

Kullanım Modeli Baz Alınarak Markov Zincirleri ile Otomatik Testler Üretilmesi ve Yazılım Güvenilirliği Hesaplanması

Automatic Test Case Generation with Markov Chains Based on Usage Model and Software Reliability Estimation

Ceren Şahin Gebizli¹, A. Tarkan Tekcan², Mustafa Gündüzalp³

¹Vestel AR-GE Tasarım Doğrulama ve Test Bölümü,
Vestel Elektronik San ve Tic. A.Ş.
ceren.sahin@vestel.com.tr

²Vestel Tic. A.Ş.
tarkan.tekcan@vestel.com.tr

³Elektrik–Elektronik Mühendisliği Bölümü, Mühendislik Fakültesi
Yaşar Üniversitesi
mustafa.gunduzalp@yasar.edu.tr

Özet

TV ve Uydu Alıcıları, kullanıcılara sundukları yeni teknolojiler ile tüketici elektroniği pazarında önemli bir yere sahiptir ve ürün kalitesi, tüketici elektroniği pazarında önemli bir kriterdir. Ürünün kalitesi, sadece donanımsal güvenilirliğe değil, üründe kullanılan yazılımın güvenilirliğine de bağlıdır.

Testin tasarımı ve çalıştırılması birbiriyle doğrudan ilişkili ve önemli iki parçadır. Doğru tasarlanmış bir testin otomatik olarak çalıştırılması, maliyetlerin azaltulmasını, ürün kalitesinin yükseltilmesini, kaynak ve zamanın daha etkin kullanılmasını sağlar.

Otomatik test ile sınırlı bir süre içinde, önemli test senaryolarının kapsandığı testler yapılabilir.

Verimli test stratejileri sadece hata bulmaya odaklanmaz. Kullanım sıklığı daha yüksek olan yerler test edilerek, testler daha verimli hale getirilebilir. Kullanım profilini baz alan test senaryolarını kullanmak, bu noktada daha anlamlı olacaktır.

Bu çalışmada, TV'ler son kullanıcıya ulaşmadan önce, AR-GE aşamasında iken yayınlanan tüm yazılımlar ile, kullanım profili baz alınarak, testler için kullanılacak test senaryolarının Markov zincirleri kullanılarak otomatik hazırlanması, TV üzerinde bu test senaryolarının otomatik çalıştırılması ve test sonuçlarına göre test edilen yazılımın güvenilirliğinin hesaplanması anlatılmıştır.

Bu çalışma ile daha etkin, daha kapsamlı, daha spesifik test senaryoları oluşturmak ve testleri otomatik olarak çalıştırıp zamanı verimli kullanmak mümkün olmuştur.

Bu çalışmada, yazılımın işlevselliğini doğrulamaya imkan sağlayan kara kutu testi konsepti kullanılmıştır.

Anahtar kelimeler:Yazılım güvenilirliği, manuel test, otomatik test, test otomasyonu, verimli test, kullanım profili, kullanım olashkları, test senaryosu, kara kutu testi

Abstract

TVs and STBs have an important place in consumer electronics market with new technologies that they introduce to users. Quality of a product is an important criterion in consumer electronics market. Product quality is not related only the hardware reliability, it is also related with the software reliability.

Two major and related parts of testing are 'Test Design' and 'Test Execution' Automation tools cannot find bugs on their own. Quality and effectiveness of executed tests completely depends on your test design. Execution of tests that designed properly, provide cost reduction, effective time and resource usage.

It is possible to test important test scenarios in limited time via automated testing.

Efficient testing strategies do not focus on only finding defects. Tests can be performed more efficient by testing the parts that have a higher likelihood of defect. Using the test scenarios based on the usage profile will be more meaningful at this point.

In this work, it is investigated the generation of test cases automatically using Markov chains based on usage profile, run these test cases automatically on a test automation tool and according to the results of the automatic tests, finally estimation of the software reliability.

With this work, it is possible to generate more efficient, comprehensive and specific test scenarios. Also automated testing provides effective time usage.

In this paper, black box testing approach is used that allows verifying the functionality of software.

Keywords: Software reliability, manuel test, automated test, test automation, efficient test, usage profile, usage probabilities, test scenarios, black box testing.

1. Giriş

Tüketici elektroniği alanında düşük fiyatlı, yüksek kaliteli ürün beklentisi giderek artmaktadır. Müşterilerin temel beklentisi, aldıkları ürünlerin son teknolojileri içermesi, ürünlerdeki özelliklerin sorunsuz çalışması yönündedir.

Son teknolojilerin bulunduğu ürünün, pazara aynı alandaki rakiplerine göre en hızlı ve en kaliteli şekilde çıkarılabilmesi için, ürün AR-GE aşamasında birçok testten geçirilir. TV ve STB(Set Top Box)'lerde ömür testleri, stres testleri, elektriksel testler gibi donanımsal testlerin yanı sıra ağırlıklı olarak yazılım testleri yapılır. Gelişen teknolojiler ile televizyonların akıllı televizyonlar haline getirilmesi [1], gelişmiş multimedya servisleri gibi geniş kapsamlı bir çok interaktif servisler ve uygulamaları içermesi [2] (örnek; IPTV, Interaktif TV, Hybrid Broadcast Broadband TV, Pay TV ve daha birçok internet uygulamaları) sebebiyle, sistemsel yapıları daha karmaşık bir hal almıştır. Bu uygulamaların her geçen gün artması yeni ürünlerin yazılım sürümlerinin de sürekli güncellenmesi anlamına gelmektedir. Bu gibi karmaşık sistemlerin detaylı test edilmesi uzun süre gerektirir. Ürün geliştirme sürecinde test aktiviteleri toplam ürün geliştirme süresinin yüzde 30'unu kapsamaktadır.

Ürün yaşam döngüsü boyunca testin büyük önemi vardır. Genel hatları ile ürün tasarım ve test yaşam döngüsü ürün yönetim ekibinin müşteri taleplerini içeren gereksinim dökümanı yayınlanması ile başlamış olur. AR-GE aşamasında TV ile yapılan ilk test, tasarım grupları tarafından yayınlanan ürün teknik detaylarının belirtildiği döküman üzerinden yapılır. AR-GE test aşamasında belirli sayıda ürün üzerinde detaylı fonksiyonel test yapılır. Artan ürün karmaşıklığı ve proje için öngörülen zaman kısıtı nedeni ile testlerin çalıştırılma tipini manuel ya da otomatik olarak ayarlamış olmanın gerekliliği artmaktadır. Testleri otomatik çalıştırmak zaman ve kaynak kullanımında verimliliği sağlar. Manuel test olarak uygulandığında uzun zaman alacak birçok test senaryosu kısaltılır ya da atlanır. Bu da birçok hatanın AR-GE aşamasında görülmeden TV'nin üretime girmesine neden olabilir.

Otomatik test prosedürlerine geçiş sadece işgücünün etkin kullanılmasını sağlamakla kalmaz, ürünün pazara giriş süresini kısaltır, maliyetini düşürür ve kaliteyi yükseltir. Bu nedenle, test otomasyonu ve yönetimi giderek kritik ve stratejik bir zorunluluk haline gelmiştir [3]. Testlerin otomatize edilmesi hem zaman hem de kaynak kullanımında verim sağlar. Ancak esas önemli nokta testin nasıl tasarlandığıdır. Otomasyon araçları kendi başlarına hata bulamazlar. Otomasyon araçları, o araca verdiğiniz test senaryolarını otomatik olarak çalıştırmayı sağlar. Sistemdeki kritik hataların bulunması ancak iyi bir test tasarımı ile mümkündür. Kullanım profili ve olasılıklarını çıkarmak da, bize TV'de hangi bölümlerin daha fazla kontrol edilmesi

gerektiği konusunda fikir vermektedir. Test tasarımını yaparken kullanım profilini ve olasılıklarını baz almak, bizi karmaşık bir sistemde daha spesifik test alanlarına götürür. Böylece müşterilerin kullanım alışkanlıklarına göre bir test tasarımı yapmış oluruz ve müşteriler tarafından görülmesi olası hataların önceden önüne geçerek, daha kaliteli ve güvenilir yazılımlar içeren ürünler üretebiliriz.

Test senaryolarının kullanım profiline göre çıkarılması, model bazlı test tekniği kullanılarak sağlanmıştır [4]. Bu şekilde kullanıcının daha çok kullandığı televizyon özellikleri için daha fazla test senaryosu elde etmiş oluruz. Bu senaryoları da mümkün olduğunca otomatize ederek test sürecini ve kaynak kullanımını azaltırız [5].

Bir sistemin güvenilirliği, o sistemin nasıl kullanıldığı ile doğrudan ilişkilidir. Bu bağlamda otomasyon ile çalıştırılacak testler doğru tasarlanmalıdır. Doğru tasarlanan testlerin, otomasyon aracında otomatik olarak çalıştırılması ile güvenilirlik analizi yapılabilir.

Bir TV projesinde yapılacak yazılım güvenilirliği analizi ile o TV projesi sahaya, son kullanıcıya ulaşmadan önce, son kullanıcıya gidecek en son yazılım hakkında değerlendirmeler yapılır. Bu değerlendirmeler sonucunda belli bir güvenilirlik değerinin altında kalan yazılımlar ile TV'ler son kullanıcıya gönderilmez. Bu güvenilirlik değeri kritik hata sayısı, hatanın ne zaman ortaya çıktığı gibi parametrelerle ilişkilidir.

Bu çalışmada, yazılım güvenilirliğini hesaplayabilmek için, kullanım profili temel alınarak hazırlanan model ile otomatik test senaryolarının oluşturulması, yazılımın işlevselliğini doğrulamaya olanak sağlayan kara kutu test yöntemi kullanılarak test senaryolarının çalıştırılması ve test sonuçlarına göre yazılım güvenilirliğinin ölçülmesi incelenmiştir.

İlerleyen bölümlerde aşağıdaki konular açıklanacaktır;

- Gerçek kullanıcılar kullanılarak kullanım olasılıklarının belirlenmesi
- Kullanım profilinden olasılık ve olayların çıkartılarak kullanım modelinin oluşturulması. Kullanım modelinin MaTeLo [6] aracı kullanılarak tasarlanması.
- MaTeLo'da tasarlanan model üzerinden otomatik test senaryolarının oluşturulması
- Oluşturulan otomatik test senaryolarının Vestel¹ Test Otomasyon Sistemi (VesTA) üzerinde çalıştırılması.
- VesTA'da [7] çalıştırılan test adımlarının sonuçlarının MaTeLo Testor [8] programı ile yazılım güvenilirliğinin hesaplanması.

1.1 TV Test Süreci

Kullanım profili temel alınarak tasarlanan test süreci aşağıdaki bölümlere ayrılır:

- Yazılım durum geçiş diyagramının çıkarılıp, olasılık bilgilerinden de faydalanılarak TV kullanım modelinin oluşturulması
- Test senaryolarının oluşturulması:

¹ <http://www.vestel.com.tr>

- o Girişlerin belirlenmesi;
- o Olasılık dağılımına göre otomatik olarak test adımlarının belirlenmesi;
- o Beklenen sonuçların tanımlanması
- Test senaryolarının çalıştırılması
- Test çıktılarının beklenen sonuçlar ile karşılaştırılması
- Yazılım güvenilirliğinin hesaplanması,

TV test süreci, MaTeLo aracında modellenen bir testin, yazılım güvenilirlik aşamasına kadar hangi süreçlerden geçtiğini gösteren genel bir diyagram, aşağıdaki şekil (Şekil 1) ile ifade edilebilir.



Şekil 1: TV Test süreci diyagramı

2. Kullanım Profili ve Kullanım Olasılıkları Belirlenmesi

TV uygulaması ile kullanıcıların etkileşimi sistemsel olarak durum-geçiş diyagramı olarak modellenebilir. Bu diyagramda aktif kullanıcı ara yüzünü ya da o anki fonksiyonel durumu “durum” olarak tanımlanır. Kumanda ya da ön panel tuşları ile gönderilen komutlar ise “geçiş”leri ifade eder.

Özel yazılımlar aracılığı ile TV-kullanıcı etkileşimini otomatik olarak çıkarmak mümkündür. Laboratuvar içerisinde hazırlanmış kullanım senaryolarına göre gerçek kullanıcılardan alınan kullanım bilgileri bir takım olasılık bilgisini de barındırması açısından büyük önem taşır. Bu veriler ile gerçek hayata yönelik yüksek olasılıklı koşulların öncelikli test senaryosu olarak kontrol edilmesi ve ürünler seri üretim aşamasına gelmeden önce olası hataların bulunması müşteri memnuniyeti açısından büyük önem taşır.

Kullanım profili ve olasılıkların belirlenmesi 2 aşamadan oluşur [9]. İlk aşamada kullanıcıların televizyon ile etkileşimleri televizyonlarına takılan boş USB'lere kaydedilir. Kaydedilen dosyalar içinde “durum” ve “geçiş”ler bulunmaktadır. Bu durumlar kullanıcının girdiği herhangi bir menünün ya da kaynağın bilgisini içerir. Geçişler ise o menülere nasıl girildiğinin bilgisini verir (hangi kumanda komutu ya ön panel tuşu ile girildiğinin bilgisi). İkinci aşamada ise kullanıcılardan alınan

dosyalar incelenerek ve ayrıştırılarak kullanım profili ve kullanım olasılıkları çıkartılmış olur.

Yapılan çalışmalarda kullanıcı profili parametresi bulunmamaktadır. İleride yapılacak çalışmalarda farklı kullanıcı kategorisi belirlenip (yaş aralığı, meslek grubu vs.) bu kategorilere göre kullanım profili oluşturulması ve modelin kullanıcı profiline göre nasıl farklılık gösterdiği belirlenebilir.

3. Kullanım Profili ve Olasılıkları Kullanılarak Kullanım Modelinin Oluşturulması

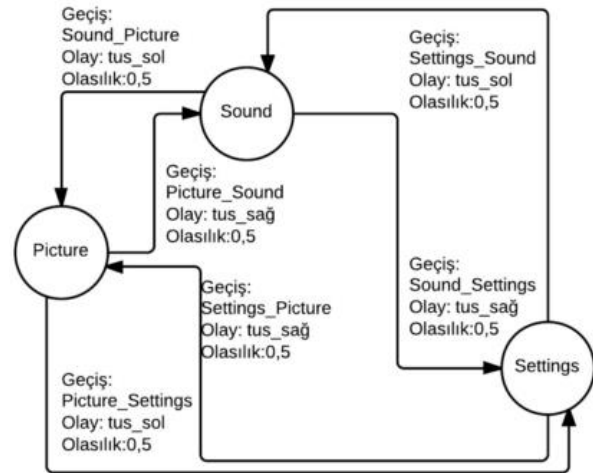
Sistem güvenilirliği analizi, sistemin kullanım modeli üzerinden yapılmalıdır. Kullanım modeli, kullanıcıların sistemi nasıl kullandığını gösterir [10][11] ve bu model kullanım profili olarak da adlandırılabilir. Kullanım profilleri, kullanım olasılıklarının durumlar arasındaki geçişlerde kullanıldığı, sonlu durum makinaları olarak tanımlanabilir.

Kullanım profili için basit bir örnek verme amacıyla hazırlanan model aşağıdaki gibidir. (Şekil 2) Verilen kullanım profili, 3 durum (Picture, Sound ve Settings) ve 6 geçiş (Picture_Sound, Sound_Picture, Sound_Settings, Settings_Sound, Picture_Settings, Settings_Picture) içermektedir.

Her bir durum aşağıdaki bilgileri içerir.

- Olası bir geçiş için bir giriş olayı.
- Bir sonraki durum bilgisi ve giriş olayının meydana gelme olasılığı.

Sistemi modellerken, daha önemli fonksiyonlar için daha yüksek geçiş olasılıkları verilir. Böylece test senaryoları oluşturulurken, yüksek olasılıklı olan geçişler daha fazla test adımında kullanılmış olur. Bu şekilde, sık kullanılan özellikler, testler sırasında daha çok kontrol edilmiş olur [10][12].



Şekil 2: Sonlu durum makinaları

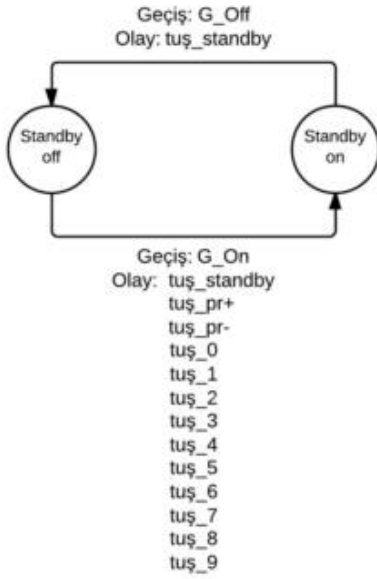
3.1 TV Modelleme

Bir TV seti modellenmesi öncelikle fonksiyonel spesifikasyon analizi ile başlar. Gerçek bir modelleme için, teste alınan sistemin tüm bileşenleri için durum, giriş, çıkış ve olasılıklar belirlenmelidir [4][12][13]. Ancak bu çalışmada kullanıcılardan alınan bilgiler doğrultusunda sistemin sadece bir kısmı için ya-

pılan modelleme ele alınacaktır. (Deneysel sonuçlar bölümünde açıklanmıştır.)

Basit bir model tablo1’de gösterilmiştir. 2 durum ve 2 geçiş içermektedir. Bir durum TV’deki herhangi bir durumun yerini belirler. (Açma/kapama, menüye girme vb.) Geçişler bir durumdan başka bir duruma geçmek için kullanılır. (TV açma durumundan, tv kapama durumuna geçilmesi) Bir geçiş bir ya da birden fazla olay içerebilir.

Şekil 3’te durumlar arası geçişleri gösteren TV açma ve kapatma modeli (“Standby off” / “on”) verilmiştir.



Şekil 3: TV açma ve kapatma (Standby off/on)

“Standby off” durumundan “on” durumuna geçmek için (“standby”, “pr+”, “pr-”, “0”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”) olaylarından biri kullanılabilir. Ancak standby off durumundan on durumuna geçebilmek için yalnızca bir olay vardır. (“standby”)

Tablo 1: Basit bir model için durum tablosu

Kaynak Durumu	Geçiş	Olay	Hedef Durum
STANDBY_OFF	G_On	tuş_standby	ON
		tuş_0	
		tuş_1	
		tuş_2	
		tuş_3	
		tuş_4	
		tuş_5	
		tuş_6	
		tuş_7	
		tuş_8	
		tuş_9	
		tuş_pr+	
		tuş_pr-	
ON	G_Off	tuş_standby	STANDBY_OFF

3.2 Markov Zinciri

Belirtilen TV modeli davranışı için Markov zinciri metodu kullanılmıştır [14].

Başarısız durumların istatistiksel analizi ya da güvenilirlik değerlendirmesi söz konusu olduğunda Markov zinciri uygun bir metottur [15]. Bir markov zinciri N durumlarının bir kümesi olarak ifade edilebilir.

$$S = \{S_1, S_2, \dots, S_n\} \quad (1)$$

Süreç başlangıç durumundan başlar ve diğer duruma geçer. Her geçiş bir adım olarak adlandırılır. Eğer mevcut durum S_i ise, bu mevcut durum olan i 'ye bağlı olan olasılık ile S_j durumuna geçer.

$$S(X_{n+1} | X_1, X_2, \dots, X_n) = S(X_{n+1} | X_n) \quad (2)$$

Bu olasılık p_{ij} ifade edilir ve bu geçiş olasılığı olarak adlandırılır. Eğer süreç aynı durumda kalacaksa bu kez olasılık p_{ii} olarak ifade edilir. Markov zinciri $N \times N$ geçiş olasılık P matrisinde gösterilir. Bu matrisin her satırındaki değerlerin toplamı 1'e eşittir.

$$\forall i \sum_{j=1}^N P_{ij} = 1 \quad (3)$$

Tablo1’deki kullanım modeli durum tablosu temel alınarak her geçiş için olasılıklar da eklenir. Bir durumdan diğer bir duruma geçişte toplam kaç geçiş varsa bu geçişlerin olasılıklarının toplamı 1’e eşit olmalıdır.

Her bir geçiş için de, kaç adet olay kullanılıyorsa bu olayların da olasılıkları toplamı 1’e eşit olmalıdır. (3).

Olasılıkların doğru dağılımı için gerçek kullanıcılardan alınan bilgiler kullanılmalıdır. (Deneysel sonuçlar bölümünde gerçek kullanıcıların tecrübeleri kullanarak hazırlanan model bulunmaktadır.)

Tablo1’de verilen modelde kullanılan geçişler ve olaylar için olasılık dağılımı örneği tablo2’de gösterilmiştir.

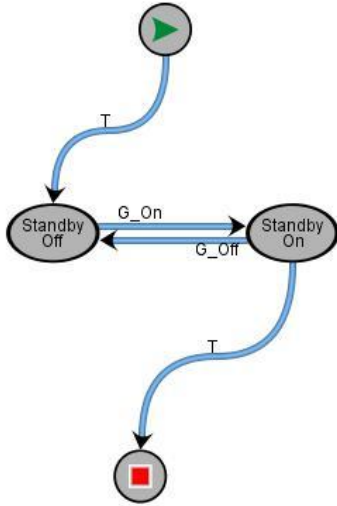
Tablo 2: Olasılık dağılımı

Kaynak Durumu	Geçiş	Geçiş Olasılığı	Olay	Olay Olasılığı	Hedef Durum
STANDBY_OFF	G_On	1	tuş_standby	0,1	ON
			tuş_0	0,06	
			tuş_1	0,15	
			tuş_2	0,12	
			tuş_3	0,06	
			tuş_4	0,06	
			tuş_5	0,06	
			tuş_6	0,06	
			tuş_7	0,06	
			tuş_8	0,06	
			tuş_9	0,06	
			tuş_pr+	0,075	
			tuş_pr-	0,075	
ON	G_Off	1	tuş_standby	1	STANDBY_OFF

Tablo2’de hazırlanan bu durum tablosu, Şekil 4’te görüldüğü gibi MaTeLo Usage Model Editor kullanılarak tasarlanır [16].

Aşağıdaki şekilde; yeşil ok bulunan durum başlangıç durumu, kırmızı karenin bulunduğu durum ise bitiş durumunu ifade etmektedir. Kullanıcı davranışlarına göre hesaplanan olasılık

bilgilerini içeren yay oklar, TV durum geçişlerini ifade etmektedir. Geçişlere “delay”, erteleme bilgisi de verilebilmektedir. Bir durumdan diğerine geçerken kaç saniye sonra geçmesi gerektiği bilgisi geçişlere atanabilmektedir.



Şekil 4: MaTeLo Usage Model Editor'de modellenme

4. Kullanım Modeli İle Otomatik Test-Senaryolarının Oluşturulması

MaTeLo ile yapılan modelleme sonrası MaTeLo Testor kullanılarak yüzlerce test senaryosu oluşturulabilir [16][17]. MaTeLo Testor'un çıktısı olarak, TV durum ve geçiş bilgilerini içeren, test senaryolarının bulunduğu XML dosyası yayınlanır. Vestel Test Otomasyon aracında (VesTA) bu test senaryolarını kullanabilmek için, XML dosyası özel olarak hazırlanan bir yazılım aracılığı ile test scriptlerine dönüştürülür. Bu scriptler, TV uzaktan kumanda otomatik navigasyonu, ses/görüntü yakalama/işleme ve karşılaştırma gibi alt işlemlerin bilgilerini içerir.

Test senaryoları modelleme sırasında model üzerinde ayarlanan kullanım olasılıklarına göre oluşturulur. MaTeLo Testor Part ile test senaryoları oluşturulurken başlangıç durumundan başlanır, verilen olasılıklar dâhilinde devam edilir ve bitiş durumu ile bitirilir.

MaTeLo Testor Part ile test senaryosu oluşturmak için kullanılacak 4 algoritma [8] bulunmaktadır.

- Olasılığı en yüksek oluşum yaklaşımı: Olasılığı en yüksek geçişler değerlendirilerek test senaryoları oluşturulur. Test senaryosu karmaşıklığı basittir. Kapsam hedefi nominal senaryolardır.
- Risk bazlı yaklaşım: Karmaşıklığı en yüksek yerler değerlendirilerek test senaryoları oluşturulur. Test senaryosu karmaşıklığı yüksektir. Kapsam hedefi risktir.
- Arcs Kapsamı yaklaşımı: Gereksinimler değerlendirilerek test senaryoları oluşturulur. Test senaryosu karmaşıklığı yüksek ve birbirini etkileyecek şekildedir. Kapsam hedefi gereksinimlerdir.
- Kullanım-odaklı: Operasyonel kapsam değerlendirilerek, kullanıcıların davranışlarına uygun test senaryoları

oluşturulur. Test senaryosu karmaşıklığı birbirini etkileyecek şekildedir. Kapsam hedefi güvenilirliktir.

Bir test senaryosu test girişleri sıraları(test input sequences), TV'de alınacak aksiyonlar(durumlar, geçişler) ve beklenen çıkış sıralarını(expected output sequences) içermektedir.

Bu test senaryolarının XML halinde yayınlanmasının nedeni bir çok farklı test otomasyon aracında çalıştırılabilir olmasıdır.

Kullandığımız sistemde, bu XML dosyalarının VesTA'da çalıştırılabilmesi için yine AR-GE'de, kendi geliştirdiğimiz bir çevirici program kullanılmaktadır. XML çıktısı olarak aldığımız test senaryolarını bu program ile VesTA'da çalıştırılabilecek şekilde düzenleyebilmekteyiz. Aynı zamanda bu XML dosyaları, manuel testler için de kullanıma uygundur.

Şekil 5'te XML dosyası, bir web tarayıcı aracılığıyla açıldığında, o XML dosyasının herhangi bir test senaryosu hakkında nasıl bir bilgi verdiği görülmektedir. Şekil 5'te 5 adımdan oluşan bir test senaryosu bulunuyor. Bu senaryoda;TV'ye güç veriliyor, kumandadaki numerik tuşlardan “6” tuşu ile TV açılıyor, kumanda üzerindeki “source” tuşu ile kaynaklar listesi açılıyor, listedeki ilk kaynak tipi seçiliyor ve ardından TV “standby” tuşu ile kapatılıyor.

Step	Event/Input
1	Plug on Event : Plug On E_Power_Connection : Power_on
2	Tuning On with Digits Input : Digits (0-9) Key E_Digit : 6
3	Pressing Source
4	Pressing OK Input : Digits (0-9) Key E_Navigation_Key : OK_Select
5	Tuning off with Standby key Input : Digits (0-9) Key E_Digit : Stby_Key

Şekil 5: XML çıktısı

5. Otomatik Test Senaryolarının VesTA'da Çalıştırılması

TV fonksiyonel testinde, genellikle manuel, yarı otomatik ya da tam otomatik test şemalarının kullanıldığı kara kutu testi yaklaşımı kullanılır [3]. Donanım ve yazılımın karmaşık bir kombinasyonu oluşturan TV için en uygun ve en verimli test yaklaşımı kara kutu testidir. Kara kutu testi sadece girdi, çıktı ve mevcut TV durumunu göz önünde bulundurduğu bir test yaklaşımıdır [18][19].

Otomasyon sistemi olarak Vestel Otomasyon Sistemi (VesTA) kullanılmıştır. VesTA otomasyon aracı, ürünün hızlı ve güvenilir bir doğrulamadan geçirilmesini sağlar.

VesTA TV fonksiyonel doğrulaması için kullanılır. VesTA sistemi, yarı otomatik test şemasını kullanır.

VesTA'da, TV tasarım spesifikasyonlarına göre, oluşturulan test scriptlerine dayalı bir otomatik test gerçekleştirilir. TV tasarım spesifikasyonu test edilen TV için girdileri belirler. Test girdileri temel alınarak, test edilen TV için çıktı verileri oluşturulur. Bu veriler, test edilen TV davranışının doğruluğunu belirlemek

için beklenen sonuçlar ile karşılaştırılarak oluşturulmuştur. VesTA test otomasyon aracı test için gerekli tüm donanımsal ve yazılımsal modulleri içerir; ses/görüntü yakalama/işleme cihazı, TV uzaktan kumanda emulatörü ve işlemlerin gerçekleştirileceği istasyon.

MaTeLo Testor Part ile en uygun algoritma seçilerek otomatik olarak oluşturulan test senaryoları, VesTA otomasyon sistemi aracılığı ile çalıştırılır. Bölüm 5'te MaTeLo Testor Part'in test senaryolarını XML formatında oluşturduğunu ve özel bir yazılım ile bu XML'in test scriptlerine dönüştürüldüğünü belirtmiştik. Test scriptine dönüştürülen bu test senaryoları, VesTA otomasyon aracının veritabanına eklenir ve bu test senaryoları için test planı oluşturulur. VesTa'da sırasıyla aşağıdaki adımlar izlenerek otomatik olarak hazırlanan test senaryoları TV üzerinde otomatik olarak çalıştırılır.

- Yeni bir test paketi oluşturulur.
- Otomatik olarak oluşturulan test senaryoları test paketine eklenir.
- Bu test senaryoları için yeni bir test planı oluşturulur.
- Önce test planı bir kez çalıştırılır ve referans resimler alınır.
- Test senaryoları karşılaştırma için tekrar çalıştırılır.

Referans resimler ile gerçek çıktılar karşılaştırma sonuçları (Pass/Fail) otomatik olarak yayınlanır. Bu sonuçlar daha sonra yazılım güvenilirliği hesabı için kullanılacaktır.

6. Otomatik Test Sonuçları ile Yazılım Güvenilirliği Hesaplanması

Yazılımın müşteri beklentilerini net bir şekilde karşılayamamasının neden olduğu maliyetler, ürünün ve şirketin başarısına yönelik negatif bir durum oluşturmaktadır. Bu durum daha güvenilir yazılımlara olan gereksinimi artırmaktadır [20][21]. Üretimde kullanılacak yazılım versiyonunun güvenilirlik hesabına göre, o yazılım versiyonunun üretimde kullanılıp kullanılmayacağı değerlendirilmelidir.

VesTA'da çalıştırılan test sonucunda hangi test senaryosunun geçtiği, hangi test senaryosunun başarısızlıkla sonuçlandığı, başarısız sonuçlanan test senaryosunun hangi test adımında başarısız olduğu, hangi saatte başarısız olduğu ve toplam testin kaç saat sürdüğü gibi bir takım bilgiler edinilir. Bu bilgiler yazılım güvenilirlik hesaplaması için kullanılacak önemli parametrelerdir.

Güvenilirlik hesaplaması için kullanılan MaTeLo Testor Part açık kaynaklı değildir, bu sebeple güvenilirlik hesabı için kullanılan metrikler detaylı bilinmemektedir. Yazılım güvenilirliği hesaplaması için kendi yaklaşımımız şu anda bulunmamakla birlikte bu konuda AR-GE Test grubumuzda çalışmalar hala devam etmektedir.

Yazılım güvenilirliği için MaTeLo'nun Testor Part'a geri dönüp bu sonuçlar işlendiğinde [22][23], test edilen yazılım sürümü için güvenilirlik hesaplaması yapılır. Bir başka yazılım ile aynı TV için, aynı test senaryoları çalıştırılıp, alınan sonuçlar Testor'da bir önceki yazılım versiyonunun test sonuçları ile karşılaştırılır ve iki yazılım arasında güvenilirlik analizi yapılabilir.

Güvenilirlik, bir süre periyodu boyunca, aktivasyon sayısına göre hesaplanır. MaTeLo Testor Part'a yüklenen test senaryo sonuçları, analizi ile güvenilirlik hesaplaması yanı sıra, MTTF (mean time to failure- ortalama hata oluşma zamanı) ve hata yoğunluğu (failure intensity) hakkında da bilgi vermektedir.

MTTF (ortalama hata oluşma zamanı) yazılım güvenilirliğini ölçmeye yarayan metriklerden biridir. Bu değer belirli deneysel koşullar altında bir hata ortaya çıkmadan önce geçen ortalama zamanı vermektedir [24].

Hata yoğunluğu [25]; bilinen hata sayısının koşulan test adımları sayısına oranıdır.

7. Deneysel Sonuçlar

Bu çalışmada, kullanım profili belirlemek ve kullanım olasılıklarını çıkarabilmek için gerçek kullanıcıların kullanım tecrübelerinden yararlanıldı.

30 kullanıcıya 30 adet aynı model, aynı özelliklerin bulunduğu özdeş TV'ler dağıtıldı. Bu TV'lerin USB kaynaklarına birer boş USB takıldı. Kullanıcılardan 10 gün boyunca TV'yi her zamanki gibi kullanmaları istendi.

Bu süre zarfında kullanıcıların TV ile etkileşimi bu USB'lere otomatik olarak kaydedildi. 10 günün sonunda TV üzerindeki USB'ler toplandı. Kaydedilen veriler işlendi ve böylece kullanıcıların TV'nin hangi durumlarını, ne kadar sıklıkla kullandığı bilgisi alındı.

Gerçek kullanıcılardan alınan bilgiler kullanım modelinin oluşturulmasında rol aldı.

Örnek olarak verilen TV için, girdi, çıktı ve olasılık bilgilerini, gerçek kullanıcı davranışlarını içeren tablo aşağıdaki gibidir:

Tablo 3: Gerçek kullanıcı davranışları modeli

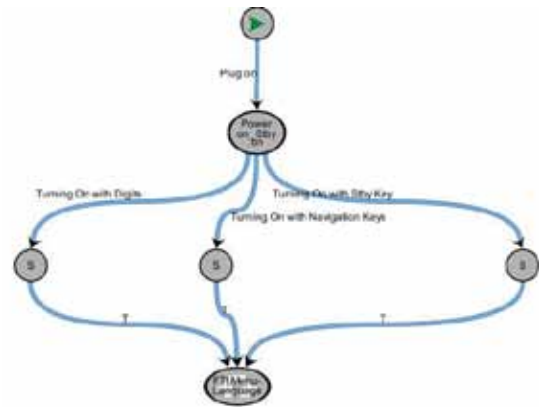
Kaynak_Durum	Geçiş	Geçiş_Olasılığı	Olay	Olay_Olasılığı	Hedef_Durum															
Initial State(Power off)	Plug on	1,0	Power_on	1,0	Power on Stbyon															
Power on_Stbyon	Turning on with Digit keys	0,3	0	0,1	RF Receive Mode															
			1	0,1																
			2	0,1																
			3	0,1																
			4	0,1																
			5	0,1																
			6	0,1																
			7	0,1																
			8	0,1																
			9	0,1																
Turning on with Navigation keys	0,3	Arrow Up	0,25	RF Receive Mode																
		Arrow Down	0,25																	
		Prog Up	0,25																	
		Prog Down	0,25																	
		Turning on with Stby key	0,4		Stby_key	1														
Turning Off with Stby	0,375	Stby_Key	1	1	Final State(Standby)															
						Pressing menu	0,1	Menu	1	menu cond										
											Pressing Source	0,125	Source	1	source cond					
																RF Receive Mode	0,4	Pressing Digit Keys	0,1	RF Receive Mode
1	0,1																			
2	0,1																			
3	0,1																			
4	0,1																			
5	0,1																			
6	0,1																			
7	0,1																			
8	0,1																			
9	0,1																			

					TV
					EXT1
					EXT2
					EXT2-S
					SIDE AV
					HDMI 1
					HDMI 2
					HDMI 3
					HDMI 4
					YPbPr
					VGA/PC
source_cond					TV mode
TV	Pressing OK	0.4	OK Select	1	TV mode
	Pressing Down	0.3	Arrow Down	1	EXT1
	Pressing Up	0.3	Arrow Up	1	VGA/PC
EXT1	Pressing OK	0.4	OK Select	1	EXT1 mode
	Pressing Down	0.3	Arrow Down	1	EXT2
	Pressing Up	0.3	Arrow Up	1	TV
EXT2	Pressing OK	0.4	OK Select	1	EXT2 mode
	Pressing Down	0.3	Arrow Down	1	EXT2-S
	Pressing Up	0.3	Arrow Up	1	EXT1
EXT2-S	Pressing OK	0.4	OK Select	1	EXT2-S mode
	Pressing Down	0.3	Arrow Down	1	SIDE AV
	Pressing Up	0.3	Arrow Up	1	EXT2-S
SIDE AV	Pressing OK	0.4	OK Select	1	SIDE AV mode
	Pressing Down	0.3	Arrow Down	1	HDMI 1
	Pressing Up	0.3	Arrow Up	1	EXT2-S
HDMI 1	Pressing OK	0.4	OK Select	1	HDMI 1 mode
	Pressing Down	0.3	Arrow Down	1	HDMI 2
	Pressing Up	0.3	Arrow Up	1	SIDE AV
HDMI 2	Pressing OK	0.4	OK Select	1	HDMI 2 mode
	Pressing Down	0.3	Arrow Down	1	HDMI 3
	Pressing Up	0.3	Arrow Up	1	HDMI 1
HDMI 3	Pressing OK	0.4	OK Select	1	HDMI 3 mode
	Pressing Down	0.3	Arrow Down	1	HDMI 4
	Pressing Up	0.3	Arrow Up	1	HDMI 2
HDMI 4	Pressing OK	0.4	OK Select	1	HDMI 4 mode
	Pressing Down	0.3	Arrow Down	1	YPbPr
	Pressing Up	0.3	Arrow Up	1	HDMI 3
YPbPr	Pressing OK	0.4	OK Select	1	YPbPr mode
	Pressing Down	0.3	Arrow Down	1	VGA/PC
	Pressing Up	0.3	Arrow Up	1	HDMI 4
VGA/PC	Pressing OK	0.4	OK Select	1	VGA/PC mode
	Pressing Down	0.3	Arrow Down	1	TV
	Pressing Up	0.3	Arrow Up	1	YPbPr
TV mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
EXT1 mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
EXT2 mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
EXT2-S mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
SIDE AV mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
HDMI 1 mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
HDMI 2 mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
HDMI 3 mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
HDMI 4 mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
YPbPr mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
VGA/PC mode	Turning Off with Stdby	1	Stby Key	1	Final State(Standby)
					Picture
					Sound
					Settings
					Install and Retune
					Channel List
menu_cond					Media Browser
	Pressing Right	0.3	Arrow Right	1	Sound
	Pressing Left	0.3	Arrow Left	1	Media Browser
	Pressing OK	0.3	OK Select	1	Picture Menu
	Turning Off with Stdby	0.1	Stby Key	1	Final State(Standby)
Picture	Pressing Right	0.3	Arrow Right	1	Settings
	Pressing Left	0.3	Arrow Left	1	Picture
	Pressing OK	0.3	OK Select	1	Sound Menu
	Turning Off with Stdby	0.1	Stby Key	1	Final State(Standby)
Sound					

	Pressing Right	0.3	Arrow Right	1	Install and Retune
	Pressing Left	0.3	Arrow Left	1	Sound
	Pressing OK	0.3	OK Select	1	Settings Menu
	Turning Off with Stdby	0.1	Stby Key	1	Final State(Standby)
Settings	Pressing Right	0.35	Arrow Right	1	Channel List
	Pressing Left	0.35	Arrow Left	1	Settings
	Pressing OK	0.2	OK Select	1	Install and Retune Menu
	Turning Off with Stdby	0.1	Stby Key	1	Final State(Standby)
Install and Retune	Pressing Right	0.2	Arrow Right	1	Media Browser
	Pressing Left	0.2	Arrow Left	1	Install and Retune
	Pressing OK	0.5	OK Select	1	Channel List Menu
	Turning Off with Stdby	0.1	Stby Key	1	Final State(Standby)
Channel List	Pressing Right	0.2	Arrow Right	1	Picture
	Pressing Left	0.2	Arrow Left	1	Channel List
	Pressing OK	0.5	OK Select	1	Media Browser Menu
	Turning Off with Stdby	0.1	Stby Key	1	Final State(Standby)
Media Browser					

Tablo 3 baz alınarak, MaTeLo Usage Model Editor’de model tasarlandı.

Model TV’nin açılması ile başlar ve kullanıcı, TV’nin ilk kurulum menüsü ile karşılaşır.

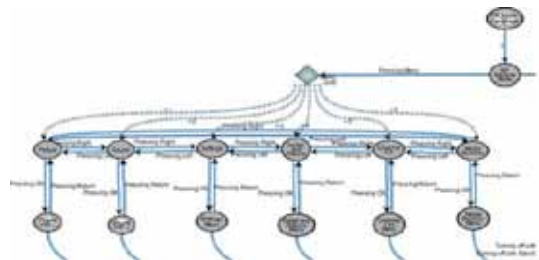


Şekil 6: TV açıldıktan sonra ilk kurulum menüsü modellenmesi

Yukarıdaki şekilde “FTI language menu” kendi içinde altzincirler içerdiği için diğer durumlardan farklı bir yapıda olduğu görülmektedir. FTI menüsü “ilk kurulum menüsünün” kısaltma adıdır. Ve bu menüde kendi içinde alt model içermektedir. Bu tarz alt modele gelindiğinde, test senaryoları oluşturulurken bu alt modelin içinde de dolaşılır.

Kurulum tamamlandıktan sonra, kullanıcıların günlük televizyon izleme alışkanlıklarından elde edilmiş veriler kullanılarak, TV menüsü ve TV haricinde diğer kaynakların (HDMI kaynağı, PC kaynağı vs.) kullanım sıklığı göz önüne alınarak model tamamlanmıştır.

Şekil 7’de menü kullanımının modellenmesi gösterilmiştir.



Şekil 7: TV Menüsünün modellenmesi

Model tasarlandıktan sonra MaTeLo Testor Part kullanılarak Test senaryoları otomatik olarak oluşturuldu. Test senaryosu oluşturmak için, Testor'daki kullanım odaklı test senaryosu oluşturma algoritması kullanıldı. Böylece test senaryoları kullanıcılarından alınan bilgiler doğrultusunda kullanılan (set edilen) olasılıklar temel alınarak oluşturulmuş oldu.

Tüm model için oluşturulan test senaryoları scriptlere dönüştürüldü ve TV'ye iki farklı yazılım versiyonu yüklenerek VesTA'da çalıştırıldı. İlk versiyonda 2 hata bulundu. İlk hatadan biri testin başlamasından 10 dk sonra bulundu. Diğer hata ise ilk hata bulunduktan 20 dk sonra bulundu. Test 3 saat 45 dk sürdü. İkinci versiyonda ise 1 hata bulundu. Bulunan hata testin 55. dakikasında bulundu. Test 3 saat 35 dk sürdü.

VesTA'da 200'er test senaryosu çalıştırıldıktan sonra test raporu yayınlandı. Bu raporda hangi test senaryosunun geçtiği, hangi test senaryosunun kaldığı bilgisi bulunmaktadır.

Tablo 4: VesTA test raporu örneği

Execution plan	Project	Test	Execution result	Execution date	Tester	Comments	Details
MB65 MaTeLo User Model Automatic Cases V0.1.0	MB65						
		TV_MaTeLo_INIT	INCONCLUSIVE	Mar 28 2012 11:33AM	Ceren Sahin		Details
		TV_MaTeLo_001_001_001	PASS	Mar 28 2012 11:39AM	Ceren Sahin		Details
		TV_MaTeLo_001_001_002	PASS	Mar 28 2012 11:46AM	Ceren Sahin		Details
		TV_MaTeLo_001_001_003	PASS	Mar 28 2012 11:47AM	Ceren Sahin		Details
		TV_MaTeLo_001_001_004	PASS	Mar 28 2012 11:53AM	Ceren Sahin		Details
		TV_MaTeLo_001_001_005	PASS	Mar 28 2012 11:57AM	Ceren Sahin		Details
		TV_MaTeLo_001_001_006	PASS	Mar 28 2012 11:59AM	Ceren Sahin		Details
		TV_MaTeLo_001_001_007	PASS	Mar 28 2012 12:03PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_008	PASS	Mar 28 2012 12:06PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_009	PASS	Mar 28 2012 12:06PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_010	PASS	Mar 28 2012 2:01PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_011	PASS	Mar 28 2012 2:10PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_012	PASS	Mar 28 2012 2:13PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_013	PASS	Mar 28 2012 2:16PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_014	PASS	Mar 28 2012 2:19PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_015	PASS	Mar 28 2012 2:25PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_016	PASS	Mar 28 2012 2:35PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_017	FAIL	Mar 28 2012 2:38PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_018	PASS	Mar 28 2012 2:50PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_019	PASS	Mar 28 2012 2:52PM	Ceren Sahin		Details
		TV_MaTeLo_001_001_020	PASS	Mar 28 2012 3:04PM	Ceren Sahin		Details

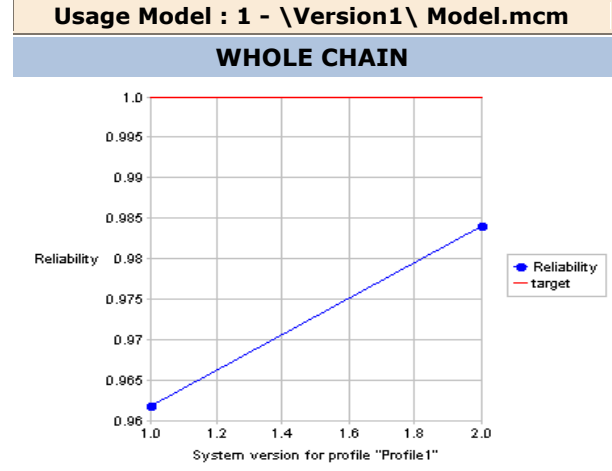
Tablo 5: Başarısız olan test senaryosu için test adımları raporu örneği

Step result	Execution date time	Comments	Vesbox ip
PASS	1.20.2012 02:38:09 PM		192.168.2.237
PASS	1.20.2012 02:38:27 PM		192.168.2.237
FAIL	1.20.2012 02:38:46 PM		192.168.2.237

Başarısız olan test senaryoları için detay bölümüne girildiğinde hangi test adımında kaldığı bilgisi tutulmaktadır.

Geçme/Kalma durumu ve kalan test senaryosu için hangi test adımında kaldığı bilgisi MaTeLo Testor partının sistem analiz bölümüne girilerek test edilen yazılım için güvenilirlik bilgisi elde edildi [23].

V0.1 ve V0.2 yazılımlarının güvenilirlik karşılaştırması sonucunda aşağıdaki veriler elde edildi:

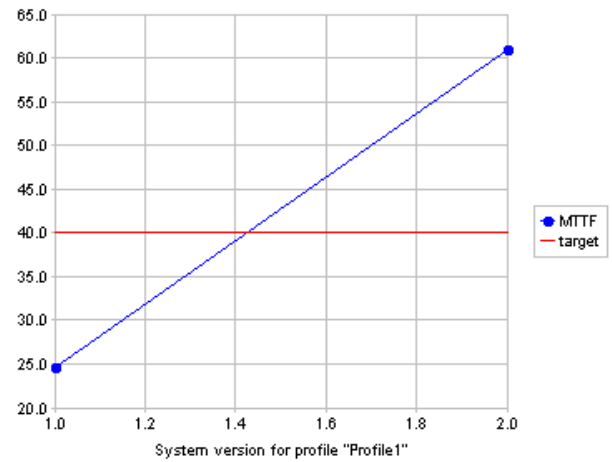


Şekil 8: İki yazılım arasında güvenilirlik analizi

Güvenilirlik, bir zaman periyodu süresince, aktivasyon sayısına göre hesaplanmıştır.

Tablo 6: Güvenilirlik ve MTTF sayısal değerleri

Reliability/MTTF for profile "Profile1" (in days)			
System version	Version label	Reliability (1.000000)	MTTF (40.00)
1	V0.1	0.961730 ^t	24.639.842
2	V0.2	0.983990 ^t	60.963.946

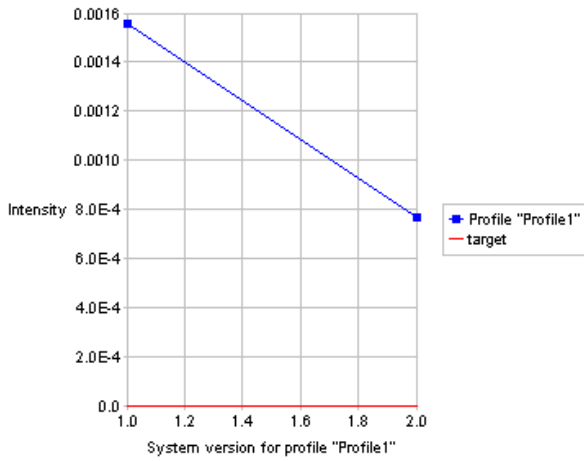


Şekil 9: İki yazılım arasında MTTF analizi

MTTF (ortalama hata oluşma zamanı) bir hata ortaya çıkmadan önce geçen ortalama zamanı vermektedir.

Tablo 7: Hata yoğunluğu sayısal değerleri

Intensity		
System version	Version label	Intensity (0.000000)
1	V0.1	0.001558
2	V0.2	0.000766



Şekil 10: İki yazılım arasında Failure Intensity analizi

Bu sayı, bulunan hataların sayısının, çalıştırılan test adımlarına oranıdır. Bu sayı normalde düşmelidir.

8. Sonuçlar

Bilinen testler ve eskiden manuel olarak çalıştırılan testler otomatize edilip zamandan ve kaynaktan kazanç sağlamayı hedefleyen yöntemler bazen son kullanıcının karşılaşılabileceği kritik hataları bulamayabilir.

Otomatik testler, otomatize edilebilen, üründe otomatik olarak çalıştırılabilecek testlerdir. Önemli olan, otomatize etmekten çok neyin otomatize edildiğidir. Bu sebeple testin tasarımı, bir testin oluşturulurken kullanılan tekniği, kritik hataların bulunmasında önemli ölçüde rol oynar.

Biz çalışmamızda testi, kullanıcılardan aldığımız kullanım profiline uygun, son kullanıcılar tarafından karşılanması daha muhtemel hataları bulmaya yönelik tasarladık. Bu şekilde hem sahada oluşabilecek kritik hataların önüne geçmeyi hedefledik, hem de modelleme sonucu otomatik olarak oluşturulan binlerce test senaryosu ile yazılım güvenilirlik hesaplaması yapabildik.

Otomatik oluşturulan test senaryolarını otomatik olarak çalıştırarak hem zaman hem de kaynak kullanımında verimlilik sağladık.

Yazılım güvenilirliği hesabı ile de, üretime verilecek yazılım ile ilgili bilimsel bir değer ortaya çıkararak, hem test hem de onay grubu olarak, hangi yazılıma neden üretim onayı verip vermediğimizi matematiksel olarak ifade edebildik.

Yazılım güvenilirliği ile ilgili hesaplama, Vestel'de geliştirilmeyen, dışarıdan bir araç ile sağlanmıştır. Şu anda yazılım gü-

venilirliği hesaplaması için Vestel Elektronik AR-GE Test grubunda çalışmalar devam etmektedir. Bu çalışma aynı zamanda yüksek lisans tez çalışması olarak devam etmektedir.

9. Kaynaklar

- [1] Y. Wu, S. J. Hirakawa, and U. H. Reimers, "Overview of Digital Television Development Worldwide", Proceedings of the IEEE, Vol. 94, No.1, pp.8 – 21, Jan. 2006.
- [2] C.-C. Lin and M.-S. Chen, "Controlling digital TV set-top box with mobile devices via an IP network", IEEE Transactions on Consumer Electronics, Vol. 52, No. 3, pp. 935 – 942, Aug 2006.
- [3] D. Marijan, V. Zlokolica, N. Teslic, V. Pekovic, and T. Tekcan, "Automatic functional TV set failure detection system", IEEE Transactions on Consumer Electronics, vol. 56, no. 1, pp. 125-133, 2010.
- [4] H. Shokry and M. Hinchey, "Model-based verification of embedded software", IEEE Computer Magazine, pp. 53-59, Apr. 2009.
- [5] J. Boberg, "Early fault detection with model-based testing", Proceedings of the 7th ACM SIGPLAN workshop on ERLANG, pp. 9-20, 2008.
- [6] MaTeLo, Model Based Testing tool, <http://www.all4tec.net/>
- [7] VesTA, "Vestel Test Automation Tool", designed by Vestel Electronics R&D, 2009.
- [8] MaTeLo "Test Strategies", [all4tec.net/index.php/en/model-based-testing/33-matelo-test-strategies](http://www.all4tec.net/index.php/en/model-based-testing/33-matelo-test-strategies)
- [9] P.Unlubas, E.Domekeli, D.Kurt "Usage Profile Based Test Case Generation", Final Project for Master Degree, Ozyegin University, 2011.
- [10] C. Nebut, F. Fleurey, Y. Le Traon, and J.M. Jezequel, "Automatic test generation: a use case driven approach", IEEE Transactions on Software Engineering, vol. 32, no. 3, pp. 140-155, 2006.
- [11] K. Weyns and P. Runeson, "Sensitivity of software system reliability to usage profile changes", Proceedings of the ACM symposium on Applied computing, 2007.
- [12] K.-K. Cheong, I. Kim, S.-K. Park, and Y.-J. Park, "User Performance Measures for Evaluating Interactive TV Pointing Devices", IEEE Transactions on Consumer Electronics, Vol. 57, No. 3, pp. 1236-1244, August 2011.
- [13] J. Philipps, A. Pretschner, O. Slotosch, E. Aiglstorfer, S. Kriebel, K. Scholl, "Model-Based Test Case Generation for Smart Cards", Elsevier Electronic Notes in Theoretical Computer Science 80, 2003.
- [14] J. A. Whittaker and M. G. Thomason, "A Markov Chain Model for Statistical Software Testing", IEEE Transactions on Software Engineering Vol. 20, NO. 10, October 1994.
- [15] S. J. Prowell, "Using Markov Chain Usage Models to Test Complex Systems", Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

- [16] A. Guiotto, B. Acquaroli, A. Martelli, “MaTeLo: Automated Testing Suite for Software Validation”, Proceedings of DASIA 2003 Jun. 2003.
- [17] W. Dulz, Z. Fenhua, “MaTeLo – Statistical Testing Using Annotated Sequence Diagrams, Markov Chains and TTCN-3”, IEEE International Conference on Quality Software, Nov. 2003.
- [18] V. Pekovic, N. Teslic, I. Resetar, and T. Tekcan, “Test management and test execution system for automated verification of digital television systems”, IEEE 14th International Symposium on Consumer Electronics, 2010
- [19] B. Beizer, “Black-box testing: techniques for functional testing of software and systems,” Wiley, 1995.
- [20] M. S. Hosain and M. S. Alam. “Software reliability using Markov Chain usage model”, 3rd International Conference on Electrical & Computer Engineering, ICECE 2004, 28-30 December 2004.
- [21] X. Zhang, M.Sharma, P. Franklin, “Evaluating System Reliability from the Customer Perspective To Improve Availability Predictions”, RAMS 2005, 0-7803-8824-0/05/IEEE 2005.
- [22] MaTeLo “MaTeLo Report Management” http://www.all4tec.net/wiki/index.php?title=%E2%80%9CReport_Management%E2%80%9D_Tab
- [23] MaTeLo “MaTeLo Testor” http://www.all4tec.net/wiki/index.php?title=Appendices_of_user_manuel_testor
- [24] Musa, J. D., “A theory of software reliability and its application.”, IEEE Transactions on Software Engineering; 1975, Vol. SE-1 Issue 3, p312-327, 16p.
- [25] S. S. Gokhale, “Software failure intensity, reliability and optimal stopping time incorporating repair policies”, International Journal of Reliability, Quality & Safety Engineering, Vol. 13, No. 5 p455–477, October 1, 2006.



Ceren Şahin Gebizli

Ceren Şahin Gebizli Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Haziran 2010 mezunudur. Özgeğin Üniversitesi Bilgisayar Mühendisliği Bölümü'nde tez aşamasında yüksek lisans öğrencisidir. 2011 yılında ISTQB-CTFL (Farkındalık seviyesi test uzmanı) ve 2013 yılında da ISTQB CTAL-TA (Uzman seviye test analisti) sertifika ve ünvanlarını almıştır. 2012 Eylül ayında Tallinn-Estonya'da Model Bazlı Testler konferansına katılmıştır. 2013 Mayıs ayında Test İstanbul 2013 konferansında, geleceğin test teknikleri ve test otomasyonu konusunda konuşmacı olarak yer almıştır. 2013 Eylül ayında UYMS 2013 sempozyumunda test tasarımı konusunda konuşmacı olarak yer almıştır. 2010 Temmuz ayından beri Vestel Elektronik AR-GE'de Tasarım Doğrulama ve Test Mühendisi olarak görev almaktadır. Yazılım ve Donanım grupları ile birlikte TV projelerinin test aktiviteleri ve test yönetimi üzerinde çalışmaktadır. Araştırma ve uzmanlık alanları içerisinde test tasarım teknikleri, model bazlı testler, tecrübeye dayalı testler ve test otomasyonu konuları bulunmaktadır.



Tarkan Tekcan

1970 doğumlu olan Tarkan Tekcan, İzmir Atatürk Lisesi'nden mezun olduktan sonra Dokuz Eylül Üniversitesi'nde elektrik-elektronik mühendisliği eğitimi gördü. Aynı bölümde yüksek lisans yaptı. Tekcan, İzmir Yüksek Teknoloji Enstitüsü'nde mühendislik işletmeciliği alanında da yüksek lisans derecesi aldıktan sonra Dokuz Eylül Üniversitesi'nde doktorasını tamamladı. Tekcan, üniversitede 4 yıl asistanlık ve 2,5 yıl kadar bir kalibrasyon merkezinde yöneticilik yaptı. Askerlik görevinin ardından 1999 yılında Vestel Elektronik'te Ar-Ge test mühendisi olarak çalışmaya başladı. 2006 yılında Vestel Elektronik Ar-Ge Genel Müdür Yardımcılığına atandı. Ekim 2012'den bu yana, Vestel Müşteri Hizmetleri Genel Müdürü olarak çalışmakta. Çalışma konusu olan elektronik ürün güvenilirliği konusunda uluslararası yayınları bulunmakta. Evli ve bir kız, bir erkek çocuk sahibi.



Mustafa Gündüzalp

1979 yılında Orta Doğu Teknik Üniversitesi Elektrik Mühendisliği Bölümü'nde öğrenimini tamamladı. 1982 yılında Boğaziçi Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümü'nden Yüksek Lisans diplomasını aldı. Daha sonra 1988 yılında Dokuz Eylül Üniversitesi'nde doktora çalışmalarını bitirdi. 1983-1987 yılları arasında araştırma asistanı olarak görev yaptı. 1987-1991 yıllarında endüstride araştırma mühendisi olarak çalıştı. 1991'de Dokuz Eylül Üniversitesi, Mühendislik Fakültesi, Elektrik ve Elektronik Mühendisliği Bölümü'nde Yardımcı Doçent olarak görev aldı. Görevini 1995-2001 yılları arasında Doçent olarak sürdürdü. 2001 yılında Profesör oldu. 2001-2010 yılları arasında aynı üniversitede görevine devam etti. 2010 yılında Yaşar Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü'nde Profesör olarak görev almış; halen bu görevini sürdürmektedir.

Devre teorisi, mikrodenetleyici/mikroişlemci tabanlı system tasarımı, sayısal sistemler, kontrol uygulamaları konularında çalışmalarına devam etmektedir.