

Gerçek Zamanlı Gömülü Yazılımlar için Test Otomasyonu: Türkiye Endüstrisinden Bir Yaklaşım ve Deneyim Raporu

Gökhan Urul¹, Vahid Garousi Yusifoğlu^{2,3}, Gökalp Urul¹

1: İntest Teknoloji Yazılım A.Ş.
Ankara, Türkiye
{gokhan.urul, gokalp.urul}@intest.com.tr

2: Sistem ve Yazılım Kalite Mühendisliği Araştırma Grubu (SySoQual)
Yazılım Mühendisliği Bölümü
Atılım Üniversitesi, İncek, Ankara, Türkiye
vahid.garousi@atilim.edu.tr

3: Yazılım Kalite Mühendisliği Araştırma Grubu (SoftQual)
Elektrik ve Bilgisayar Mühendisliği Bölümü
Kalgari Üniversitesi
Kalgari, Alberta eyaleti, Kanada

Öz.

Gömülü sistemler, özel çevre birimleriyle yüksek etkileşim halinde olan genellikle gerçek zaman ihtiyaçlarını karşılayan sistemlerdir. Bu bildiride gömülü sistemler içerisinde yer alan gömülü yazılımların kara-kutu test süreci üzerinde durulacaktır. Gömülü yazılımların test çalışmalarında kullanılacak test otomasyon çözümleri değerlendirilerek, yeni bir yöntem öne sürülmüştür. Önerilen yöntem bir projede uygulanmış ve test çalışmalarına yapılan yatırım verimini ön planda tutan bir endüstriyel deneyim raporu oluşturulmuştur.

Anahtar Kelimeler: Gömülü sistemler, Test otomasyonu, Kara-kutu test

1 Giriş

Gömülü sistemler, gerçek zamanlı gereksinimleri karşılayarak, özel çevre birimleri ile yüksek etkileşimli olarak çalışan sistemlerdir. Bu yazılımlar doğrudan bir donanım üzerinde çalışabilecekleri gibi, gerçek zamanlı bir işletim sistemi üzerinde de çalışabilirler. [1] Tüm gömülü yazılımların ortak özelliği, kendilerine bağlı olan sensörlerini veya haberleşme içinde buldukları çevre birimlerini kullanarak buldukları ortam ile etkileşim içinde bulunmalarıdır.

Test otomasyonu, elle yapılan yazılım testlerinin maliyetini oldukça düşürebilecek yaygın kullanılan bir yaklaşımdır. Ancak, iyi planlanıp, düzgün yürütülmediğinde elle yapılan test yöntemine göre çok daha fazla maliyete neden olabilmektedir. Endüstride

test otomasyon çalışmalarında başarı elde edilebilmesi için otomasyon araçlarının seçimi ve otomasyon yaklaşımı önemli rol oynamaktadır.

Gömülü yazılımlar genellikle çok özel amaçlara yönelik tasarlanmaktadır.(örneğin, araç motor kontrolü, ev gereçleri veya radar sistemleri gibi) Gömülü sistemlerin çok geniş bir spektrumda yer almasıyla ve çok farklı karakteristik özelliklere sahip olmalarıyla birlikte tüm sistemlere yönelik tek bir test otomasyon aracı veya çerçevesi (framework) geliştirmek bir hayli güç olmaktadır. Günümüzde gömülü sistemler üzerine odaklanan birçok test otomasyon aracı geliştirme projesi yürütülmesine rağmen test mühendisleri hala test edilecek gömülü yazılımların tüm testlerini piyasada bulunan ücretsiz veya ticari test otomasyon çözümleriyle tam olarak otomatize edememektedirler. Berner, test otomasyonu alanında edindiği birçok ders ve tecrübe ışığında makalesinde bir sistemin tam otomasyonunun oldukça zor olduğunu belirtmektedir [2].

Bu bildirin devami şu şekilde yapılandırılmıştır. Test otomasyon uygulamalarında karşılaşılan problemin tanımı ve kapsamı 2. Kısım'da sunulmuştur. İlgili çalışmaların bir incelemesine 3. Kısım'da yer verilmiştir. Önerdiğimiz test otomasyon yaklaşımı 4. Kısım'da açıklanmaktadır. 5. Kısım sunduğumuz yaklaşımı gerçekleştirdiğimiz endüstriyel örnek uygulamanın sonuçlarını özetlemektedir. Son olarak, 6. Kısım, sonuçları ve gelecek araştırmalar için önerileri sunmaktadır.

2 Problem Tanımı ve Kapsamı

Günümüzde gömülü yazılım sistemleri çevrelerindeki ortam ile bağlantı kurabilmek için CANBus, UART, RS232, RS422, RS485, LIN, Ethernet, 1553, HDLC, ARINC, PCI, PCIe, DIO, GPIB, SPI, I2C, gibi birçok farklı fiziksel arayüz kullanmaktadır. Aynı fiziksel arayüz üzerinde dahi birçok farklı üst katman haberleşme protokolü (TCP/IP, CORBA, DDS, UDP, Modbus, Profibus, CANopen, FlexRay, XML-RPC, WCF, vb.) kullanılabilir.

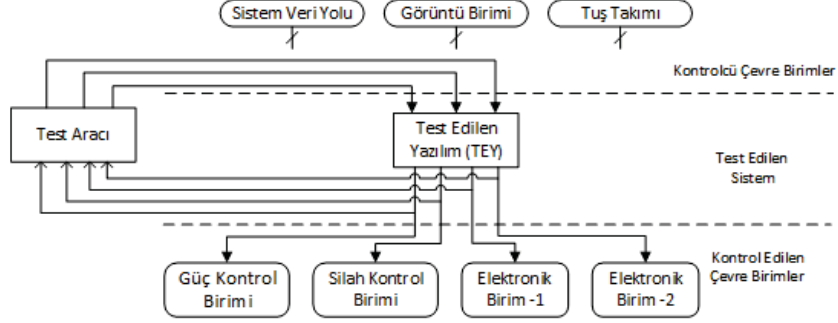
Çalışmamız kapsamında yer alan, Test Edilen Yazılım(TEY) savunma sanayinde yer alan gömülü bir yazılımdır. Bu çalışmada gizlilik sebebiyle TEY hakkında fazla detay verilmeyecek, uygulanan test yaklaşımı üzerinde durulacaktır. TEY birçok farklı elektronik birimin merkezinde yer alan gerçek zamanlı gömülü bir yazılımdır. TEY çevresindeki radar, silah kontrol birimi, güç birimi gibi elektronik kontrol birimleriyle haberleşerek, bir silah sisteminin ana kontrol işlevlerini yönetmektedir. TEY geliştirme ve bütünleşme çalışmaları sırasında birçok test aşamasından geçmektedir. Ancak, TEY'in en geniş kapsamlı kara-kutu testleri diğer çevre birimleri ile bütünleşmiş edildikten sonra yapılabilir. Kara-kutu testleri sırasında, sistemde bulunan birimler ile TEY arasında gerçekleşen trafik ve oluşan veri alışverişi, TEY'in gereksinimlerinin doğrulanmasında kullanılabilir. TEY birçok farklı haberleşme arayüzüne sahip olabilmektedir. TEY'e ait tüm arayüzler üzerinde gerçekleşen trafiğin senkronize bir şekilde aynı test otomasyon altyapısı üzerinden kontrolü bir problem olarak karşımıza çıkmaktadır. Bu çalışmadaki test yaklaşımı TEY ve TEY'in çevre birimleri ile olan arayüzleri arasındaki uyumluluğu ve doğrulanabilirliği üzerine odaklanmaktadır.

Kara-kutu test aktiviteleri temel olarak sisteme uygulanan girdi ve gözlenen çıktılara dayanmaktadır. Bu çalışmada kullanılan test otomasyon altyapısı bünyesinde girdi ve çıktıları “alınan mesajlar” ve “gönderilen mesajlar” olarak tanımlamaktayız. Alınan ve gönderilen mesajlar gösterimi “Message Sequence Charts” [3] dilinde bulunan alınan ve gönderilen olaylar ile benzer anlamda kullanılmaktadır.

Çalışmamızda belirtilen TEY, çevre birimleri ve sensörleri ile haberleşme içerisindedir. Bu haberleşme “Alınan Mesajlar” ve “Gönderilen Mesajlar” olarak modellenebilmektedir. Kara-kutu testleri TEY ve çevre birimleri arasındaki fiziksel arayüzler üzerinde gerçekleşen, alınan ve gönderilen mesajlara dayandığından, test aktivitelerinde haberleşme büyük öneme sahiptir. Testlerde otomasyon kullanımı TEY’in etrafındaki çevre birimlerin simüle edilmesini gerektirmektedir. İdeal bir test ortamında, tüm çevre birimlerin simüle edilmesi tamamiyle otomatik çalışan testlerin geliştirilmesini sağlayabilir. Ancak, projelerde test çalışmalarına ayrılan bütçeler ve test otomasyon yatırımının geri dönüşü ele alındığında nadiren tamamiyle simüle edilmiş bir test ortamı tasarlanabilmektedir. Tamamiyle otomatize edilmeyen test ortamları da bir takım otomasyon işlevlerini barındırabilmektedir. Bütçe ve yatırımın geri dönüşü kısıtlarına göre karar verilerek TEY’in bazı temel arayüzleri simüle edilirken, diğer arayüzler için gerçek birimler kullanılabilir. Simüle edilecek olan bu temel arayüzler, özellikle TEY’e komut gönderen arayüzler içerisinden seçilmelidir. Böylece yarı-otomatik bir test ortamı oluşturulabilir.

Çalışmamızdaki test edilen gömülü sistemde, sistemdeki çevre birimler TEY tarafından kontrol edilmektedir. Gömülü sistemlerde uygulanacak test otomasyonunun başarısı için, sistemin test edilebilirliği olduğu kadar kontrol edilebilirliği ve gözlemlenebilirliği de yüksek seviyede tutulmalıdır [4]. Örnek olarak, yüksek seviyede kontrol edilebilirlik ve gözlemlenebilirlik için, test edilen yazılım ve çevre birimleri arasındaki iletişim altyapısının esnek ve test edilebilir bir yapıda tasarlanması gerekmektedir. Bu çalışma kapsamında kullanılan test otomasyon aracı temel 2 yeteneğe sahiptir. Birincisi sistemde bulunan çevre birimlerin yerine geçerek, bu birimleri simüle edebilmektedir. Diğer önemli özelliği ise gerçek çevre birimler ve TEY arasında bulunan haberleşme ağını dinleyebilmektedir. Bu araç hem izleyici(sniffer), hem de bir simülatör olarak davranabilmektedir.

TEY, sistem içerisinde merkezi bir noktada yer almaktadır. Kontrolcü çevre birimler temel olarak TEY’e komut göndermektedirler. TEY’i kontrol eden arayüzler daha çok kontrolcü çevre birimler tarafında bulunmaktadır. Bir diğer çevre birim türü kontrol edilen çevre birimlerdir. TEY kontrol edilen çevre birimlere komut göndererek, onları kontrol etmektedir.



Şekil 1: TEY Mimarisi ve Test Altyapısı

Tüm çevre birimleri simüle etmenin maliyeti yüksek olacağından, kontrol edilen çevre birim ve TEY arasındaki arayüzlerde izleme yöntemi kullanılmaktadır. İzleme yöntemi kullanılırken hata enjekte etme yöntemi ile birimlerin hatalı durumları da oluşturulabilmektedir.

İzleme yönteminin dezavantajı doğru çalışan bir sistemdeki hatalı senaryoların gerçekleştirilememesi olmaktadır. Yalnızca izleme yöntemi kullanılan testlerde sistemin hatalı senaryolarını kapsayan gereksinimler test edilemeyebilmektedir. Maliyet açısından bir test çalışmasının kapsamı ele alındığında, izleme yöntemi ve simülatör geliştirme yöntemi arasına bir adım daha eklenmesi gerektiği ortaya çıkmaktadır. Hata enjeksiyonu yöntemi, yüksek maliyet gerektiren simülatör geliştirme çalışmasını ortadan kaldırırken, pasif bir işlev olan izleme yöntemine stimülasyon yeteneği kazandırarak maliyet etkin şekilde test kapsamını artırabilmektedir.

Bir yazılım projesi kapsamında test aşaması 4 ana bölüme ayrılabilir [5]. (1) test tanım tasarımı, (2) test script hazırlanması, (3) testlerin koşturulması, (4) testlerin değerlendirilmesi. Çalışmamız kapsamında değineceğimiz çözümler testlerin koşturulması aşamasında yer almaktadır.

3 İlgili Çalışmalar

Gömülü sistemlerin kara-kutu testlerini gerçekleştirmek amacıyla kullanılacak birçok çerçeve, araç ve çalışma bulunmaktadır. Bunlar içerisinde öne çıkanlar; TTworkbench [6], Wireshark [7], CANoe [8], ve MESSINA [9] bu bölümde incelenecektir.

TTworkbench [6], test otomasyon projeleri için kullanılan Eclipse tabanlı bir test geliştirme ve koşturma çevre yazılımıdır (IDE). TTworkbench, eklenti (plugin) temelli bir çerçeve sunarak yazılım ürünlerinin ve servislerinin testlerini kullanılan teknoloji-den, işletim sisteminden ve uygulandığı ortamdan bağımsız hale getirmektedir. Avrupa Telekomünikasyon Standartları Enstitüsü'nün (ETSI) bir standardı olan Testing and Test Control Notation 3 (TTCN-3) [10] dili kullanılarak TTworkbench aracı üzerinde test tanımları tasarlanabilmektedir. TTworkbench, haberleşme sıralaması, gönderme,

alma, bekleme gibi test otomasyonu için önemli fonksiyonları barındırmasına rağmen gerçek zamanlı haberleşme kanalı izlemeye(sniffer) olarak vermemektedir. Test alanında oldukça özelleşmiş olan TTCN-3 dili, ancak bu konuda uzman test mühendislerinin kullanabileceği bir ortam sağlamaktadır.

Temel olarak, Wireshark [7] bir ağ izleme(sniffer) aracıdır. Wireshark aracının temel amacı test faaliyetleri olmasa da, ağ arayüzlerini görünür kılarak, ağ üzerindeki trafiği kullanıcıya sunmaktadır. Doğrulama amaçlı yapılan analiz yönteminin gerçekleştirilebileceği bir düzlem sunmaktadır. Bir ağ üzerindeki görünürlük doğrulanabilir bir ağ yapısını beraberinde getirmektedir. Wireshark yalnızca Ethernet veri yolu üzerinde çalışmasına rağmen, farklı fiziksel arayüzler için ilham vermektedir. Analiz ve izleme yönteminin kara-kutu testlerde önemli bir yaklaşım olduğu düşünüldüğünde Wireshark aracının kullandığı izleme(sniffer) metodunun bir test otomasyon altyapısında bulunmasının çok faydalı olacağı değerlendirilmektedir.

CANoe [8] elektronik kontrol birimi(EKB) geliştirilmesinde, testlerinde ve analizlerinde kullanılan bir yazılım aracıdır. Otomotiv endüstrisinde ve özellikle CAN(Controller Area Network) veri yolu üzerinde uzmanlaşan CANoe yazılımı, veri yolu simülatörü, test çevresi, ve tam ekipmanlı tanımlayıcı(diagnostic) yeteneklerine sahiptir. [11]'de belirtilen çalışmada test ortamındaki tüm birimler simüle edilerek CANoe aracıyla testler gerçekleştirilmiştir. CANoe aracının fonksiyonel yetenekleri test otomasyon çalışmaları için oldukça uygun olsa da, çalışmamızın amacı olan farklı fiziksel arayüzler için bir çözüm sunmamaktadır.

Donanım ve yazılım katmanlarından bağımsız otomatik test tanımları hazırlanmasına ve koşturulmasına olanak veren bir diğer test otomasyon aracı da MESSINA [9]'dir.

Palmieri [12]'nin tez çalışmasında özellikle otomotiv alanında günümüzde yer alan test teknikleri ve metodlarına yer verilmiştir. Bu çalışma içerisinde 28 adet açık kaynak kodlu ve ticari test aracının karşılaştırması yapılarak, sistem testlerine olan katkıları değerlendirilmiştir. Palmieri, TTCN-3 test dili üzerine odaklanarak CAN veri yolu üzerindeki deneyimlerini bu çalışmada sunmuştur.

4 Test Otomasyon Yaklaşımı

Çalışma kapsamında sistemde yer alan arayüzler üzerinde izleme(sniff), kaydetme, oynatma yapabilen eklenti tabanlı bir otomasyon altyapısı kullanılarak gerçekleştirilen bir test metodu öneriyoruz. Çevre birimlerin her birinin yerine test amaçlı bir simülasyon yazılımı gerektirmeden, yalnızca ilgili TEY-çevre birim arasındaki haberleşme protokolünün geliştirilerek test otomasyon altyapısına entegre edilmesi test otomasyonuna katkıları büyük olmaktadır. Böylece test altyapısı TEY'i kontrol altına alırken, TEY ve kontrol edilen çevre birimler arasındaki haberleşme de test altyapısı tarafından dinlenebilecektir.

Gömülü TEY arasındaki çevre birimler iki kategoriye ayrılmaktadır: (1) kontrolcü birimler ve (2) kontrol edilen birimler. En üst seviye otomasyon amacına ulaşmak için kontrolcü birimlerin tam olarak simüle edilmesi gerekirken, özellikle karmaşık yapıdaki kontrol edilen birimler için simülasyona gerek olmayabilmektedir. Kontrol edilen

karmaşık birimler ve TEY arasındaki haberleşme dinlendiğinde, test otomasyonunun uygulanabildiği testler önemli derecede artmaktadır.

Test altyapısını 3 başlık altında ele alacağız: (1) mevcut test araçlarının yeniden kullanımını ya da adaptasyonu, (2) test stratejisi, (3) test aracı geliştirme.

4.1 Mevcut Test Araçlarının Yeniden Kullanımının ve Adaptasyonunun Değerlendirilmesi

Çalışma kapsamında ticari ya da ücretsiz olarak gömülü sistemlerin testlerinde kullanılacak olan test araçlarının alanda kullanımına ve adaptasyonuna dair sistematik bir çalışma yürütülmüştür.

Sistematik değerlendirmemizin sonucunda, yalnızca Ttworkbench [5] ve yeteneklerinin, ihtiyaçlarımızı ve kapsamı karşılayabileceği değerlendirilmiştir. Ttworkbench'in kullanıldığı küçük demo bir proje sonunda yapılan fizibilite analizi sonucunda, bu araca izleme(sniffer) yeteneğinin eklenmesinin mümkün olmadığı ve şirket kapsamında bir test otomasyon aracı geliştirilmesi değerlendirilmiştir. Ttworkbench'in gerektirdiği TTCN-3 dilinin öğrenilmesi şirket içerisinde bu aracı kullanacak olan personel sayısını da önemli ölçüde azaltacağı öngörülmüştür. Yalnızca TTCN-3 dilini öğrenen test mühendislerinin değil, projenin çeşitli seviyelerinde yer alan yazılım, donanım, test ve sistem mühendislerinin de test aracına hâkimiyeti hedeflenmiştir.

4.2 Test Stratejisi

Kurum içinde geliştirilen test altyapısının yetenekleri göz önüne alındığında, test edilecek sistemde bulunan çevre birim arayüzlerinden hangilerinin test aracı içerisinde yer alması gerektiği ve hangi seviyede simülasyon gereksinimi olduğu değerlendirilmelidir. Yöntemimizde sistemdeki bazı arayüzler tamamen simüle edilirken, diğerleri üzerinde yalnızca dinleme(sniffing) yapılmaktadır. Dinleme yapılan arayüzler üzerindeki trafik kullanıcı ve test aracı tarafından görünür kılarak doğrulanabilir bir seviyeye getirilmektedir. Bu adımda, haberleşme arayüzleri karmaşıklıklarına ve kontrol seviyelerine göre kategorize edilecektir. İlk kategoride TEY ve kontrol edilen birimler arasındaki arayüzler yer almaktadır. Bu kategorideki arayüzlerin tam olarak simüle edilmesi yüksek maliyetlere neden olabileğinden bu arayüzleri izlemek düşük maliyetli bir otomasyon yeteneği sağlamaktadır. Entegrasyon testleri sırasında bu kategorideki çevre birimlerin simülasyonları değil kendileri kullanılmaktadır. Test altyapısının paralel kullanıma uygunluğundan dolayı tam olarak simüle edilen arayüzlerden gönderilen komutların çıktısı olan bilgiler bu arayüzlerin izlenmesi ile doğrulanabilmektedir. Diğer kategoride TEY ve kontrolcü birim arasındaki arayüz yer almaktadır. TEY'e komut gönderen arayüzlerin büyük ölçüde simüle edilerek test altyapısına bağlanması otomasyon oranını oldukça artırmaktadır.

4.3 Test Ortamının Geliştirilmesi

Firma içerisinde eklenti temelli tasarlanan test aracı, yeni haberleşme protokollerine özel yeni eklentilerin kolayca tasarlanmasını sağlamaktadır. Çalışılan alanda, farklı

projelerde aynı haberleşme protokolleri kullanılmaktadır. Bu protokollere özel eklentiler farklı projelerde ortak kullanılırken, yeni karşılaşılan protokoller için yeni eklenti geliştirilmesi yapılmaktadır. Eklenti geliştirmeyi yazılım geliştirme alanında uzman mühendislerin yapması daha uygun olmaktadır.

Ayrıca firmada geliştirilen test aracı yazılım ve donanım soyutlamasını sağlayan eklenti temelli katmanlı bir yazılım mimarisine sahiptir. Gizlilik ve projelerin hassasiyetinden dolayı, kullanılan test aracı ile ilgili detaylı bilgi bu makale kapsamında paylaşılamamaktadır.

5 Yaklaşımın Değerlendirilmesi

Bir sonraki bölümde projemizde uygulanan test yaklaşımının ve aracının değerlendirilmesi yapılacaktır.

5.1 Test Tanımlarının Tasarımı

Gömülü bir sistemde test otomasyonu oldukça zor bir problemdir. Sistemi ve davranışını tanımlayan mesaj akış tabloları tasarlanmış ve düzgün dokümente edilmiş olsa da, sistem üzerinde gerçekleşen haberleşme tutarlı olmayabilir. TEY'in paralel yürüttüğü görevlerden(multithreaded) kaynaklı oluşan gerçek haberleşme, sistemde herhangi bir hata olmamasına rağmen tasarlandığı gibi gerçekleşmeyebilir. Bu noktada keşif amaçlı(exploratory) [13] test yöntemi önemli bir rol oynamaktadır. Test tanımları mesaj akış diyagramı şeklinde oluşturulmaktadır.

Bazı test tanımları tam otomatik bir şekilde tasarlanabilmektedir. Temel olarak, TEY'e bir girdi yaparak TEY'in çıktılarının kontrolü yöntemi test aracının ilgili arayüzleri tarafından kapsanıyorsa bu test tanımları tam otomasyon kapsamına alınabilir.

Diğer bir metot ise yarı otomatik test yöntemidir. Yarı-otomatik test elde etmek için test aracı TEY'e bir komut gönderir ve TEY'in çıktıları anlık ve gerçek zamanlı olarak yine test aracı tarafından kayıt edilir. Test aracı sistemdeki tüm arayüzleri senkronize bir şekilde kaydettiğinden, kaydedilen komutlar ve cevaplar test sorumlusu tarafından daha sonra doğrulanabilir bir şekilde ulaşılmıştır.

Bir diğer yarı-otomatik test metodu, otomatik testler ile manuel yöntemlerin birleştirilmesi ile meydana gelmektedir. Test aracında hazırlanan bir senaryoya, mesaj kutusu eklenerek operatörün bir işlem yapması ya da bir kontrol gerçekleştirilmesi istenebilir. Operatör müdahalesi sonrasında test adımları otomatik olarak devam edebilir.

5.2 Kazançlar

Bu çalışmanın amacı gömülü bir yazılımın elle yapılan yöntem ya da otomasyon yöntemi ile test edilmesi arasındaki dengeyi sağlayarak test otomasyonuna yapılan yatırımların verimini artırmaktır. Test otomasyonunun bu alanda iki büyük avantajı olduğu görülmektedir. Bunlar; (1) teste ayrılan iş gücünün azalması, (2) testlerin daha kaliteli bir şekilde gerçekleştirilebilmesi.

Teste ayrılan iş gücünün azalması, test altyapısını kurma ve testi gerçekleştirme aşamalarındaki kazanımlardan elde edilmektedir. Sıfırdan test araçları geliştirmek yerine test altyapısı kullanılarak yeni eklentiler geliştirmenin önemli ölçüde iş gücünü azalttığı gözlenmiştir.

Gömülü sistemlerde test otomasyonu TEY'in sadece fonksiyonel değil fonksiyonel olmayan testlerine, dolayısıyla kalitesine de katkı sağlamaktadır. Tekrarlanan test senaryoları sistemde büyük problemlere yol açabilmektedir. Bazı testleri yüksek miktarlarda tekrarladığımızda fonksiyonel olmayan bir takım çok önemli ve yıkıcı olarak kategorize edilen hatalarla karşılaştık. Otomasyon yaklaşımı olmadan bu hatalarla karşılaşmak pek mümkün gözükmemektedir.

Firma bünyesinde geliştirilen test otomasyon altyapısı, bir izleme aracı olarak da kullanılabilirdiğinden, sistemdeki tüm veri yolu trafiğine erişim sağlamaktadır. Trafik içerisinde gerçekleşen verilerin, zaman etiketi eklenerek eşzamanlı bir şekilde kaydedilmesiyle birlikte test mühendisleri doğrulama yöntemlerinden analiz yöntemini aktif bir şekilde kullanabilmişlerdir.

Elle, yarı-otomatik ve otomatik test metodlarını bir arada kullanarak tasarladığımız test sisteminde, toplam test tanımlarının %30'unda tam otomasyon sağlanabilmiştir. Ancak test altyapısı tüm test tanımlarında kullanılmıştır.

5.3 Öğrenilen Dersler

Bu çalışmada olduğu gibi karmaşık bir gömülü sistemde tüm gereksinimlerin testini otomatize etmek maliyeti çok artırmaktadır. Yatırımın verimini en uygun noktaya çekebilmek için, test tanımları duruma göre yarı-otomatik, otomatik ve manuel olarak tasarlanmıştır.

Test otomasyon çalışmalarında yatırımın verimine etki eden 2 önemli faktörün kullanılan test otomasyon aracının ve aracı kullanan test mühendisinin nitelikleri olduğu gözlenmiştir. İyi tasarlanmış bir test otomasyon altyapısının harcanan toplam iş gücünü azaltarak, testlerin kalitesinde önemli bir artışı sağlayabileceği gözlenmiştir. Çoğu durumda, otomasyonun yatırım verimini olumlu bir şekilde etkilediği fakat TEY'in hangi gereksinimleri için testlerin otomatize edilmesi konusunda iyi tasarlanmış bir metodun olmadığı görülmüştür.

5.4 Farklı Endüstrilere Uyarlama

Önerilen metodu savunma sanayinde bulunan bir gömülü sisteme uygulamamıza rağmen bu yaklaşım farklı endüstrilerde yer alan problemlere de uyarlanabilecektir. Şekil 1'de gösterilen sistem tasarımı otomotiv ve endüstriyel otomasyon alanlarında yer alan sistemlerle oldukça benzerdir. Bu metod diğer gömülü sistemlerin testlerine uyarlanabilir.

5.5 Sonuç ve Gelecek Çalışmalar

Bu çalışmada, önerilen test yöntemi, firmada geliştirilen gömülü yazılımların test çalışmalarına sistematik bir yaklaşım getirmiştir. Geliştirilen test aracı firma içerisinde

aktif olarak kullanılmaktadır. Karmaşık gömülü sistemlerin testleri kolay bir şekilde yapılamadığından oldukça maliyetli bir test çalışması oluşabilmektedir. Test çalışmalarına ayrılan yatırımın verimi göz önüne alındığında düzgün planlanmış bir test altyapısı ile elle, yarı-otomatik ve tam-otomatik bir şekilde test tanım tasarımları yapılabilir. Sunduğumuz temel fikir olan, kontrol altındaki birimlerin arayüzlerinin dinlenmesi ve kontrolcü birimlerin arayüzlerinin simüle/stimüle edilmesi, testleri semi-otomatik bir şekilde getirerek, maliyetleri düşürmektedir. Bu sayede, zamandan tasarruf, test kalitesinde artış ve yatırımın veriminde kazanç sağlanabilmektedir. Proje kapsamında oldukça başarılı bir test sürecinden geçilerek, birçok kritik hata bulunmuştur. Farklı projelerde yer alan ekip üyeleri, yapılan prototip çalışmayı görerek, uygulanan test stratejisini kendi projelerinde de uygulamak istemişlerdir.

Bu çalışmada, bir sistemin testlerinde kullanılacak otomasyon oranını etkileyen iki önemli faktör olduğu tecrübe edilmiştir. Faktörlerden biri testlerde kullanılan test aracının niteliği ve kalitesi, diğeri ise test aracını kullanan test mühendisinin alana ve test aracına olan yatkınlığı olmuştur.

Gelecek dönemde gerçekleştirilecek çalışmaların başında bu metodu farklı projelerde uygulamak gelmektedir. Farklı projelerde uygulanırken aşamalarla ilgili metriklerin sistematik bir şekilde alınması test otomasyon çalışmalarında gerçekleşen yatırım veriminin daha iyi hesaplanabilmesinde kullanılabilir.

Kaynakça

- [1] E. Kahraman ve V. Ünal, "Gerçek Zamanlı Gömülü Sistem ve Yazılım Tasarımı'nda ASELSAN Yaklaşımı." III. Ulusal Yazılım Mühendisliği Sempozyumu UYMS 7: 27-30.
- [2] B. Stefan, R. Weber ve R. K. Keller, "Observations and lessons learned from automated testing." Proceedings of the 27th international conference on Software engineering. ACM, 2005.
- [3] D. Harel ve P. S.Thiagarajan, "Message sequence charts." UML for Real. Springer US, 2003. pp. 77-105.
- [4] L. Zhao, "A new approach for software testability analysis." Proceedings of the 28th international conference on Software engineering. ACM, 2006.
- [5] Y. Amannejad, V. Garousi, R. Irving, ve Z. Sahaf, "A Search-based Approach for Cost-Effective Software Test Automation Decision Support and an Industrial Case Study." et al. s.l.: co-located with the Sixth IEEE International Conference on Software Testing, Verification, and Validation (ICST), , April 2014, Vols. pp. 302-311.
- [6] Ttworkbench, <http://www.testingtech.com/products/ttworkbench.php>, Last accessed: May 2014.
- [7] Wireshark, www.wireshark.org, Last accessed: May 2014.
- [8] CANoe, http://vector.com/vi_canoe_en.html, Last accessed: May 2014.
- [9] MESSINA, <http://www.berner-mattner.com/en/berner-mattner-home/products/messina/>, Last accessed: May 2014.
- [10] Testing and Test Control Notation version 3(TTCN-3), <http://www.ttcn-3.org/>, Last accessed: May 2014.

- [11] F. Zhou, L. Shuqin, ve H. Xia, "Development method of simulation and test system for vehicle body CAN bus based on CANoe." Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on. IEEE, 2008.
- [12] M. Palmieri, "System Testing in a Simulated Environment." (2013).
- [13] J. Bach, "Exploratory testing explained." Online: <http://www.satisfice.com/articles/et-article.pdf>, Last accessed: May 2014.