DO-178B SERTİFİKASYONUNA UYGUN OLARAK GELİŞTİRİLEN İNSANSIZ HAVA ARACI(İHA) UÇUŞ KONTROL BİLGİSAYARI YAZILIMININ GEREKSİNİM GÖZDEN GEÇİRME SÜRECİ, BU SÜREÇTE BULUNAN HATALARIN ANALİZİ VE BU SÜREÇTEN ÖĞRENİLMİŞ DERSLER

İbrahim Seyfullah BABAARSLAN[1]

TUSAŞ Türk Havacılık ve Uzay Sanayii A.Ş., Ankara

ÖZET

Bu bildiride, İHA Uçuş Kontrol Bilgisayarına ait yazılımın DO-178B sertifikasyon isterlerine uygun olarak geliştirilen doğrulama yaklaşımının bir parçası olan gereksinim gözden geçirme sürecinde karşılaşılan hataların analizi ve bu süreçten öğrenilen dersler sunulmaktadır. Yazılım doğrulama süreci, hava aracının uçuş kontrol bilgisayarının belirlenen gereksinimlere uygun davranış sergilediğini ve beklenmeyen sonuçlar üretmediğini doğrulamak için kullanılan bir süreçtir. Bu sürecin en önemli bölümlerinden birisi yazılım gereksinimlerin sistem gereksinimleriyle tam uyumlu ve eksiksiz olduğunun kontrol edildiği yazılım gereksinimleri gözden geçirme sürecidir. Bildiride ilk önce yazılım mimarisi anlatılacak, daha sonra yazılım mimarisine uygun olarak yazılan gereksinimler için yapılan gereksinim gözden geçirme sürecinden bahsedilecektir. Sonrasında gereksinim gözden geçirme için kullanılan kontrol listesi maddeleri anlatılacak, gereksinimlerin gözden geçirmesinde bulunan hataların kritiklik seviyesine göre sınıflandırması ve en yaygın bulunan hatalar üzerinde analizler yapılacaktır. Son olarak da DO-178B sertifikasyon sürecindeki yazılım gözden geçirme sürecinden kazanılmış dersler sunulacaktır.

GİRİŞ

Sabit Kanatlı Askeri İHA'lar için geçerli olan STANAG 4671'e göre yazılım ve kompleks donanımların uyması gereken geliştirme seviyeleri Tablo 1'de verilmiştir[1]. İlgili yazılımların ve birimlerin SAE-ARP-4761 "Guidelines and Methods for conducting the Safety Assessment Process on Civil Airborne Systems and Equipment" dokümanı referans alınarak gerçekleştirilen emniyet analizlerine göre yazılımlara geliştirme seviyesi hedefi atanır[2]. Uçuş Kontrol Bilgisayarı Yazılımı yapılan emniyet analizlerine göre "Ölümcül" kategoridedir ve bu sebepten ötürü İHA'lar için en yüksek Geliştirme Seviyesi olan DAL-B uyumu göstermesi gerekmektedir.

[1] İHA Sistemleri, Yazılım Doğrulama., E-posta: isbabaarslan@tai.com.tr

Tablo 1: STANAG 4671 Geliştirme Güvence Seviyesi Hedefleri Tablosu

Sistem ve Mimarinin her bir birimi için Gerekli Geliştirme Güvence Seviyesi Atanması		Gerekli Geliştirme Güvence Seviyesi
Kritiklik Seviyesi	Ölümcül	DAL B
	Tehlikeli	DAL C
	Önemli	DAL D
	Az Önemli	DAL E
	Emniyet Etkisi Yok	DAL E

DO-178B standardı bir sistemde meydana gelen hatanın sonuçlarının kritiklik seviyesine göre yapılması gereken gereklilikleri tanımlar. Bu seviyeler, oluşan bir hatanın can kaybı ve ciddi maddi zararlar doğurması ile hiçbir etkisinin olmaması arasında değişmektedir[3]. Her bir seviye için beklenen isterler Tablo 2'de belirtilmiştir:

Tablo 2 : DO-178B Kritiklik Seviyelerine Göre İsterler Tablosu

DO-178 Özelliği	**Seviye A**	**Seviye B**	**Seviye C**	**Seviye D**
Bağımsızlık Seviyesi	Yüksek	Orta	Az	Çok Az
Yazılım Planları	Evet	Evet	Evet	Evet
Yazılım Standartları	Evet	Evet	Evet	Hayır
Satır Yapısal Kapsama	Evet	Evet	Evet	Hayır
Karar Yapısal Kapsama	Evet	Evet	Hayır	Hayır
UKK Yapısal Kapsama	Evet	Hayır	Hayır	Hayır
Doğrulanabilir ÜS Gereksinim	Evet	Evet	Evet	Hayır
Doğrulanabilir AS Gereksinim/Kod	Evet	Evet	Hayır	Hayır
ÜS-AS ve AS-Kod İzlenebilirliği	Evet	Evet	Evet	Hayır
AS Gereksinim Test Kapsaması	Evet	Evet	Evet	Hayır
Kod Gözden Geçirmeleri	Evet	Evet	Evet	Hayır
Konfigürasyon Yönetimi	Yüksek	Yüksek	Orta	Az
YKG Geçiş Kriterleri	Evet	Evet	Evet	Hayır
Mimari, Algoritma Doğrulama	Evet	Evet	Evet	Hayır

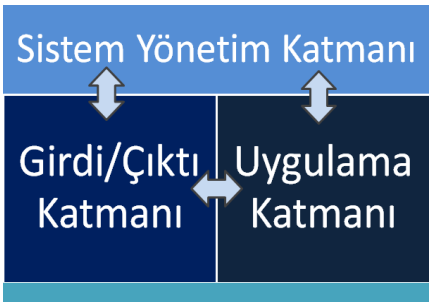
Bir yazılımda kritiklik seviyesi arttıkça yazılımla ilgili detaylı dokümanlar, testler ve analizler de artmaktadır. Bunların sonucundan maliyet artmaktadır. Bağımsızlık ve gözden geçirmelerin sürece dahil olmasıyla daha kontrollü bir yapı kurulmuş ve daha kaliteli ürünler ortaya çıkmıştır.

Tablo 2'de verilen DO-178B Kritiklik Seviyelerine Göre İsterler Tablosuna göre Bağımsızlık Seviyesi kriteri "Orta" olarak istenmesine rağmen, geliştirdiğimiz İHA için bu seviye "Yüksek" olarak sağlanmıştır. Geliştirici ve doğrulayıcı ekiplerin tamamen ayrılması, yapılan değişikliklerin hem yazılımcılar hem doğrulamacılar tarafından çapraz doğrulanması sağlanmıştır. Yazılım Planlarının süreçte olması kriteri yazılım geliştirme yaşam döngüsü süreçleri içerisinde yer alan tüm süreçlere ait yazılım plan dokümanlarının yazılmış ve herkesin erişebileceği şekilde açık bulunmasıyla sağlanmaktadır. Bu planlar; yazılım sertifikasyon irtibat süreci planı, yazılım kalite güvence planı, yazılım konfigürasyon yönetim planı, yazılım geliştirme planı ve yazılım doğrulama planıdır. Yazılım standartları olarak yazılım gereksinimleri standardı, yazılım kodlama standardı vs. standartlar bulunmaktadır. Yapısal kapsama analizi için Seviye B için yeterli görünen kriter olarak "Karar Yapısal Kapsama" kullanılmaktadır. Yazılımda doğrulanabilir üst seviye gereksinimler ve bu gereksinimlerle ilişkili doğrulanabilir alt seviye gereksinimler bulunmaktadır. Üst seviye gereksinimlerin alt seviye gereksinimler ile ve alt seviye gereksinimlerin kod ile izlenebilirliği kurulmuştur. Alt seviye gereksinimlerin doğrulandıkları testler ile izlenebilirlikleri kurulmuştur. Kodlar geliştirildikten sonra geliştiren kişi dışında bir kişi tarafından kod geliştirme kontrol tablosuna göre gözden geçirilmektedir. Yazılıma ait tüm ürünler(kod, test, plan dokümanları, gereksinimler vs.) konfigürasyon yönetim araçlarında tutulmakta, yetkisiz erişimleri önlemek için her bir ürün için bu ürünü değiştirme, okuma vs. yetkiler tanımlanmaktadır. Birbirine bağlı olarak ilerleyen süreçlerin kesintisiz devam edebilmesi için bir sürecin başlaması ve tamamlanması için gerekli olan geçiş kriterleri tanımlanmakta ve uygulanmaktadır. Mimari ve algoritma doğrulamaları entegre ve bölüt bazlı testler ve kod gözden geçirmeler ile sağlanmaktadır.

İHA Uçuş Kontrol Bilgisayarına ait yazılımın gereksinimlerinin gözden geçirme süreci üzerine hazırlanan bu bildiride öncelikle yazılım mimarisi açıklanacaktır. Ardından hava aracının yazılımına ait bölüt bazlı ve entegre gereksinimlerden bahsedilecektir. Sonrasında ilk kez geliştirilen veya güncellenen bir gereksinim seti için gereksinimlerin gözden geçirilme yöntemi anlatılacaktır. Değişiklik ve hata yönetiminin nasıl işlediği açıklanacak, gözden geçirme kontrol listesi tanıtılacak ve bu gözden geçirme kontrol listesine göre tüm gereksinimlere verilen yorumların tip, öncelik ve kontrol listesi maddesine göre karşılaştırması yapılacaktır. Son olarak da yazılım gereksinimleri gözden geçirme sürecinden öğrenilen dersler açıklanacaktır.

YAZILIM MİMARİSİ

Uçuş Kontrol Bilgisayarı Yazılımı(UKBY), gerçek zamanlı işletim sistemi üzerinde çalışan, belli işlevleri yerine getirmek üzerine ayrılmış bölütlerin(partition) zaman ve adres bazında çalışmaları üzerine tasarlanmış yazılımdır. UKBY, işlevsel fonksiyonların yönetildiği Uygulama Katmanı, Hava Aracındaki diğer ekipman/altsistemlerle ve Kontrol İstasyonuyla haberleşmeyi sağlayan Girdi/Çıktı Katmanı ve katmanların sağlık bilgilerini ve çalışma performanslarını izleyen Yönetim Katmanından oluşur. Yazılım mimarisine ait gösterim Şekil 1'de verilmektedir.



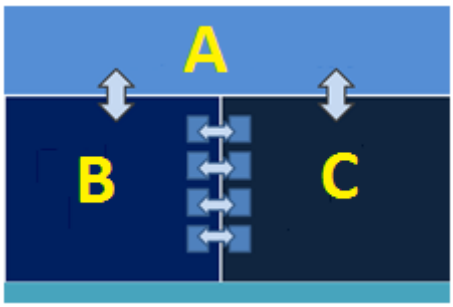
Şekil 1: Uçuş Kontrol Bilgisayarı Yazılım Mimarisi

Her bir katman işlevine göre farklı alt katmanlara ayrılmaktadır. Uygulama Katmanında model tabanlı yazılım geliştirme aracı ile kodlar üretilmektedir. Girdi/Çıktı ve Yönetim Katmanı ise manuel kodlarla yazılmıştır. Her bir katmanın işlevsel gereksinimlerini açıklayan bölüt bazlı gereksinimler ve uçuş kontrol bilgisayarının girdi/çıktı katmanlarından yazılan ve tüm bölütlerin birlikte çalışması durumundaki işlevsel gereksinimler ile uçuş kontrol bilgisayarının fonksiyonalitesi tanımlanmaktadır.

GEREKSİNİMLER

Bölüt Bazlı Gereksinimler

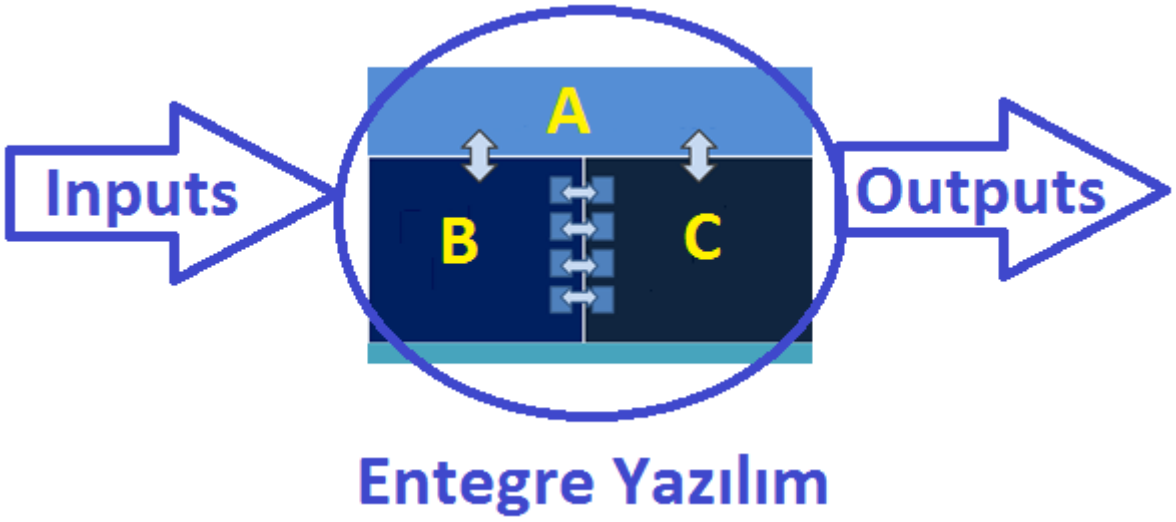
Her bir bölüt için o bölütün işlevini anlatan yazılım gereksinimleri bulunmaktadır. Bu yazılım gereksinimleri o bölüt dışındaki bölütleri yok sayarak sadece o bölütün girdisi ve çıktısı cinsinden yazılmıştır. Her bir bölüt için farklı gereksinimler, o bölütün haberleştiği arayüzler cinsinden yazılmaktadır. Bölüt bazlı gereksinim yaklaşımı Şekil 2'de gösterilmektedir.



Şekil 2: Bölüt Bazlı Gereksinim Yaklaşımı

Entegre Gereksinimler

Bölüt bazlı yazılan gereksinimlerin yanında bütün bölütlerin birlikte çalışmasını sağlayacak entegre seviyede yazılmış gereksinimler de bulunmaktadır. Bu gereksinimlerin testleriyle birlikte tüm bölütlerin beraber doğru olarak çalıştığı ve beklenen sonuçları ürettiği doğrulanmış olur. Entegre gereksinim yaklaşımı Şekil 3'te gösterilmektedir.



Şekil 3: Entegrasyon Gereksinim Yaklaşımı

GEREKSİNİM GÖZDEN GEÇİRME SÜRECİ

Yazılım geliştirme ekibi tarafından bölüt veya tüm yazılım bölütlerinin birlikte çalışması durumundaki davranışlarını açıklayan entegre seviyede yazılan gereksinimler yazılım test sürecine girdi olarak gelmektedir. Bu gereksinimler gözden geçirme kontrol listelerine göre gözden geçirilmektedir. İlk kez geliştirilen bir yazılım gereksiniminin durumunun yönetimiyle ilgili aşamalar Şekil 4'te gösterilmektedir.

Şekil 4: İlk Kez Geliştirilen Yazılım Gereksinimin Durumunun Yönetimi

İlk kez geliştirilen bir yazılım gereksinimi sistem gereksinimlerine uygun olarak hazırlanmaktadır. Gözden geçirme sürecinden önce bu gereksinime ait anahat alınmış olması ve sistem gereksinimlerinin hazır halde bulunması vs. giriş koşulları bulunmaktadır. Geçiş kriter formu gereksinim seti hazırlanmadan önce hazırlanması gereken referans dokümanların hazır bulunmasını ve gözden geçirilecek doküman için anahat alınmasını garanti altına alan bir dokümandır. Bu form hazırlandıktan sonra gözden geçirme yapacak kişiler belirlenir. Bu kişilerin farklı disiplinlerden seçilmesi farklı bakış açılarının dokümana yansıtılması için önemlidir. Standardımıza göre bir yazılım gereksinim seti gözden geçirmeye en az bir yazılım geliştirici ve en az bir yazılım testçi katılmalıdır. Gözden geçirme sürecinde "Moderator", "Author" ve "Reviewer" rolleri vardır. Moderator, gözden geçirme süreciyle ilgili duyurunun yapılması ve sürecin yönetilmesi, gözden geçirme sürecinde bir sorun olduğunda bu sorunun çözülmesinden sorumludur. "Author", gözden geçirilecek dokümanı hazırlayan kişidir. "Reviewer(s)" gözden geçirmeyi belirli kontrol listelerine göre yapacak kişi/kişilerdir. Gözden geçirme sürecinde "Reviewer(s)" olarak atanan kişiler belirli kontrol listelerine göre gereksinimleri gözden geçirerek hataları belirlerler ve bunu formlarla bildirirler. Gözden geçirme toplantısında tüm yorumların üzerinden gidilerek yorumlarla ilgili Kabul/Ret kararı verilir ve sonrasında kabul edilen yorumlar "Author" tarafından düzeltilir. Yorum veren kişi hatanın düzeltildiğini kontrol eder, bir sorun yoksa

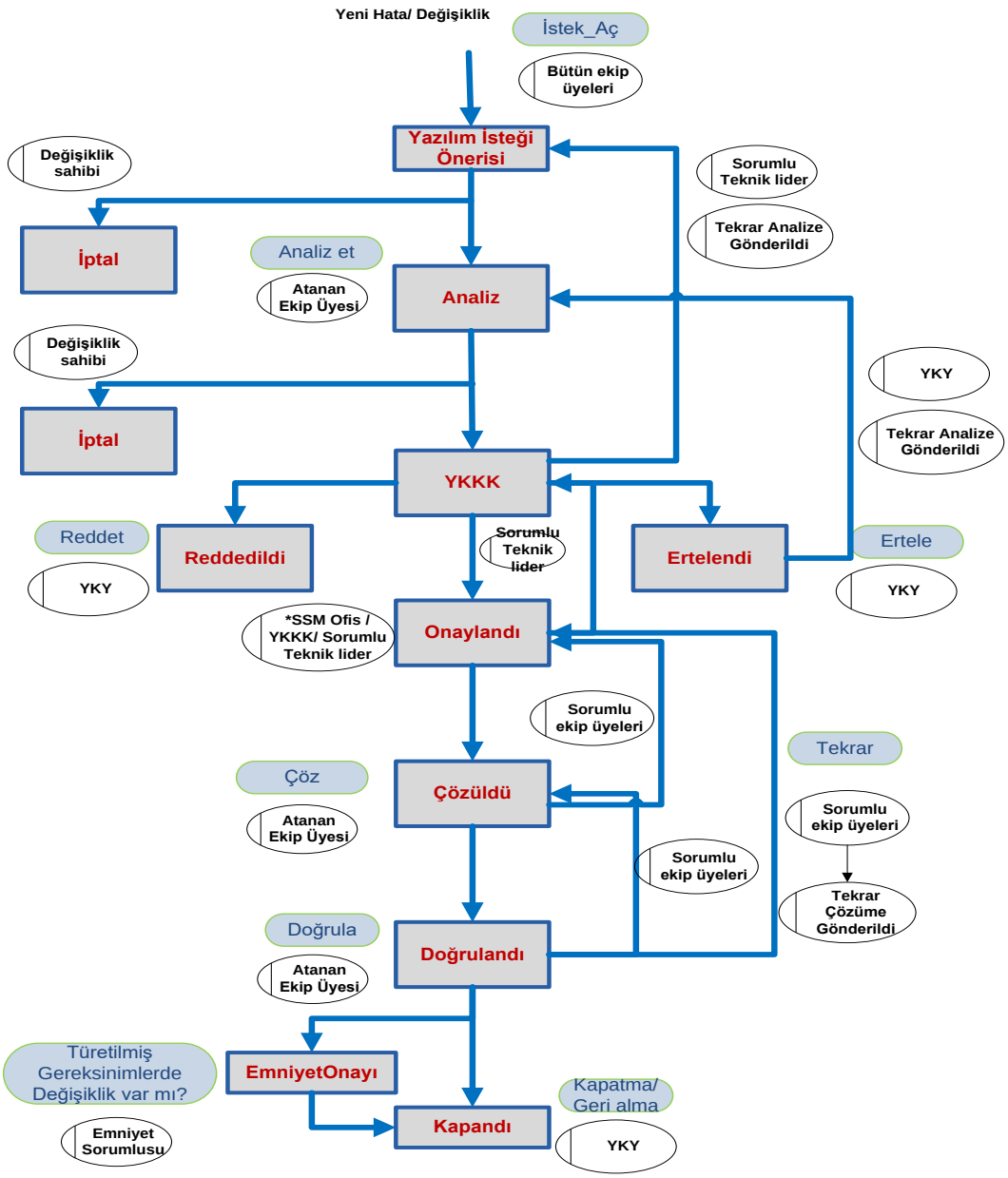
gözden geçirme süreci kapatılır ve süreçle ilgili kalite kontrollerinden sonra gözden geçirilen doküman yayımlanır.

Sistem gereksinimlerinde yapılan bir güncelleme sonucunda değişiklik yapılması gerekecek yazılım gereksinimleri için de yazılım değişiklik isteği açılmakta ve hata/istek yönetim aracı vasıtasıyla değişikliğin, yapan kişi dışında biri tarafından kontrol edilmesi sağlanmaktadır. Benzer şekilde yazılım gereksiniminde bir iyileştirme ya da hata düzeltme de yapılabilir. Yayımlanmış bir yazılım gereksiniminin durumunun yönetimiyle ilgili aşamalar Şekil 5'te gösterilmektedir.

Şekil 5: Yayımlanmış Gereksinim Durumunun Yönetimi

DEĞİŞİKLİK VE HATA YÖNETİMİ

Sistem gereksinimlerinin değişmesiyle birlikte ilgili sistem gereksinimlerine karşılık yazılan yazılım gereksinimleri ve kod da değişir. Yazılımın herhangi bir parçasına (kod, test durumu, gereksinim vb.) bir değişiklik yapılması gerektiğinde ilgili parça için yazılım değişiklik talebi açılır. Yazılım değişiklik talebi, açıldıktan sonra değişiklik yapacak kişi tarafından analiz edilerek yazılım değişiklik kontrol kuruluna gönderilir veya iptal kararı alınır. Yazılım değişiklik kontrol kurulunda değişikliğin yapılmasının gerekliliği, neye dayandığı vb. konular değerlendirilerek değişikliğin kabul edilmesine veya reddedilmesine karar verilir. Kabul edilen bir değişiklik isteği, sorumlu kişi tarafından işleme alınır ve doğrulanmak üzere bu işlemi yerine getiren kişi dışında bir kişiye atanır. Değişikliğin doğru yapıldığı kontrol edildikten sonra değişiklik isteğinin süreçsel kontrolü de yazılım kalite süreç yöneticileri tarafından yapılarak istek kapatılır. Yayımlanmış bir üründe değişikliklerin kontrollü yapılabilmesi için ürün üzerinde sadece kabul edilmiş bir değişiklik talebi bulunması durumunda değişikliğe izin verilmektedir. Bu kontrol, konfigürasyon yönetim aracı ile değişiklik/hata yönetim aracının entegre çalışması ile sağlanmaktadır. Değişiklik veya hatanın yaşam döngüsü boyunca geçirdiği durumlar Şekil 6'da gösterilmektedir.



Şekil 6: Değişiklik/Hata Yaşam Döngüsü

GÖZDEN GEÇİRME KONTROL LİSTESİ

Yazılım gereksinimleri herkes tarafından aynı standardın oluşturulması için yazılım gözden geçirme kontrol listesine göre gözden geçirilmektedir. Bu liste yazılım gereksinimleri ile ilgili yapılabilecek tüm hataları içeren bir kontrol listesidir. Yazılım gereksinimleri gözden geçirme için kullanılan gereksinim gözden geçirme listesi Tablo 4: Yazılım Gereksinimleri Gözden Geçirme Kontrol Listesi tablosunda verilmiştir.

Tablo 4 : Yazılım Gereksinimleri Gözden Geçirme Kontrol Listesi

Gözden Geçirme Tarih(i/leri):		Proje:
Gözden Geçiren(ler):		Gözden Geçirilen Doküman:
DO-178B Seviyesi:		YKP:
SRS teki Bölüm Adı:		SRS teki Bölüm Numarası:

	Genel	**(Evet/Hayır/UD)**	**Yorum #**
1	Üst seviye yazılım gereksinimleri yazılım konfigürasyon yönetimi kontrolü altındadır.		
2	Her yazılım gereksinimin benzersiz bir kimliği vardır.		
3	Her yazılım gereksinimi yalnızca bir kez belirtilir.		
4	Tüm kısaltmalar tutarlı bir şekilde tanımlanmış ve kullanılmıştır.		
	Sistem Gereksinimleri		
5	Üst seviye gereksinimler sistem gereksinimleriyle uyumludur.		
6	Üst seviye gereksinimler sistem gereksinimleriyle izlenebilir olmalıdır.		
7	Üst seviye yazılım gereksinimleri tamdır; yani yazılım gereksinimleri, yazılıma atanan işlev, performans ve güvenlikle ilgili sistem gereksinimlerini belirtmek için yeterlidir.		
	Yazılım Gereksinimleri		
8	Türetilmiş yazılım gereksinimleri Sistem Emniyet Değerlendirme Grubu tarafından gözden geçirilmiştir.		
9	Üst seviye gereksinimler doğru ve tutarlıdır.		
10	Algoritmalar doğrudur.		

11	Bir nesne veya fonksiyona tek bir isim verilmiştir.		
12	Bir nesne ya da fonksiyon kendine özgü özellikleriyle ayrı ayrı tanımlanmıştır.		
13	Değişkenlerin değer aralıkları ve sabitler birimleriyle verilmiştir.		
14	Verilen tüm listeler tamdır, vb., bunun gibi veya benzeri ifadeler kullanılmamalıdır.		
15	Yazılımdaki durumların tamamı eksiksiz tanımlanmıştır, durumlar arası geçişler açık ve tutarlıdır.		
	Donanımla İlgili		
16	Üst seviye gereksinimler hedef bilgisayar ile uyumludur.		
	Doğrulanabilirlik		
17	Üst seviye gereksinimler doğrulanabilirdir.		
18	Tüm üst seviye yazılım gereksinimleri uygun olan yerlerde toleranslarıyla birlikte nicel terimleriyle belirtilmiştir.		
	Standarda Uygunluk		
19	Üst seviye gereksinimler gereksinim standartlarıyla uyumludur.		

YORUMLAR SAYFA ____ - ____

#	**Gözden Geçirenlerden Yorumlar.**

Geçti:		**Kaldı:**	
Gözden Geçiren İmza:		**Tarih:**	

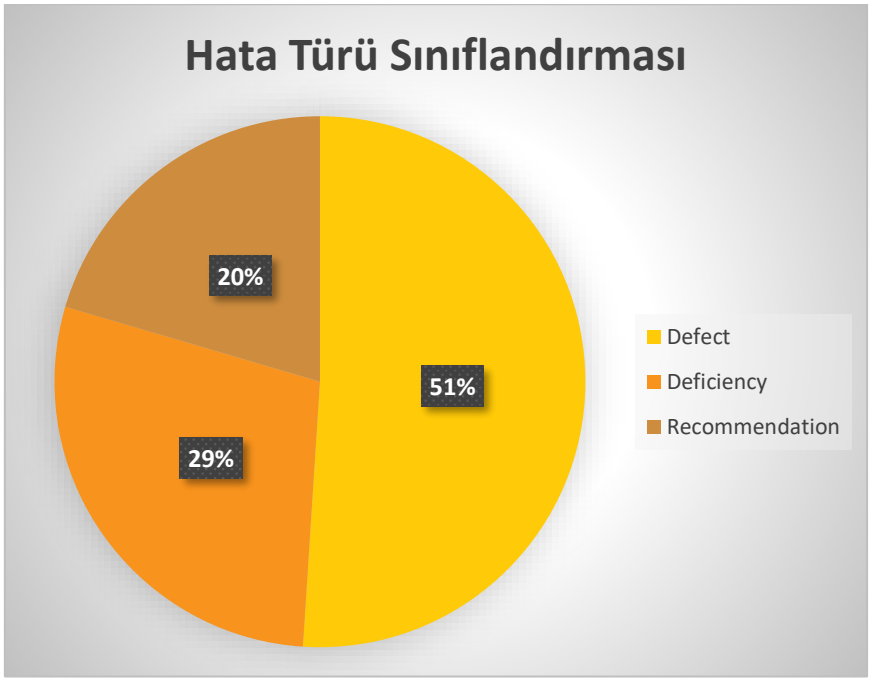
Gereksinim gözden geçirme kontrol listesine göre gözden geçirilen gereksinimlere verilen yorumlar eğer herhangi bir kontrol maddesiyle ilişkili ise bu yorum “Hata(Defect)” türünde seçilmektedir. Bir eksikliğe işaret eden fakat kontrol listesi maddeleri ile ilişkilendirilemeyen hatalar “Eksiklik(Deficiency)” türünde seçilmektedir. Sadece bir öneri olarak verilen yorumlar ise “Öneri(Recommendation)” türünde seçilmektedir.

Verilen yorumun önceliği ise “Ölümcül(Catastrophic)”, “Tehlikeli(Hazardous)”, “Önemli(Major)”, “Az Önemli(Minor)” ve “Etkisi Yok(No Effect)” olarak gruplandırılmaktadır.

Hata(Defect) tipinde verilen yorumların hangi kontrol maddesi ile ilişkili olduğu seçilmekte ve buna göre bir kontrol listesi doldurulmaktadır.

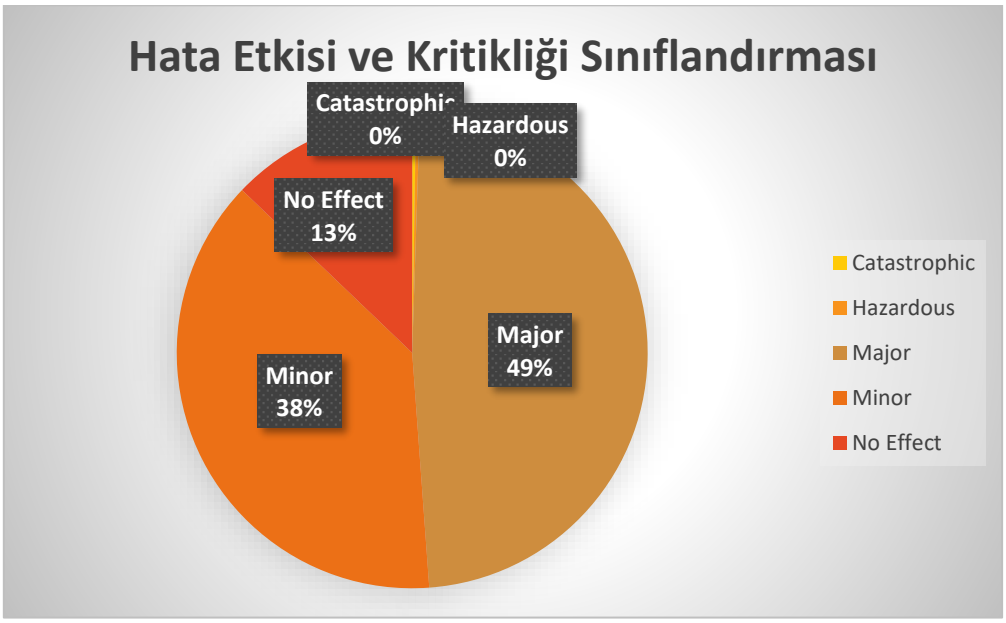
DEĞERLENDİRME VE SONUÇ

3038 yazılım gereksinimi üzerinde yapılan gözden geçirme sonuçlarına göre toplamda 778 tane yorum verilmiş, bu yorumlardan 397’si Hata(Defect), 222’si Eksiklik(Deficiency), 159’u Öneri(Recommendation) olarak girilmiştir. Herhangi bir kontrol listesine göre hata bildirilen yorumlar ya Hata(Defect) ya Eksiklik(Deficiency) olarak seçilmiştir. Bu hata türlerine ilişkin grafik Şekil 7’de gösterilmektedir.



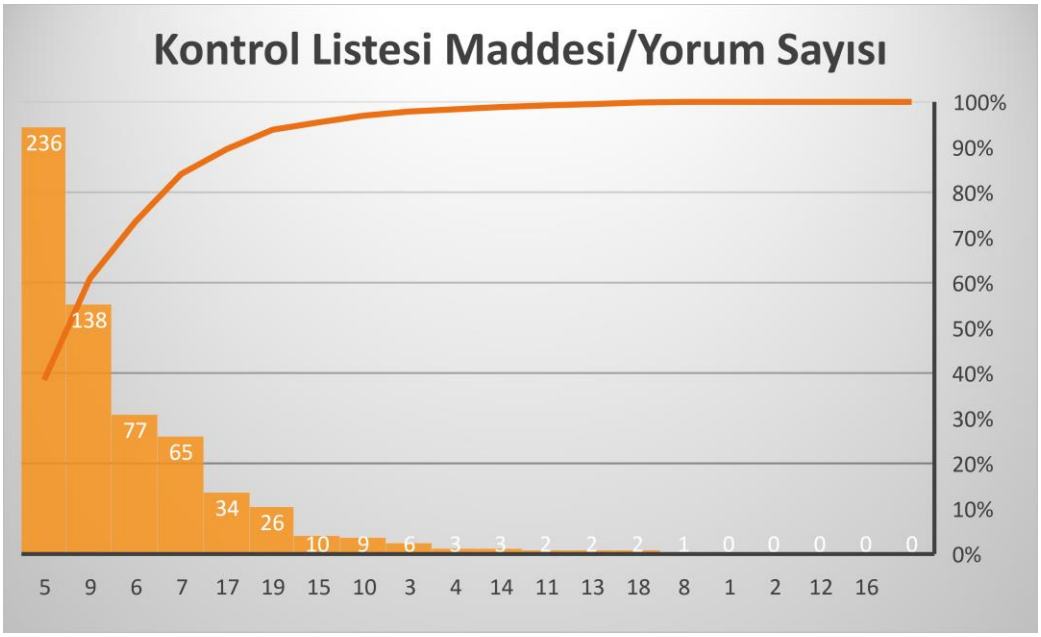
Şekil 7 : Hata Türü Sınıflandırması

Yorumlar hatanın etkisi ve kritikliği anlamında 2 yorum Ölümcül(Catastrophic), 2 yorum Tehlikeli(Hazardous), 376 yorum Önemli(Major), 298 yorum Az Önemli(Minor), 100 yorum Etkisi Yok(No Effect) olarak girilmiştir. Bu öncelik ayrımında reviewerlar arasında ortak bir dil kullanılmadığı, bazı kişilerin Ölümcül(Catastrophic) bulduğu bir hatanın başka biri tarafından Önemli(Major) bulunabildiği görülmüştür. Hata etkisi ve kritikliğine ilişkin sınıflandırma ile ilgili grafik Şekil 8’de gösterilmektedir.



Şekil 8 : Hata Etkisi ve Kritikliği Sınıflandırması

Yorum verilen kontrol listesi maddeleri belirlendiğinde 5, 9, 6, 7, 17 ve 19. Maddelerin yoğunlukta olduğu görülmektedir. En çok yorum verilen madde sistem gereksinimleri ile yazılım gereksinimleri arasındaki farklılıklara işaret eden kontrol listesi maddesi 5'tir. Bu madde sistem gereksinimlerinin yanlış anlaşılması, sistem gereksinimlerin değiştirilmesi sebebiyle doğru sistem gereksinimi versiyonunun referans olarak kullanılmaması, benzer modüller için daha önce yazılan gereksinim setlerinden kopyalanan fakat ilgili modüle göre güncellenmesinin atlanması vs. sebeplerden kaynaklanmaktadır. Gereksinimlerin tam ve eksiksiz tanımlanması ile ilgili kontrol listesi maddesi olan 9 en çok yorum verilen 2. maddedir. Sonra ise yazılım gereksinimlerinin sistem gereksinimler ile izlenebilirliğinin kurulduğunu ve bu izlenebilirliğin hatasız olmasıyla ilişkili kontrol listesi maddesi 6 gelmektedir. Sistem gereksiniminde anlatılan işlevselliğin tam olarak yazılıma aktarıldığı ile ilişkili kontrol maddesi 7 en çok yorum verilen 4. Madde olarak karşımıza çıkmaktadır. Sonrasında yazılım gereksinimlerinin doğrulanabilirliği ile ilgili 17. Madde ve sonrasında gereksinimlerin belirlenmiş gereksinim standardına uyumlu olup olmadığı ile ilişkili 19. Madde gelmektedir.



Şekil 9 : Kontrol Listesi Maddesi/Yorum Sayısı

Gözden geçirme süreci, sistem gereksinimlerinden yazılan yazılım gereksinimlerinin en başta sistem gereksinimleriyle uyumlu, eksiksiz ve doğrulanabilir olduğunu garanti altına alan bir süreçtir. Bu aşamada gözden geçirme için gerekli zamanın ayrılması, yapılan bir hata sonrasında koda ve teste de yansıyacağı için projenin ileri aşamasında ortaya çıkabilecek hataların erken bulunmasını sağlamıştır.

Gözden geçirme sürecindeki geçiş kriter formu yazılım gereksinimlerinin hazır olduğunu garanti altına aldığı için anlaşmazlıklar noktasında katkıda bulunmuştur.

Gözden geçirme sürecine farklı disiplinlerden insanların katılması dokümanların kapsamlı şekilde gözden geçirilmesini ve farklı bakış açılarının dokümana yansıtılmasını sağlamıştır.

Gözden geçirme kontrol listelerinde hata türü ve önceliklendirme derecesi ile ilgili kafa karışıklığı olduğu görülmektedir, bu konuda herkesin ortak hareket etmesi için hata türü ve önceliklendirmesi ile ilgili ortak dilin kullanılması anlamında eğitim verilebilir. Gözden geçirme toplantılarında çok farklılık içeren önceliklendirmelerin örnek teşkil etmesi açısından moderatör tarafından düzeltilebilir.

Gözden geçirme süreci, projenin ilerleyen aşamalarında ortaya çıkacak sorunların erken teşhisini sağladığı için diğer tüm süreçler kadar önemsenmeli ve kişilere bu işi yapmaları için gerekli zaman tanınmalıdır. Gözden geçirme süreci, doğrulama ve standartlar ile ilgili eğitimler alınarak bu sürecin daha verimli olması sağlanabilir.

Kaynaklar

1. NATO, STANAG 4671 Unmanned Aircraft Systems Airworthiness Requirements (USAR) (2017).
2. SAE, ARP 4761 – Guidelines and Methods for conducting the Safety Assessment Process on Civil Airborne Systems and Equipment (1996).
3. RTCA, DO-178B, Software Considerations in Airborne Systems and Equipment Certification (1992).