

Büyük Ölçekli bir Gömülü Yazılımın Geliştirme ve Otomatik Test Deneyimi

Eda Gürler¹ Murat Yılmaz²

^{1,2} Radar Elektronik Harp ve İstihbarat Sistemleri (REHİS) Grubu, ASELSAN A.Ş. Ankara
¹tverdi@aselsan.com.tr , ²muratyilmaz@aselsan.com.tr

Özetçe. ASELSAN REHİS grubu bünyesinde geliştirilmekte olan büyük ölçekli ve uzun süreli bir elektronik harp projesinde, bir gömülü kontrol yazılımının çevik yöntemlerle geliştirilme süreci ve bu süreçte kullanılan otomatik test kurgusu bu bildirinin konusudur. Bildiride gömülü kontrol yazılımlarında geliştirme ve test aşamalarında yaşanan sıkıntılar, bu sıkıntıların çevik yöntemler ve otomatik test kurguları ile giderilmesi ve bu çalışma ile elde edilen deneyim paylaşılmaktadır.

Anahtar Kelimeler: Gömülü yazılım otomatik test çevik yöntemler
Fitnesse

1 Giriş

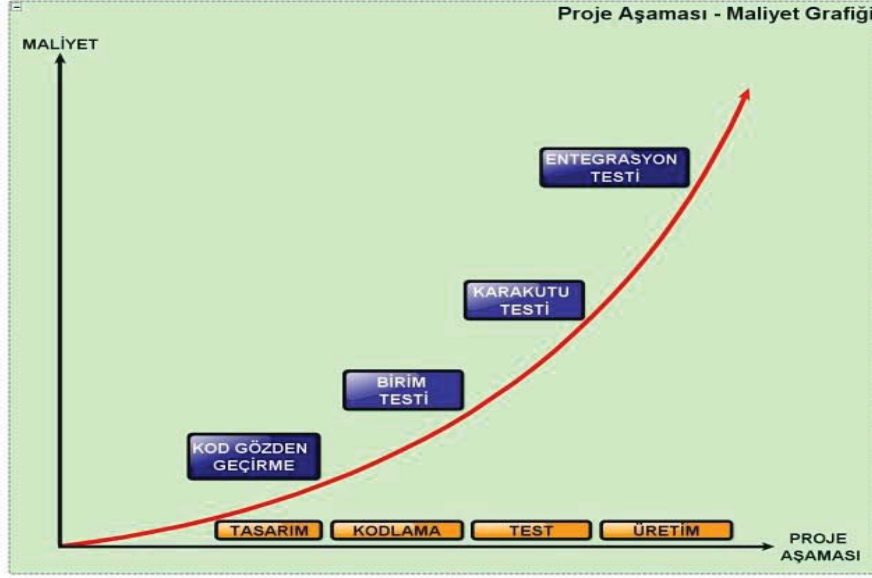
Çevik yöntemler ile yazılım geliştirme, ASELSAN REHİS Grubu Görev Yazılımları Müdürlüğü (GYM) bünyesinde 10 yılı aşkın süredir uygulanmaktadır [1].

Çevik yöntem pratiklerinin kullanılması ile test edilmiş çalışan kodlar projenin ilk koşullardan itibaren oluşturulmaktadır. Her koşulda yeni yeteneklerin kodlanması ile geliştirilmekte olan çalışır durumdaki yazılım daha olgun hale gelmektedir. Yazılımın yeteneklerinin ve karmaşıklığının artması ile yeni eklenecek olan yeteneklerin mevcut yazılımda hata oluşmasına yol açma ihtimali de artmaktadır.

Yazılımların kaliteli ve hatasız şekilde geliştirilmeleri için geliştirme safhası boyunca test edilmeleri faydalı bir pratiktir. Geliştirme safhası boyunca test edilen yazılımların içerdiği hataların daha erken aşamada yakalanabildiği görülmektedir. Hataların daha erken aşamada yakalanması hem çözüm maliyeti, hem yazılım kalitesi açısından olumludur.

Şekil 1'de hataların yakalanma aşaması ile çözüm maliyeti arasındaki ilişki görülebilir.

Bu bildiride, büyük ölçekli bir gömülü yazılımın çevik yöntemler [2] ile geliştirilmesi ve otomatik test kurgusu ile doğrulanması ele alınmaktadır.



Şekil 1. Proje Aşaması – Hata Maliyeti Grafiği [3]

2 Gerçek Zamanlı Gömülü Yazılım Geliştirme

Gömülü yazılımların geliştirilme ve test süreçleri, kişisel bilgisayar ortamında çalışan yazılımların geliştirilme ve test süreçlerinden bazı yönlerden farklılık göstermektedir. Gömülü yazılımların geliştirildikleri ortam ile çalıştıkları ortam aynı değildir. Kodlar kişisel bilgisayar ortamında geliştirilip derlenir ancak geliştirildikleri ortamda çalışamazlar. Bu kodlar özel donanım kartları üzerine yüklenerek bu kartlar üzerindeki işlemcilerde çalışırlar. Bu durum test ve hatadan arındırma süreçlerinde çeşitli zorluklar ortaya çıkarır:

- Donanım kartının açılması ve test edilecek yazılımın kart üzerine yüklenerek çalıştırılması, her testte bu işlemin tekrarlanması zaman almaktadır.
- Donanım kartlarının maliyetli olması sebebiyle yazılım geliştirme mühendisleri ve test mühendisleri tarafından paylaşılarak kullanılması sıkıntı yaratmaktadır.
- Donanım kartları yüksek soğutma ihtiyacı sebebiyle geliştirme ortamından ayrı bir odada bulunmaktadır. Fiziksel olarak bu donanımlara erişim kısıtlıdır. Donanımlara herhangi bir müdahale yapılması gerektiğinde (gücün kesilip yeniden verilmesi, seri kanal bilgilerinin izlenmesi gibi) erişim sıkıntısı yaşanmaktadır.

Elektronik harp projelerinde kullanılan gömülü kontrol yazılımları farklı dillerde geliştirilmiş çeşitli yazılımlarla ve farklı veri yolları üzerinden çeşitli donanımlarla haberleşmek mecburiyetindedir. Bu sebeple hem kod geliştiren yazılımcının kendi

testlerini yapması hem de yazılımın doğrulama testlerinin yapılması sırasında sıkıntılar yaşanmaktadır.

Gerçek zamanlı gömülü kontrol yazılımları, tanımlı zaman limitleri aşılmadan belli görevleri yerine getirmek durumundadır. Ortamdaki veri miktarının yoğunluğu, işlenmesi gereken verinin karmaşıklığı gibi sebepler ile ilgili görevlerin zamanlarının aksatılmaması gerekir. Bu gereksinim gerçek zamanlı gömülü kontrol yazılımlarının değişken ortamlarda, farklı tip ve miktarda veri ile test edilmesinin ne denli önemli olduğunu ortaya koymaktadır.

Elektronik harp projelerinin görevlerinde başarıya ulaşması için hatasız, gereklerini eksiksiz ve uygun zaman diliminde yerine getirebilen gömülü kontrol yazılımlarının kullanılması gereklidir. Elbette ki bu yazılımların kaliteli ve hatasız geliştirilebilmeleri için test yöntemleri büyük önem taşımaktadır.

3 Geliştirme ile Eşzamanlı Test

Bu bildiride anlatılmakta olan elektronik harp projesinin yazılımlarının geliştirilmesi sırasında çevik yöntem pratikleri kullanılmıştır. Kullanılan pratikler eXtreme Programming (XP) ve SCRUM pratiklerinin bir karmasıdır [4][6][7]. Projenin yazılım gereksinimleri hikayelere dönüştürülmüş, hikayeler arasında önceliklendirme yapılmış ve her koşu için önceliğine uygun olarak gerçekleştirilmiştir. Bu kapsamda yazılım geliştirme mühendisleri tarafından:

- Yeni yetenekler kodlanmış
- Birim testleri gerçekleştirilmiştir.

Gömülü kontrol yazılımının geliştirilmesi sırasında, yazılım mühendisleri tarafından gerçekleştirilen her hikaye için ilgili test hikayesi de test mühendisleri tarafından gerçekleştirilmiştir. Bu kapsamda test mühendisleri tarafından:

- İlgili simülatör(ler)e gerekli güncellemeler yapılmış
- Otomatik test aracında test senaryoları tanımlanmış
- Test aracı ile simülatörler arasındaki bağlantı (fixture) kodları yazılmıştır.

Her koşulun sonunda yazılım geliştirme ve yazılım test ekibi bir araya gelip ilgili koşulda gerçekleştirilen yeteneklerin testlerini otomatik test aracı altyapısı ile yaparak sonuçlarını değerlendirmiştir. Bu kapsamda:

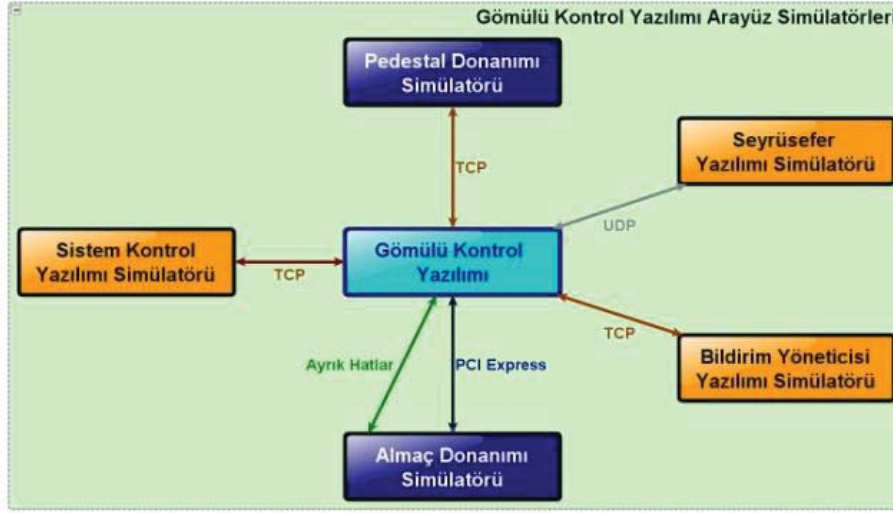
- Ortaya çıkan hataların kaynağı tespit edilmiş
- Gömülü kontrol yazılımında ve simülatör kodlarında gerekli düzeltmeler yapılmış
- Gerekliyse otomatik test aracındaki testler güncellenmiş
- Testler tekrarlanarak hatasız yazılım/simülatör/test grubu elde edilmiştir.

Bu sayede, gömülü kontrol yazılımı yeni yetenekleri içerecek şekilde gelişirken, test ortamı da eşzamanlı olarak güncellenmiştir. Var olan testlere ek olarak yeni testler yazılmış, yazılıma yeni gereksinimler eklendikçe yazılım tüm testleri kapsayacak şekilde teste tabi tutulmuştur.

4 Simülâtör Altyapısı

Geliştirilen gömülü kontrol yazılımı, iki farklı donanım ve üç farklı yazılım arayüzüne sahiptir. Bu arayüzlerin tamamı için simülâtörler geliştirilmiştir. Otomatik test altyapısı, bu simülâtörleri kullanarak gömülü kontrol yazılımına belli senaryolar dahilinde mesaj ve/veya veri gönderebilir. Ayrıca bu simülâtörler, gömülü kontrol yazılımından gelen mesaj ve/veya veriyi test sonuçlarında kullanılmak üzere otomatik test altyapısına göndermektedir. Gömülü kontrol yazılımı, herhangi bir başka yazılıma veya çalıştığı donanım kartı dışında herhangi bir donanıma ihtiyaç duymadan test edilebilmektedir.

Şekil 2’de geliştirilen gömülü kontrol yazılımı ve haberleştiği simülâtör arayüzleri görülebilir.

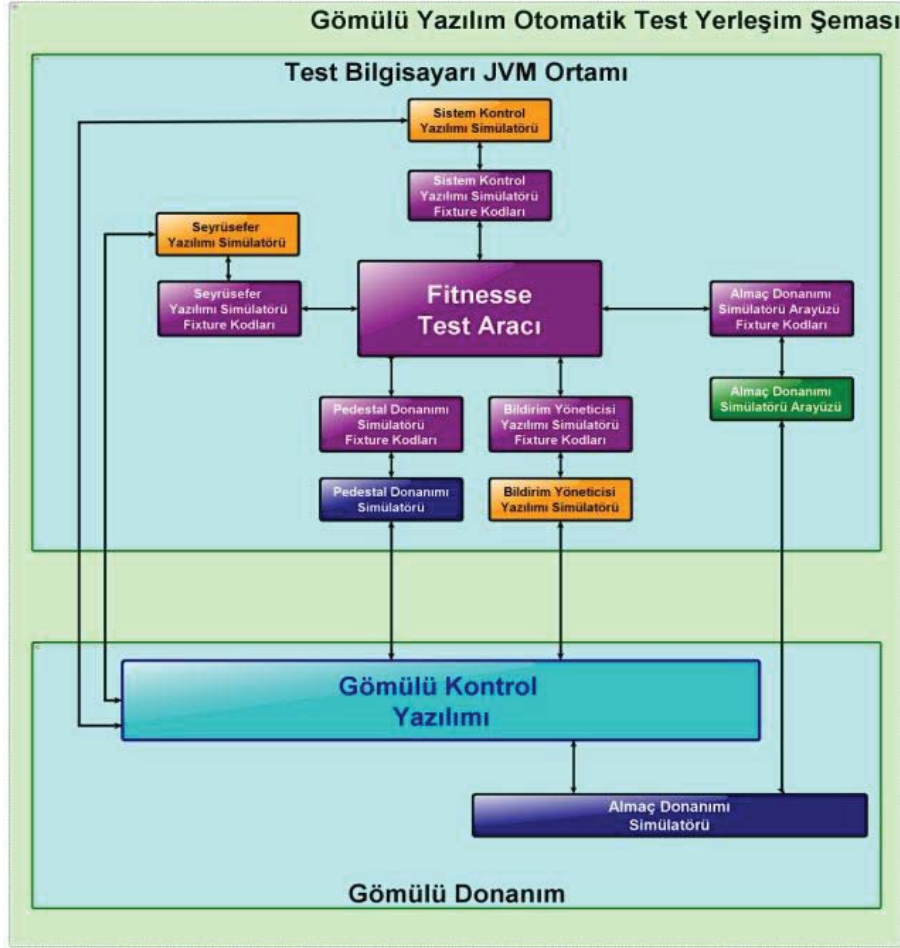


Şekil 2. Gömülü Kontrol Yazılımı Arayüz Simülâtörleri

5 Fitness Testleri

Fitness [5] açık kaynak kodlu, Wiki tabanlı bir entegrasyon test aracıdır. FIT (Framework for Integrated Testing) altyapısını kullanır.

Fitness altyapısının az sayıda gereği olan bir gömülü yazılımın testlerinde kullanılması daha önce ASELSAN REHİS grubunda denenmiştir [4]. Ancak büyük ölçekli bir askeri projede ve gereksinim sayısı 200’den fazla olan bir gerçek zamanlı gömülü kontrol yazılımında, simülâtörler aracılığı ile Fitness test altyapısının kullanılması ASELSAN REHİS grubunda ilk kez gerçekleştirilmiştir. Kullanılan simülâtörler ve test altyapısı Şekil 3’te görülebilir.

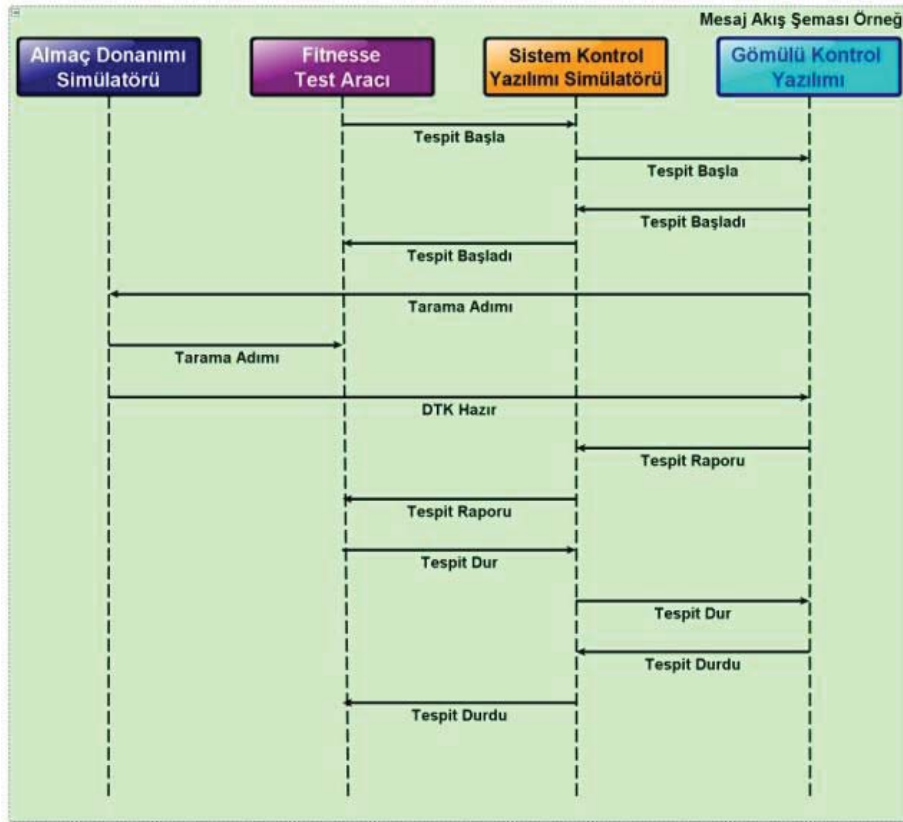


Şekil 3. Gömülü Yazılım Otomatik Test Yerleşim Şeması

6 Otomatik Test Kurgusu

Fitnessse testleri, bir senaryo kapsamında test edilecek gömülü kontrol yazılımına belli mesaj ve verinin simülatörler üzerinden gönderilmesini tetikler. Gömülü kontrol yazılımı tarafından bu tetiklenme üzerine üretilen mesajlar yine simülatörler üzerinden Fitnessse test aracına ulaşır ve beklenen zaman aralığı içinde beklenen mesaj ve parametrelerin üretilip üretilmediği değerlendirilir. Gömülü kontrol yazılımı test senaryosunda tanımlandığı şekilde çalışmışsa test adımı başarılı olur. Tüm test adımlarının başarılı olması durumunda ilgili test grubu başarılı olur. Herhangi bir test adımı hata olması durumunda gerekli düzeltme ve güncellemeler yapıldıktan sonra tüm test grubu yeniden çalıştırılır.

Şekil 4’te yazılıma ait bir yeteneğin test kurgusunun mesaj akış şeması görülmektedir. Bu kurguda test aracı tarafından “Tespit Başla” mesajının gönderilmesinden belirli bir süre sonra “Tespit Başladı” mesajının alınması gerekmektedir. Bu mesajın alınmasını takiben tarama adımlarının doğru parametreler ve doğru zamanlama ile gönderilmesi ve tespit raporlarının yine doğru parametreleri içerecek şekilde periyodik olarak alınması gerekmektedir. “Tespit Dur” mesajının gönderilmesi ve “Tespit Durdu” mesajının alınması ile bu test senaryosu sona erer.



Şekil 4. Mesaj Akış Şeması Örneği

Bildiriyeye konu olan projenin gömülü kontrol yazılımının geliştirme ve testlerinin tamamlanması otuz ay kadar sürmüştür. Bu süre boyunca ortalama iki yazılım geliştirme mühendisi ve iki yazılım test mühendisi görev almıştır. Herhangi bir hata ile karşılaşılması durumunda, testlerin tamamının otomatik olarak yapılması kırk beş dakika zaman almaktadır. Aynı senaryoların otomatik test kurgusu kullanılmadan test edilmesi bu zamanı yaklaşık olarak yirmi dört saate çıkarmaktadır.

7 Kazanımlar

Bildiride bahsi geçen gömülü kontrol yazılımı çevik yöntemler ile geliştirilmiş ve otomatik olarak test edilmiştir. Bu sayede, daha erken aşamalarda çalışır yazılım ortaya çıkmıştır. Ortaya çıkan yazılıma ait hatalar daha erken aşamalarda tespit edilmiştir.

Yazılım farklı tip ve miktarda veri ile test edilerek zaman kritik fonksiyonların doğrulanması mümkün olmuştur.

En önemlisi, yazılımda yapılan değişikliklerin ardından bağlanım testlerinin yapılması kolaylaşmış ve daha kısa zaman almıştır. Kritik durumlarda bağlanım analizi yapmak ve sadece bazı testleri tekrarlamak yerine testlerin tamamı fazla zaman ihtiyacı olmadan tekrarlanmıştır.

8 Sonuç

Bu bildiride büyük ölçekli bir elektronik harp projesinde, gömülü kontrol yazılımının geliştirilmesi ve otomatik olarak test edilmesi sürecinde edinilen kazanımlar paylaşılmıştır.

9 Teşekkür

Yazarlar, gömülü kontrol yazılımının geliştirilmesinde ve otomatik test kurgusunun oluşturulmasında emeği geçen tüm ASELSAN REHİS GYM ve TMM çalışanlarına; simülatör ve fixture altyapısının geliştirilmesindeki katkılarından dolayı Uğur Zöngür'e, Hüseyin Bursalı'ya ve Özgür Kızılay'a teşekkür eder.

10 Kaynakça

1. Özzeybek, Özköse Erdoğan, Bozbey, Işık, Taşel, "Büyük Ölçekli Bir Elektronik Harp Projesinde Uç Programlama (XP Deneyimi)", UYMS'07
2. Ünal, Kahraman, "Gerçek Zamanlı Gömülü Sistem ve Yazılım Tasarımında ASELSAN Yaklaşımı", UYMS'07
3. www.agilemodeling.com Erişim tarihi: 08/05/2014
4. Gürler, Ceylan Karabörk, Sancar Yılmaz, Esen Temel, "Gömülü Yazılımlarda Müşteri Testleri", UYMS'11
5. Object Mentor Group, Fitnesse, www.fitnesse.org Erişim tarihi: 08/05/2014
6. Kent Beck: Test-Driven Development: By Example, Addison-Wesley, 2002.
7. Kent Beck: Extreme Programming, Addison-Wesley, 2003.