.

#### 4 Sonuçlar

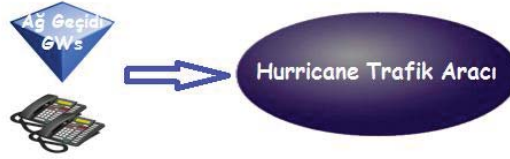
Çok bileşenli, üzerinde 1300 civarı hizmet ve uygulama bulunduran ve nerdeyse tüm dünyadaki haberleşme standartlarını destekleyen Netaş’ın Ar-Ge sorumlusu olduğu tümleşik VoIP sisteminin karmaşık yazılım bloklarının kalitenin korunması adına iyi test edilmesi gerekmektedir. Bu sistemin testi, Netaş’ın test stratejisi doğrultusunda

yapılmaktadır. Bu test stratejisi gereğince her 1000 satır kod için en az bir problem raporlanmalıdır. Aksi durumda yapılan test sorgulanmaktadır.

Yüksek kapasiteli santral tasarımında yukarıda verilen test stratejisine ek olarak yeni yaklaşımlar kullanılmıştır. Bu yaklaşımlar sonucunda uzun sürede yapılacak testler çok kısa sürede tamamlanmıştır. Ayrıca bu süredeki tasarrufun yanında tasarımdaki tüm riskli noktalar sınanmıştır. Bununla birlikte alanlara göre test senaryoları yazılarak tüm sistemi kapsayan testler yapılmıştır ve sistemi etkilemesi muhtemel problemlerin tespiti gerçekleştirilmiştir. Bu sistemin kalitesini artırırken maliyetleri düşürmüştür.

**Çizelge 1.** Yüksek kapasiteli santral tasarım projesinin test senaryo dağılımı ve hata tespiti yapılan alanlar grafiği

Functional Area	Total TCs	Executed TCs	Failed TCs
Backup and Restore	9	9	0
Capacity and Stress	8	7	0
Error Path and Negative Scenarios	76	75	6
Fault Management	34	34	6
Last GW EP Focus	17	17	1
Mismatch Recovery	28	28	3
Operational	147	144	20
Patch	8	8	0
Performance	23	22	1
Provisioning	1106	1102	15
Robustness and Recovery	212	211	15
Tools	84	84	8
Upgrade	28	28	6
<b>Grand Total</b>	<b>1780</b>	<b>1769</b>	<b>81</b>



**Şekil 5.** Hurricane aracının simüle ettiği yapı

Bu çalışmada verilen test stratejisine ve yeni yaklaşımlara, plan aşamasında olan test otomasyonunun da eklenmesiyle daha etkin hale gelecektir. Bu konuda çalışmalara başlanmıştır.



## Bilgilendirme

Bu çalışmada adı geçen, ‘‘Yüksek Kapasiteli Yeni Nesil Merkezi Santral Tasarımı’’ TÜBİTAK-Teknoloji ve Yenilik Destek Programları TEYDEB tarafından 3120226 numaralı proje olarak desteklenmiştir.

## Kaynakça

1. Goode, B., ‘‘Voice over Internet Protocol (VoIP)’’, Proceedings of the IEEE, vol. 90 (6), pp. 1495-1517, (2002)
2. Aktürk, E., Demirkıran F., Uysal Ö., Hacıbeyođlu B., ve Yavuz O., ‘‘IMS-TDM Entegrasyonu: AGCF Çözümü’’ IEEE 22. Sinyal İşleme ve İletişim Uygulamaları (SİU 2014), pp.1-4, (2014)
3. Mahmood, S., ‘‘Survey of component-based software development’’ IET Software, vol. 1(2), pp. 57-66, (2007).
4. Aoyama, M., ‘‘Agile Software Process and its experience’’ Proceedings of the 1998 International Conference on Software Engineering, pp. 3-12, (1998).
5. Varadan G.S., ‘‘Trends in reliability and test strategies’’ IEEE Software, vol. 12(3) (1995).
6. Li, Z.J., Tan, H.F., Liu, H.H., Zhu, J. ‘‘Business-process-driven gray-box SOA testing’’ IBM Systems Journal, vol. 47(3), pp. 457-472, (2008).
7. Chen, T.Y., Poon, P-L. ‘‘Experience with teaching black-box testing in a computer science/software engineering curriculum’’ IEEE trans. on Education, vol. 47(1), pp. 42-50, (2004)
8. Cashin, P. M., Joliat, M. L., Kamel, R. F. ve Lasker, D. M., ‘‘Experience with a modular typed language: PROTEL’’, Proceedings of the 5th international conference on Software engineering ICSE '81, pp.136-143, (1981).