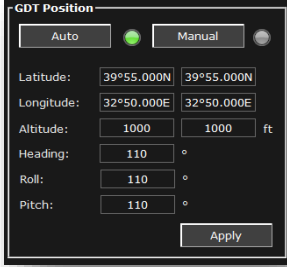


- Gereksinim yönetimi ve doğrulama faaliyetleri DO178B ve CMMI seviye 3 standartları rehber alınarak gerçekleştirilmektedir.
- Gömülü sistemler için MIL-STD -1553, ARINC-429 ve TCP/UDP fiziksel arayüz mesajlaşma testleri yarı/tam otomatik olarak gerçekleştirilmektedir.



Kontrol İstasyonu ve Hava Aracı ekipmanlarının uçuş öncesi kalibrasyon/konfigürasyon işlemleri, kullanıcı yönetimi, telemetri/telekomut ve hata kayıtlarının sorgulanması, uzak komuta ekipmanlarının yönetimi, Otomatik İniş-Kalkış Sisteminin konfigürasyon işlemleri, Hava Aracından gelen motor, veri link ve benzeri sistemlere yönelik durum verilerinin gösterimleri otomatik test senaryoları ile doğrulanmıştır. Bu senaryolar .NET Framework 4.5.1 altyapısı ile C# programlama dilinde MS Visual Studio ile taşınabilir (portable) ve sistem bağımsız olarak oluşturulmuştur.

Hava Aracını uçuşa hazırlama kapsamında ön kontrollerin yapılması, altsistem/ekipman sağlık durumlarına yönelik otomatik testlerin koşulması ve sonuçlarının raporlanması da şirketimizce hazırlanmış olan uyumluluk ve taşınabilirlik prensiplerini içeren otomatik test araçları ile yapılmıştır.

Hava aracından gelen telemetri, video ve test verilerinin, Kontrol İstasyonundan Hava Aracına gönderilen telekomut mesajlarının, Kontrol İstasyonu yazılımları arası iç mesajlaşmanın kayıt edilmesi işlemlerinin doğrulanmasına yönelik otomatik testlerin koşulması ve sonuçlarının raporlanması da veri monitörü bütünlüğüyle otomatik test araçları ile yapılmıştır.

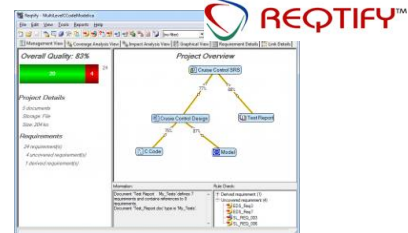


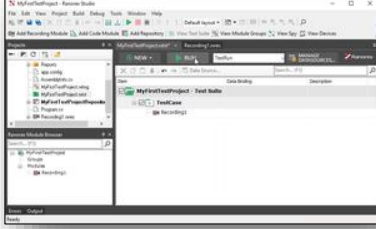
Kontrol İstasyonu ve Hava Aracı aviyonik mimarilerinde yer alan yazılım/donanım bileşenlerin tek tek doğrulanması, sistem entegrasyonu ve tümleşik doğrulanmasına yönelik otomatik testlerin koşulması ve sonuçlarının raporlanması da ham halde XML olarak veri seti şeklinde gelmiş olan paketlerin; otomatize üreteç uygulama ile baz yapıyı kullanarak otomatik test senaryosu yaratılmıştır.

Yazılım doğrulama test senaryolarının geliştirilmesi sürecinde DOORS gereksinim yönetim aracı kullanılmıştır.



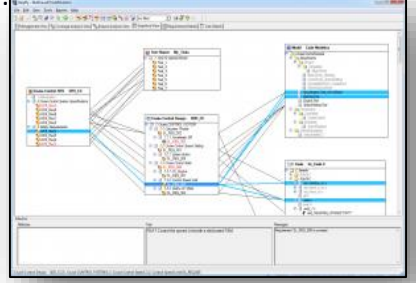
Tüm yazılım doğrulama faaliyetleri boyunca, farklı sistemlerde programlar ve proje seviyelerinde gereksinimleri, izlenebilirlikleri ve etki analizini yönetmek için Reqtify kullanılmıştır.





Kullanıcı servis arayüzlerini doğrulamak için geliştirilen test senaryoları Ranorex otomatik test aracı ile manüplasyon tekniği esas alınarak oluşturulmuştur.

C# dilinde geliştirilen test senaryoları uygulamanın fonksiyonel metodları Otomatize Edilmiş Fonksiyonel Testler ile doğrulanmıştır.



C# dilinde geliştirilen test senaryoları uygulamanın fonksiyonel metodları Otomatize Edilmiş Fonksiyonel Testler ile doğrulanmıştır. Bu bağlamda nesne tabanlı (object – oriented) yazılım esası göz önünde bulundurularak baz yapı, yeniden kullanılabilirlik (reuse) ve yazılımda miras mantığı ile çözümlenmeler uygulanmıştır.

Mevcut ve güncelleme faaliyetleri devam eden uygulamalar üzerinde, testlerin tekrar tekrar oluşturulması sırasında en az maliyetle regresyon testleri yapılmıştır.

İç ve dış veri kaynaklarından beslenen (data driven) uygulamalar simülasyon araçları, basit tablo yapıları, SQL bağlantıları kullanılarak test edilebilmiştir. API arayüzleri aracılığıyla gerçekleştirilen modeller arasındaki mesajlaşma da otomatik testlerle doğrulanmıştır.

Kullanıcı arayüzlerindeki GUI elemanları ve bu elemanların kullanıcı etkileşiminin kullanılabilirliği otomatik testlerle yapılmıştır. Grafikselle kullanıcı arabirimlerinin otomatik olarak test edilmesiyle tutarlı bir test performansı garantisi sağlanmıştır. RanorexPath tanımlayıcısı ile dinamik olarak oluşturulmuş kullanıcı arabirimleri doğru bir şekilde tanınmış ve tüm yazılı özelliklerini karşıladığından emin olunmuştur.

Geliştirilen test senaryolarının uygulamanın operasyonel ve fonksiyonel gereklerine izlenebilirlikler sağlanmıştır. Doğrulama faaliyetlerinin yaşam döngüsü boyunca izlenebilirlikleri sürekli takip edilmiş, sürdürülebilirlik ve değiştirilebilirlik yüksek düzeyde sağlanabilmiştir. Karşılanmayan, değişen gereksinim durumları zamanında kolayca uygulanabilmiştir.

Test senaryolarının her birinde testin amacı, hangi gereksinimleri doğrulamak üzere geliştirildiği, test girdileri, koşulları, beklenen sonuçları verilmiştir.

Her bir test koşumu sonrasında test raporları otomatik olarak alınıp, gözlemlenen test sonuçları, geçen, kalan, koşulan, koşulmayan test adımları bilgileri verilmiştir. Beklenen test sonucu ile uyuşmayan test adımları için **JIRA** üzerinde hata kayıtları oluşturulmuştur.