T.C

ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ

MÜHENDİSLİK - MİMARLIK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

2023-2024 GÜZ YARIYILI

YAZILIM MÜHENDİSLİĞİ DERSİ

ISO/IEC 12207:1995, ISO/IEC 15026, Yazılım Yaşam Döngüsü Süreçleri, Yazılım Geliştirme Standartları

Grup-4
152120201039 Hakan Yavaş
152120201098 Emre Kart
152120201054 Alperen Güneş
152120201058 Ayşe Ayhan
152120201086 Abdulkadir Sönmezışık
(Tüm üyeler katkı sağlamıştır ve katkı oranları eşittir.)

Dr. Öğr. Üyesi Uğur GÜREL

ARALIK 2023

İçindekiler

1. GİRİŞ

1.1. Özet ve Amaç

Bu rapor, ISO - International Organization for Standardization tarafından yayınlanan ISO/IEC 12207:1995 ve ISO/IEC 15026 standartlarının tanımları, içerikleri, kullanım alanları bilgilerini içermektedir. Raporda sağlanacak bilgiler doğrultusuna bu standartların anlaşılması hedeflenmektedir.

1.2. Konu ve Kapsam

Raporda Özet ve Amaç kısmında bahsedilen 2 standart hakkında detaylı bilgiler yer alacaktır. ISO/IEC 12207:1995 standardı yazılım geliştirme süreçlerini standart haline getirirken, ISO/IEC 15026 standartı ise yazılım geliştirme süreçlerindeki güvenlik ihtiyaçlarını, bunların nasıl standart haline getirileceği konusunda rehberlik etmektedir. Bu halde bakıldığında iki standart birbirinin tamamlayıcısı konumunda bulunmaktadır, çünkü yazılım geliştirme süreçlerinin içinde izlenilen yola ek olarak yazılımın güvenli bir şekilde geliştirilmesi de mevcuttur.

1.3. Anahtar Kelimeler

ISO/IEC 12207:1995, ISO/IEC 15026, Yazılım Yaşam Döngüsü Süreçleri, Yazılım Geliştirme Standartları

2. ISO/IEC 12207:1995

2.1. Tanım

ISO/IEC 12207:1995 “ISO/IEC 12207:1995 Information Technology - Software Life Cycle Processes”, Yazılım geliştirme alanında yayınlanmış, geliştirilecek yazılım sistemlerinin yaşam döngüsü süreçlerini belirli bir plan dahilinde düzenlemek amacıyla oluşturulmuş uluslararası bir standarttır. International Organization for Standardization (ISO) ve International Electrotechnical Commission (IEC) tarafından 1995 Aralık ayında yayınlanmıştır. Yayınlandığı süreçten günümüze dek standart üzerinde değişiklikler yapılmış ve bu değişiklikler farklı versiyonlar ile yayınlanmıştır. Tüm versiyonlar listelendiği gibidir:

1. ISO/IEC 12207:1995 Information technology, Software life cycle processes
2. ISO/IEC 12207:2008 Systems and software engineering, Software life cycle processes

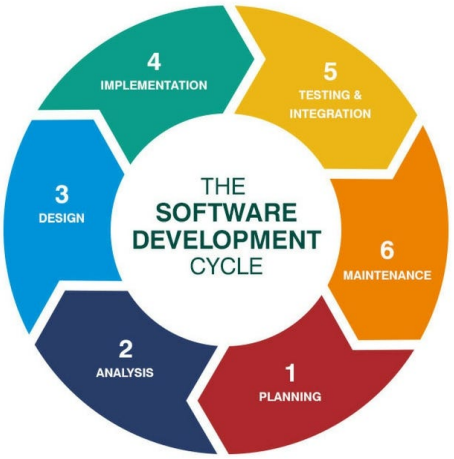
3. ISO/IEC/IEEE 12207:2017 Systems and software engineering, Software life cycle processes (Yayında olan versiyon)

Kaynak 1: https://www.iso.org/standard/21208.html

2.2. Kapsam ve Amaç

ISO/IEC 12207:1995, bir yazılımın geliştirilme sürecinin başlangıç evresinden yazılımın sürdürüldüğü tüm süreçleri kapsar. Planlama, analiz, tasarım, implementasyon, birim test, entegrasyon testleri ve destek süreçlerinin yönetimini standart haline getirir. ISO/IEC 12207, yazılım mühendisliği ve yazılım geliştirme ile ilgilenen tüm alanlarda proje yönetimi ve geliştirme süreçlerin iyileştirilmesi konularında rehberlik sağlayan bir belge olarak kabul edilir.

ISO/IEC 12207:1995'in ortaya çıkmasındaki başlıca sebepler arasında standart öncesi geliştirilmekte olan yazılımların bir standart içerisinde geliştirilmemesinden kaynaklanan yazılımda ortaya çıkan performans sorunları, güvenlik açıkları, müşteri ile yaşanan anlaşmazlıklar yer almaktadır.



Kaynak:https://medium.com/@jilvanpinheiro/software-development-life-cycle-sdlc-phases-40d46afbe384

2.3. ISO/IEC 12207 Standartının Genel Yapısı

ISO/IEC 12207 standartının ele aldığı başlıklar genel yapısı itibariyle şu şekildedir:

2.3.1. Planlama

Planlama süreci, yazılımın geliştirilmesinin planlandığı süreçtir. ISO/IEC 12207 standartı; proje, kaynak, risk, kalite ve dokümantasyon süreçlerinin planlanmasına etkin rol oynar.

2.3.2. Gereksinim Analizi

Gereksinim analizi süreci, yazılım geliştirme süreçlerinin en önemli aşamalarındandır. Bir yazılım elde edilen gereksinimler üzerinden şekillenir ve bu aşamada yapılan eksiklikler daha sonrasında yazılım geliştirme sürecinin aksamasına neden olur. ISO/IEC 12207 standatı gereksinim analizi süreçlerinin dokümente edilmesi gerektiğini belirler.

2.3.3. Tasarım

Tasarım süreci, yazılımın gereksinim analizinden elde edilen veriler doğrultusunda tasarlandığı aşamadır. Gereksinim analizi verilerinden tasarıma başlandığı için gereksinim analizi süreci, tasarım sürecinin temel dayanak noktasıdır. Tasarım süreci; veritabanı tasarımı, yazılım tasarımı, arayüz tasarımı gibi süreçleri bünyesinde bulundurur. ISO/IEC 12207 standartı tasarım süreçlerinin geliştirilmesi ve tasarımın gereksinim analizinde elde edilen verilerle doğrulanmasını, yapılan işlemlerin ne şekilde dokümente edilmesi gerektiğini açıklar.

2.3.4. İmplementasyon

İmplementasyon süreci, tasarım sürecinin tamamlanmasıyla ortaya çıkan uygulamanın realize edildiği aşamadır. Bu süreçte kodlama, veritabanının ve arayüzlerin oluşturulması, sistemlerin birbirine entegre edilmesi gibi adımlar bulunur.

2.3.5. Test

Test süreci, implemente edilmekte/edilmiş olan sistemin belirli parçalarının (birim test-unit test) ve sistemin diğer tüm modüller ile test edilmesinin (entegrasyon testi-integration test) gerçekleştirildiği süreçtir. ISO/IEC 12207 standardı test süreci için testlerin ne şekilde yapılacağı, alınan sonuçların nasıl belgeleneceğini, diğer adımlarla olan ilişkisinin ne şekilde olacağını yönetir.

2.3.6. Yazılımın Sürdürülmesi

Bu süreç, yazılımın testlerinin tamamlanıp ürünün ortaya çıkarıldığı aşamadır. Bu süreç ve daha sonraki süreçte yazılımda ortaya çıkabilecek eksikliklerin tamamlanması, kullanıcılara gerekli eğitimlerin verilmesi, sistemin bakımının yapılması planlanır ve işletilir.

2.4. Yazılım Yaşam Döngüsü Süreçleri

Yazılım yaşam döngüsü süreçlerinin yazılım projelerine dahil edilmesi, projenin sistematik bir şekilde ilerlemesini sağlar. Aynı zamanda proje yönetimini kolaylaştırır. Bu sayede ilgili projeler istenildiği şekilde başarılı bir şekilde tamamlanır. Yazılım yaşam döngüsünde her aşamada detaylı dökümantasyon ve izlenebilirlik, yazılım geliştirme süreçlerinin şeffaf olmasını sağlar. Bu sayede proje ekibi her aşamada bilgi sahibi olur.

2.4.1. Birincil Süreçler

Uluslararası Standart, bir yazılım projesinin kavramsallaştırılmasından, kullanımdan kaldırılmasına kadar uzanan yaşam döngüsünde bir veya başka bir dönemde meydana gelen bir dizi birincil süreci tanımlar. Birincil süreçler, yazılımın edinilmesi, tedariki, geliştirilmesi, işletilmesi ve bakımında yer alan kilit taraflara hizmet eder. Her birincil süreç, kendisini oluşturan faaliyetler ve görevler açısından tanımlanır. Her birincil süreç bir önsözle (bir gereklilik değil) başlar, bir dizi kurumsal düzeyde eylemle, yazılım ürünleri ve hizmetlerinin sağlanmasına yönelik bir dizi etkinlik ve ilgili görevlerle devam eder.

2.4.1.1. Satın Alma Süreci

Bu yaşam döngüsü süreci, yazılım ürününü veya hizmetini sözleşmeye bağlı olarak satın alan alıcının faaliyetlerini ve görevlerini tanımlar. Bir ürün veya hizmete ihtiyacı olan kuruluş, sahibi olabilir. Mal sahibi, satın alma görevlerinin tamamını veya bir kısmını bir acenteye devredebilir. Alıcı, kullanıcıların ihtiyaçlarını ve gereksinimlerini temsil eder. Satın alma süreci, bir yazılım ürünü veya hizmeti edinme ihtiyacının tanımlanmasıyla başlar. Süreç, teklif talebinin hazırlanması ve iletilmesi, tedarikçi seçimi ve sistemin kabulü yoluyla satın alma sürecinin yönetilmesi ile devam etmektedir. Bu süreç aşağıdaki faaliyetlerden ve bunların özel görevlerinden oluşur:
Başlatma; Teklif Talebi hazırlama; Sözleşme hazırlama ve güncelleme; Tedarikçi izleme; Kabul ve tamamlama. İlk üç faaliyet anlaşmadan önce, son ikisi ise anlaşmadan sonra gerçekleşir.

2.4.1.2. Tedarik Süreci

Bu yaşam döngüsü süreci, tedarikçinin aktivitelerini ve görevlerini içerir. Süreç, alıcının teklif talebine yanıt vermek üzere bir teklif hazırlama kararıyla ya da ediniciyle bir yazılım hizmeti sağlamak üzere bir sözleşme veya anlaşmanın imzalanması ve akdedilmesiyle başlatılabilir. Hizmet, bir yazılım ürününün veya yazılım içeren bir sistemin geliştirilmesi, bir sistemin yazılımla çalıştırılması veya bir yazılım ürününün bakımı olabilir. Süreç, hizmetin alıcıya teslimi yoluyla planların geliştirilmesi ve uygulanması da dahil olmak üzere, hizmeti yönetmek ve güvence altına almak için gereken prosedürlerin ve kaynakların tanımlanmasıyla devam eder. Tedarik süreci, özel görevleriyle birlikte aşağıdaki faaliyetlerden oluşur:

Başlatma; Yanıtın hazırlanması; Sözleşme; Planlama; Yürütme ve kontrol; İnceleme ve değerlendirme; ve Teslimat ve tamamlama. İlk iki faaliyet anlaşmadan önce, son beşi ise anlaşmadan sonra gerçekleşir.

2.4.1.3. Geliştirme Süreci

Bu yaşam döngüsü süreci, yazılım geliştiricisinin aktivitelerini ve görevlerini içerir. Geliştirme terimi hem yeni bir yazılımın geliştirilmesini hem de mevcut bir yazılımın değiştirilmesini ifade eder. Geliştirme sürecinin en az iki şekilde kullanılması amaçlanmaktadır: (1) Prototip geliştirmek için veya bir ürünün gerekliliklerini ve tasarımını incelemek için bir metodoloji olarak veya (2) Ürünleri üretmeye yönelik bir süreç olarak. Bu süreç, yazılımın tek başına bir varlık olarak veya daha büyük, toplam bir sistemin ayrılmaz bir parçası olarak geliştirilmesini sağlar. Geliştirme süreci, belirli görevleriyle birlikte aşağıdaki faaliyetlerden oluşur:
Sürecin uygulanması; Sistem gereksinimleri analizi; Sistem tasarımı; Yazılım gereksinimleri analizi; Yazılım mimari tasarımı; Yazılım detaylı tasarımı; Yazılım kodlama ve test etme; Yazılım entegrasyonu; Yazılım yeterlilik testi; Sistem entegrasyonu; Sistem yeterlilik testi; Yazılım yükleme; ve Yazılım kabul desteği. Bu faaliyetlerin konumsal sırası mutlaka bir zaman sırası anlamına gelmez. Bu aktiviteler yinelenebilir ve üst üste gelebilir veya varsayılan Şelale dizisini dengelemek için bir aktivite yinelenebilir. Bir aktivitedeki tüm görevlerin ilk veya herhangi bir yinelemede tamamlanması gerekmez, ancak bu görevlerin son yineleme sona erdiğinde tamamlanmış olması gerekir. Bu faaliyetler ve görevler, bir proje veya organizasyon için bir veya daha fazla gelişimsel model (Şelale, artımlı, evrimsel, Spiral veya bunların bir kombinasyonu gibi) oluşturmak için kullanılabilir.

Standart, ürünün geliştirilmesi sırasında önceden belirlenmiş noktalarda gereksinimlerin, tasarımın ve kodun temellendirilmesine olanak tanır ancak geliştirme sürecinin kontrolü dahilindedir. Zamanında temel belirleme, bu gereksinimlerde zamanından önce veya planlanmamış değişiklikleri engeller ve etkili değişiklik kontrolünü destekler. Standart aynı zamanda ilgili tarafların temel belirleme sürecine dahil olması için forumlar (yani ortak inceleme ve denetim süreçleri) sağlar. Geliştirme sürecinin konfigürasyon yönetimi sürecini ve temel oluşturma görevlerini yönettiğine dikkat edilmelidir.

2.4.1.4. Operasyon Süreci

Bu yaşam döngüsü süreci, bir yazılım sisteminin operatörünün aktivitelerini ve görevlerini içerir. Yazılımın çalışması toplam sistemin çalışmasına entegre edilmiştir. Süreç, yazılımın çalışmasını ve kullanıcılara operasyonel desteği kapsar. Bu süreç aşağıdaki faaliyetlerden ve bunların özel görevlerinden oluşur:
Sürecin uygulanması; Operasyonel test; Sistemin işleyişi; ve Kullanıcı desteği.

2.4.1.5. Bakım Süreci

Bakım süreci bakımcının aktivitelerini ve görevlerini içerir. Bu süreç, bir sistem; bir hata, eksiklik, sorun veya iyileştirme veya uyarlama ihtiyacı nedeniyle kodda ve ilgili belgelerde değişikliklere uğradığında etkinleştirilir. Amaç, bütünlüğünü korurken mevcut bir sistemi değiştirmektir. Bir yazılım ürününde değişiklik yapılması gerektiğinde, değişiklikleri uygun şekilde gerçekleştirmek ve tamamlamak için geliştirme süreci başlatılır. Süreç sistemin kullanımdan kaldırılmasıyla sona erer. Bu süreç aşağıdaki faaliyetlerden ve bunların özel görevlerinden oluşur:
Sürecin uygulanması; Problem ve değişiklik analizi; Değişiklik uygulaması; Bakım incelemesi/kabul; göç; ve Yazılımın kullanımdan kaldırılması.

2.4.2. Destekleyici Süreçler

Bu standart sekiz destekleyici süreçten oluşan bir dizi içerir. Destekleyici bir süreç, farklı bir amacı olan bütünleyici bir parça olarak diğer süreçleri destekler, projenin başarısına ve kalitesine katkıda bulunur. Gerektiğinde satın alma, tedarik, geliştirme, işletme veya bakım süreci ya da başka bir destekleyici süreç tarafından bir destekleyici süreç başlatılır. Destekleyici süreçler bir önsözle başlar, kurumsal düzeydeki bir dizi eylemle devam edebilir ve diğer yaşam döngüsü süreçlerini destekleyen bir dizi etkinlik ve ilgili görevlerle devam edebilir.

2.4.2.1. Dokümantasyon Süreci

Bu, bir yaşam döngüsü süreci tarafından üretilen bilgilerin kaydedilmesine yönelik bir süreçtir. Süreç; yöneticiler, mühendisler ve sistem kullanıcıları gibi ilgili herkesin ihtiyaç duyduğu belgeleri planlayan, tasarlayan, geliştiren, düzenleyen, dağıtan ve sürdüren faaliyetleri tanımlar. Dört faaliyet ve görevleri şunlardır: Sürecin uygulanması; Tasarım ve gelişim; Üretme; ve bakım.

2.4.2.2. Konfigürasyon Yönetim Süreci

Bu süreç, bir sistemdeki yazılım öğelerini tanımlamak, belirlemek ve temel oluşturmak için kullanılır; öğelerin değişikliklerini ve sürümlerini kontrol etmek; öğelerin durumunu ve değişiklik taleplerini kaydetmek ve raporlamak; öğelerin eksiksizliğini ve doğruluğunu sağlamak; ve öğelerin depolanmasını, taşınmasını ve teslimatını kontrol etmek. Bu süreç aşağıdakilerden oluşur:
Sürecin uygulanması; Yapılandırma tanımlama; Yapılandırma kontrolü; Yapılandırma durumu muhasebesi; Yapılandırma değerlendirmesi; ve Sürüm yönetimi ve teslimatı.

2.4.2.3. Kalite Güvence Süreci

Bu süreç, ürün veya hizmetlerin sözleşme gerekliliklerine ve belirlenmiş planlara uygunluğunun bağımsız ve objektif bir şekilde (edinen veya müşteri) güvence altına alınması için bir çerçeve sağlar. Tarafsız olmak gerekirse, yazılım kalite güvencesi, ürünleri geliştirmekten veya hizmetleri sağlamaktan doğrudan sorumlu kişilerden kurumsal özgürlükle sağlanır. Bu süreç aşağıdakilerden oluşur:
Sürecin uygulanması; Ürün güvencesi; Süreç güvencesi; ve Kalite sistemlerinin güvencesi.

2.4.2.4. Doğrulama süreci

Bu süreç, belirli bir faaliyetin ürün veya hizmetinin doğrulanmasına ilişkin değerlendirmeleri sağlar. Doğrulama, bir sistemin gereksinimlerinin tam ve doğru olup olmadığını ve bir faaliyetin çıktılarının daha önceki faaliyetlerde kendilerine dayatılan gereksinimleri veya koşulları yerine getirip getirmediğini belirler.

Süreç, sürecin, gereksinimlerin, tasarımın, kodun, entegrasyonun ve belgelerin doğrulanmasını kapsar. Doğrulama, bir sürece atanan değerlendirmeleri hafifletmez; tam tersine onları tamamlar.

2.4.2.5. Onaylama Süreci

Onaylama, nihai, inşa edilmiş sistemin özel kullanım amacını yerine getirip getirmediğini belirler. Onaylamanın kapsamı projenin kritikliğine bağlıdır, diğer değerlendirmelerin yerini almaz ancak onları tamamlar. Onaylama veya geçerli kılma, alıcı, tedarikçi veya bağımsız bir taraftan yapılabilir. Tedarikçi veya geliştiriciden bağımsız bir kuruluş tarafından yürütüldüklerinde bağımsız doğrulama ve onaylama (IV&V) süreci olarak adlandırılırlar.

2.4.2.6. Ortak İnceleme Süreci

Bu süreç, gözden geçiren ile incelenen arasındaki etkileşimlerin çerçevesini sağlar. Bunlar sırasıyla alıcı ve tedarikçi de olabilirler. Ortak bir incelemede, gözden geçiren kişi, bir projenin yaşam döngüsü faaliyetinin durumunu ve ürünlerini yorum (veya onay) için gözden geçiren kişiye sunar. İncelemeler hem yönetim hem de teknik düzeydedir.

2.4.2.7. Denetim süreci

Bu süreç, bir tedarikçinin ürün veya hizmetlerinin resmi, sözleşmeye dayalı olarak oluşturulmuş denetimleri için bir çerçeve sağlar. Bir denetimde denetçi, denetlenen kurumun ürünlerini ve faaliyetlerini gereksinimlere ve planlara uygunluğa vurgu yaparak değerlendirir. Tedarikçi üzerinde satın alan işletme tarafından bir denetim yapılabilir.

2.4.2.8. Sorun Çözüm Süreci

Bu süreç, sorunların çözümü için kapalı döngü sürecinin kurulmasına ve sorunların tespit edilmesiyle ortadan kaldırılmasına yönelik düzeltici önlemlerin alınması için mekanizma sağlar. Ayrıca süreç, rapor edilen sorunların nedenlerinin belirlenmesiyle analizini ve eğilimlerin tersine çevrilmesini gerektirmektedir. "Sorun" terimi uyumsuzluğu da içerir.

2.4.3. Organizasyonel Süreçler

Bu standart dört organizasyonel süreçten oluşan bir dizi içerir. Bir kuruluş, organizasyonel, kurumsal düzeyde, genellikle projelerin ötesinde veya projeler arasında işlevleri yerine getirmek için organizasyonel bir süreç kullanır. Bir organizasyonel süreç başka herhangi bir süreci de destekleyebilir. Bu süreçler diğer süreçlerin kurulmasına, kontrol edilmesine ve iyileştirilmesine yardımcı olur.

2.4.3.1. Yönetim Süreci

Bu süreç, satın alma süreci, tedarik süreci, operasyon süreci, bakım süreci veya destekleme süreci gibi bir yazılım yaşam döngüsü sürecinin yöneticisinin genel aktivitelerini ve görevlerini tanımlar. Faaliyetler şunları kapsamaktadır: Başlangıç ve kapsam tanımı; Planlama; Yürütme ve kontrol; İnceleme ve değerlendirme; ve Kapatma. Her ne kadar birincil süreçler genel olarak benzer yönetim faaliyetlerine sahip olsa da, farklı amaçları, hedefleri ve operasyon yöntemleri nedeniyle ayrıntılı düzeyde yeterince farklıdırlar. Bu nedenle, her birincil süreç, yönetim sürecinin bir örneğidir (belirli bir uygulaması).

2.4.3.2. Altyapı Süreci

Bu süreç, bir yaşam döngüsü süreci için temel altyapıyı oluşturmak ve sürdürmek için gereken faaliyetleri tanımlar. Bu süreç aşağıdaki faaliyetlere sahiptir:
sürecin uygulanması; Altyapının kurulması; ve Altyapının bakımı. Altyapı donanım, yazılım, standartlar, araçlar, teknikler ve tesisleri içerebilir.

2.4.3.3. İyileştirme Süreci

Standart, bir kuruluşun (yani satın alma, tedarik, geliştirme, işletme, bakım veya destekleyici süreç) yaşam döngüsü sürecini değerlendirmek, ölçmek, kontrol etmek ve iyileştirmek için ihtiyaç duyduğu temel, üst düzey faaliyetleri sağlar. Faaliyetler şunları kapsamaktadır: Süreç kurulumu; Süreç değerlendirmesi; ve Süreç iyileştirme. Organizasyon bu faaliyetleri organizasyonel düzeyde kurar. Yaşam döngüsü süreçlerinin projelere uygulanmasından elde edilen deneyimler, süreçleri iyileştirmek için kullanılır. Hedefler, organizasyon genelinde süreçlerin iyileştirilmesi ve yazılım teknolojilerinin geliştirilmesi, mevcut ve gelecekteki projelerin bir bütün olarak organizasyonun yararına sağlanmasıdır.

2.4.3.4. Eğitim Süreci

Bu süreç, yönetim ve teknik düzeylerde personel kaynaklarının ve becerilerinin belirlenmesi ve bu kaynakların edinilmesi veya geliştirilmesi için zamanında hazırlık yapılması için kullanılabilir. Süreç, bir eğitim planının geliştirilmesini, eğitim materyallerinin oluşturulmasını ve personele eğitimlerin zamanında verilmesini gerektirmektedir.

2.4.4. Uygulama Süreci

Normatif olan Ek A, projelere yönelik birinci düzey uyarlamanın gerçekleştirilmesi için gereken faaliyetleri içerir. Standartta uyarlama, uygulanamayan veya etkili olmayan süreç, faaliyet ve görevlerin çıkarılmasıdır. Standartta yer almayan ancak bir projeyle ilgili olan bir süreç, faaliyet veya görev, anlaşma veya sözleşmeye dahil edilebilir. Standart, standardın uygulanmasından etkilenecek tüm tarafların uyarlama kararlarına dahil edilmesini gerektirmektedir. Ancak bu sürecin kendisi uyarlanamaz.

3. ISO/IEC 15026 System and software Integrity Levels

3.1. Tanım

ISO/IEC 15026, tam adıyla “ISO/IEC 15026 System and software Integrity Levels” yazılım geliştirme aşamalarındaki güvenlik önlemlerini, güvenli bir yazılım geliştirmek için gerekli olan kıstasları belirten uluslararası bir standarttır. ISO/IEC 15026, yazılım süreçlerindeki güvenlik gereksinimlerini, yazılım güvenliği konusunda genel bir ölçek olan bütünlük seviyeleri (Integrity Levels) ile sağlar. Standart içinde yer alan bütünlük seviyeleri yazılım ile ilgili olan ve olmayan tüm bileşenleri içerir. Bu sayede yazılım süreçlerine ek olarak; ürünün bakımı ve sürdürülebilirliği, müşteri ilişkileri gibi konuları da kapsayan güvenli bir sistem geliştirmeye olanak sağlar.

ISO/IEC 15026 standardı kendi içerisinde kategorilere ayrılmıştır. Bunlar: ISO/IEC 15026-1, ISO/IEC 15026-2, ISO/IEC 15026-3 ve ISO/IEC 15026-4 şeklindedir.

1. **ISO/IEC 15026-1**

ISO/IEC 15026-1 standardı, ISO/IEC 15026 genelinde kullanılacak olan terimleri, bu terimlerin diğer terimler ile olan ilişkisini açıklar. 3 farklı versiyonu bulunmaktadır:

   a. ISO/IEC TR 15026-1:2010
   b. ISO/IEC 15026-1:2013
   c. ISO/IEC/IEEE 15026-1:2019

2. **ISO/IEC 15026-2**

ISO/IEC 15026-2 standardı, ISO/IEC 15026 genelinde kullanılacak teknik ve yöntemlerin tartışmasının yapıldığı, bunları uygulamak için gerekli olan sistem ve yapıları açıklar. 2 farklı versiyonu bulunmaktadır:

a. ISO/IEC 15026-2:2011
b. ISO/IEC/IEEE 15026-2:2022

3. **ISO/IEC 15026-3**
   ISO/IEC 15026-3 standardı, ISO/IEC 15026 sistem geliştirme süreçlerinde kullanılacak olan bütünlük seviyelerinin tanımlarını, bu seviyelerin içeriklerini, seviyelerin birbirlerinden ayrıldıkları noktaları, bu seviyeleri uygulamak için gerekli olan sistem ve yapı gereksinimlerini açıklar. 4 farklı versiyonu bulunmaktadır:
   a. ISO/IEC 15026:1998
   b. ISO/IEC 15026-3:2011
   c. ISO/IEC 15026-3:2015
   d. ISO/IEC/IEEE 15026-3:2023

4. **ISO/IEC 15026-4**
   ISO/IEC 15026-4, yazılım ve sistem geliştirme süreçlerinde kullanılacak olan yapıların, eylemlerin, görevlerin açıklanması, bunların standart dahilinde nasıl gerçekleştirileceği ile ilgili öneriler ve değerlendirmeler sağlar. 2 farklı versiyonu bulunmaktadır:
   a. ISO/IEC 15026-4:2012
   b. ISO/IEC/IEEE 15026-4:2021

3.2. Kapsam ve Amaç

ISO/IEC 15026 Uluslararası Standardı, yazılım bütünlük seviyelerini ve yazılım bütünlük gereksinimlerini belirleme süreçlerini tanımlayan ve oluşturduğu her sürece gereksinimler koyan, yazılım için özelleşmiş bir standarttır. Yazılım ürünleri veya sistemlerinin geliştirilmesinde kullanılmak üzere tasarlanmıştır. Bu uluslararası standart, belirli bir bütünlük seviyesi veya yazılım bütünlük gereksinimleri kümesini belirlemez. Bunlar, ya proje bazında ya da belirli bir sektör veya ülke için belirlenmelidir. Sistem bütünlük seviyesi ve yazılım olmayan bileşenlerin bütünlük seviyeleri, yalnızca yazılım bileşenlerinin bütünlük seviyelerini belirlemek için bu standartta gerekli olanlardır. Bu standart, oluşturulan ürünlerin ve sistemlerin idari ve teknik destek gereksinimlerini kapsar. Bir yazılım bütünlük seviyesi, sistem risklerini tolere edilebilir sınırlar içinde tutmak için gerekli olan bir yazılım özelliğinin değer aralığını belirtir. Bir azaltma işlevi gerçekleştiren yazılım için, özellik, yazılımın azaltma işlevini ne kadar güvenilir bir şekilde gerçekleştirmesi gerektiğidir.

Bir sistem tehdidine yol açabilecek yazılım hatasının neden olduğu durumda, özellik, hatanın sıklığı veya olasılığı üzerindeki sınırdır. Yazılım bütünlük gereksinimleri, yazılım geliştirme sürecinde kullanılan yazılım mühendisliği süreçlerinin karşılaması gereken gereksinimlerdir;

bunlar aynı zamanda yazılım mühendislik ürünlerinin de karşılaması gereken gereksinimlerdir; veya zaman içinde yazılımın performansıyla ilgili olarak doğru olması gereken gereksinimlerdir, bu da yazılımın bütünlük seviyesi ile orantılı bir güven düzeyi sağlamak amacıyladır. Bu uluslararası standart, bütünlük seviyesi belirlemenin, genel sistem mühendislik yaşam döngü süreçleri ile nasıl entegre edildiğini belirlemez.

3.3. Kapsam dahilindeki sistem ve yazılımların tanımı

Bu bölümde, sistem ve yazılım bütünlük seviyelerine odaklanan ISO/IEC 15026 standardı için kapsamın içinde bulunan sistem ve yazılımların, standardın kapsamına dahil olma durumları tanımlanır.

Standardın Gerçek Hayat Kullanım Örnekleri:

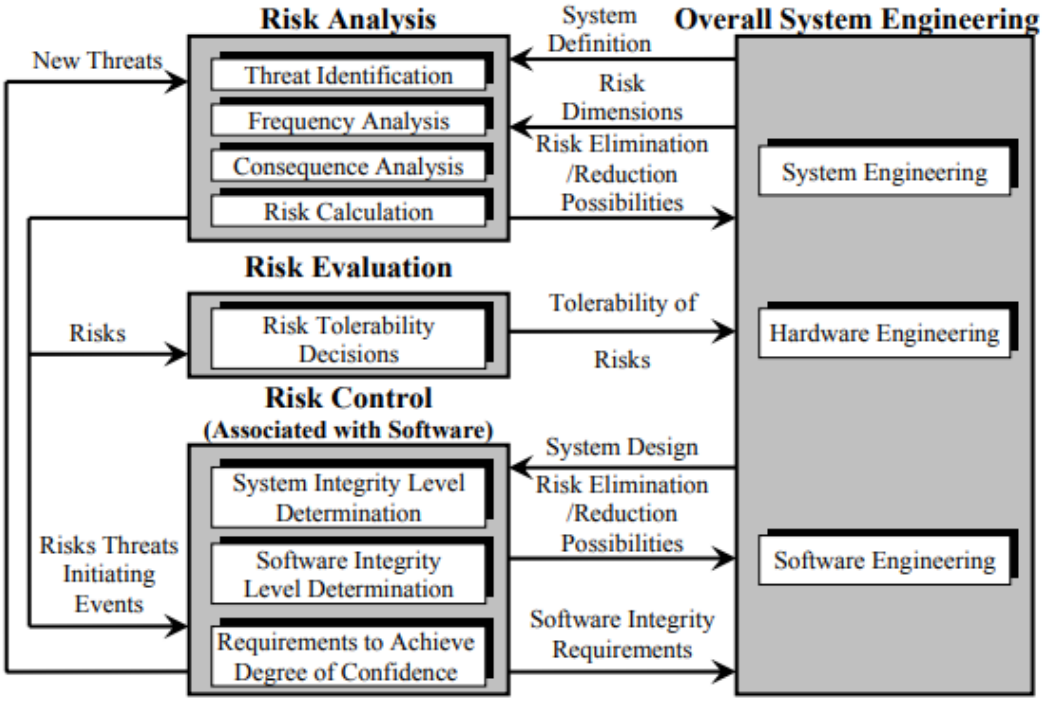
Tıbbi Cihazlar ve Sağlık Yazılımları: ISO/IEC 15026 standardı, tıbbi cihazlar ve sağlık sektöründe kullanılan yazılımların bütünlüklerini belirlemek ve sağlık uygulamalarında veri güvenliğini sağlamak için kullanılabilir. Örneğin; bir hastane bilgi yönetim sistemi, hasta bilgilerinin güvenliği ve doğruluğu için bu standartlara uyumlu olmalıdır. Aynı zamanda tıbbi cihazların güvenilirliğini arttırarak hastaların sağlığına zarar verme riskini azaltmaktadır
Finansal Teknoloji Uygulamaları: ISO/IEC 15026 standardının finansal teknoloji alanındaki uygulamalarda kullanılma amacı uygulamalar üzerinde bilgi güvenilirliğini üst seviyeye taşımak ve bütünlüğünü değerlendirmektir. Örneğin; online ödeme sistemleri, blokchain teknolojisi yada finansal veri işleme yazılımlarında, yazılımların bütünlük seviyelerini belirleme işlemi bu standartlarla sağlanabilir Bu durum finansal verilerin güvenliği ve işlemlerin doğruluğu açısından önemlidir.
Endüstriyel: ISO/IEC 15026 standardının endüstriyel yazılımlarda kullanımı, ürün ve hizmetlerin geliştirilmesi sürecinde sistemlerin bütünlük seviyelerini belirlemek ve güvence sağlamak için önemlidir. Endüstri gibi karmaşık ve çok yönlü sistemlerin geliştirildiği savunma sanayi, havacılık, otomotiv gibi alanlarda bu standart sıklıkla kullanılmaktadır.
Otomotiv Sektörü: Özellikle otonom sürüşün günümüz otomobillerinde yaygınlaşmasıyla, var olan yazılım sistemlerinde risk ve kontrol sistemlerinin önemi artmıştır. Standart, geliştirme sürecindeki riskleri belirlemeye, değerlendirmeye ve riskleri azaltacak yöntemler geliştirmeye yardımcı olur. ISO/IEC 15026'nın uygulanmasındaki amaç; otomotiv endüstrisindeki şirketlerin, mühendislerin, araçların yüksek güvenliği ve güvenilirlik standartlarının karşılanmasını sağlamaktır.
Uzay Sistemleri: Hatanın geri dönüşü olmadığı, yüksek miktarda para ve zaman yatırımı yapılan uzay sistemlerinde; ISO/IEC 15026 standardı risk analizi için kullanılmaktadır. Özellikle uydu kontrol sistemleri, uzay araçları ve karmaşık uzay haberleşme sistemlerinde yaygın olarak bu standardın kullanımı vardır.
Telekomünikasyon Sistemleri: ISO/IEC 15026-4, güvenlik kritik olan telekomünikasyon sistemlerinin geliştirilmesi ve değerlendirilmesi için de kullanılabilir. Örneğin, kritik altyapıyı destekleyen ağ sistemleri.

3.4. Bütünlük Seviyeleri(Integrity Levels):

Bütünlük seviyeleri, belirli risk seviyeleri için veya bir güvence durumunu desteklemek ve uygulamak için kullanıma uygundur. Paydaşlar arasında anlaşmaya varmak için kullanılan güven derecesinin bir temsili olarak görülebilir. ISO / IEC 15026-3 önce bir bütünlük seviyesi belirlemeye yardımcı olur. Standardın geri kalanı bütünlük seviyelerini kullanma, risk analizi yaparak sistem veya ürün bütünlük düzeyi belirleme, sistem ögelerine bütünlük seviyesi ataması yapma, kanıtları kullanarak bütünlük seviyelerinin gereksinimlerini karşılama ve yetkililerle yapılan anlaşmaları içerir.

Bütünlük düzeyi gereksinimleri, bir sistemin veya sistem öğesinin iddia edilen özelliklere sahip olduğunu göstermek amacıyla yapılması gerekenleri içerir. Bir sistemin bütünlük düzeyi tüm sistemin özellikleri açısından neyin yeterli olacağını belirtir. Böylece, özellikleri gösteren sistemi ve çevresini içeren daha büyük taleplerin karşılanmasını göstermede temel bir role sahiptir, arzu edilen veya istenmeyen sonuçları inceler.

Uygulamada, dürüstlük düzeyleri sıklıkla, gereksinimlerin karşılanması için gereken kanıtların vurgulanması açısından tartışılmaktadır. Sistem bütünlük seviyesi gerekliliklerini yerine getirmelidir ve böylece iddiaları destekleyen sistemin kendi özellikleriyle ilgili argümanlar için kanıt sağlar. Bütünlük seviyesi gereklilikleri, ilgili bütünlük seviyesine ulaşıldığını göstermesi açısından da, bu kalitenin belirsizlikler üzerindeki etkisi nedeniyle önemlidir. Argüman, kanıt ve varsayımla ilgili belirsizlikler bütünlük seviyesi gerekliliklerini oluşturmanın bir parçasıdır.



Şekil 1. Yazılım Bütünlüğü Düzeyi Belirleme Sürecine Genel Bakış

3.4.1. Farklı bütünlük seviyelerinin açıklamaları

Bütünlük Seviyesi 1 (IL1):
Açıklama: Bu seviye, düşük riskli sistemleri kapsar.
Gereksinim ve Karakteristikler: Temel güvenlik önlemleri, basit yapılar, düşük karmaşıklık.

Bütünlük Seviyesi 2 (IL2):
Açıklama: Orta düzeyde riskli sistemleri kapsar.
Gereksinim ve Karakteristikler: Temel güvenlik önlemleri, daha karmaşık yapılar, daha fazla güvenlik gereksinimi.

Bütünlük Seviyesi 3 (IL3):
Açıklama: Yüksek riskli sistemleri kapsar.
Gereksinim ve Karakteristikler: Karmaşık yapılar, güçlü güvenlik önlemleri, detaylı risk analizi.

Bütünlük Seviyesi 4 (IL4):
Açıklama: Çok yüksek riskli sistemleri kapsar.
Gereksinim ve Karakteristikler: En karmaşık yapılar, en yüksek düzeyde güvenlik önlemleri, detaylı ve özel risk analizi.

3.4.2. Bütünlük Seviyelerinin Uygulama Alanları

Havacılık (Uzay) Endüstrisi:
Uygulama: Uçuş kontrol sistemleri, aviyonik sistemler.
Bütünlük Seviyeleri: Yüksek bütünlük seviyeleri, uçuş sistemlerinin güvenliği için kritiktir.

Tıbbi Cihazlar:
Uygulama: Hayati önem taşıyan sistemler, tanı ekipmanları.
Bütünlük Seviyeleri: Kritik, çünkü tıbbi cihazlardaki hatalar hayati tehlikelere yol açabilir.

Nükleer Enerji Sistemleri:
Uygulama: Nükleer reaktör kontrol sistemleri.
Bütünlük Seviyeleri: Çok yüksek, nükleer enerji ile ilişkilendirilen potansiyel riskler göz önüne alındığında.

Otomotiv Endüstrisi:
Uygulama: Gelişmiş sürücü destek sistemleri (ADAS), otonom araçlar.
Bütünlük Seviyeleri: Yüksek bütünlük, otomatik sürüş sistemlerinin güvenliği için hayati önemlidir.

Telekomünikasyon:
Uygulama: Ağ altyapısı, iletişim protokolleri.

Bütünlük Seviyeleri: Kritik, güvenilir ve güvenli iletişimi sağlamak için.

Savunma ve Askeri:
Uygulama: Komuta kontrol sistemleri, silah sistemleri.
Bütünlük Seviyeleri: Çok yüksek, savunma sistemlerinin güvenilirliği temel önemdedir.

Finansal Sistemler:
Uygulama: Bankacılık sistemleri, finansal işlem platformları.
Bütünlük Seviyeleri: Yüksek, finansal işlemlerin doğruluğu ve güvenliği için.

Endüstriyel Otomasyon:
Uygulama: Proses kontrol sistemleri, üretim otomasyonu.
Bütünlük Seviyeleri: Kritik, çünkü hatalar üretim süreçlerini ve işçi güvenliğini etkileyebilir.

Uzay Sistemleri:
Uygulama: Uydu kontrol sistemleri, uzay keşif ekipmanları.
Bütünlük Seviyeleri: Çok yüksek, uzayın zorlu koşulları ve uzay misyonlarının karmaşıklığı göz önüne alındığında.

Sağlık Bilişimi:
Uygulama: Elektronik sağlık kayıtları, sağlık bilgi sistemleri.
Bütünlük Seviyeleri: Yüksek, hasta verilerinin bütünlüğünü ve gizliliğini sürdürmek için.

3.4.3. Risk analizi

Risk analizi, tüm sistem için gerekli bütünlük düzeyini belirler. Risk analizi devam eden ve henüz bilinemeyen ile bilinmesi gerekeni dengelemesi gereken yinelenen bir süreçtir. Risk analizinden ortaya çıkarılan bütünlük seviyeleri, sonuçların değerlerinin sistem koşullarının veya davranışlarının oluşumları ve zamanlamalarının risk analizine çevrilmesidir. ISO/IEC/IEEE 15026 (tüm bölümler), risk analizini ayrıntılı olarak kapsamaz. Risk analizi için rehberlik sağlayan yönergeler sunan ve potansiyel risklerin belirlenmesine yardımcı olabilecek birçok standart ve belgeler mevcuttur. IEC 61508 ve IEC 31010, risk analizine yönelik yaklaşımlar sağlar. Güvenlik odaklı olarak IEC 31010 terminolojisi kullanıldığı için “tehlike” ve “zarar” terimleri sırasıyla“tehlikeli durum” ve “olumsuz sonuç” olarak yorumlanmalıdır. Bu sonuca IEC 60300 ayrıca rehberlik sağlar. Diğer özel standartlar arasında makinelere ilişkin ISO 13849, uzay sistemlerine ilişkin ISO 14620, yangın durumları için ISO 19706 yer almaktadır. Sağlık bilişimi üzerine ISO/TS 25238, bilgi güvenliği üzerine ISO/IEC 27005 ve UK CAP 760 hava trafiği ve havaalanları için uygulanmaktadır.
Ayrıca daha genel risk yönetimi standartları ISO/IEC 16085 ve ISO 31000 da gözden geçirilmelidir.

3.4.4. Risk değerlendirilmesi ve aşamaları ve açıklamaları

Risk Değerlendirmesi:
ISO/IEC/IEEE 15026 standardı, risk değerlendirmesi ve bütünlük seviyelerinin belirlenmesi süreçlerini içerir. Bu standart doğrultusunda gerçekleştirilen risk değerlendirmesi sürecinin temel adımları:

Risk Analizi:
İlk adım, sistemin karşılaşabileceği potansiyel riskleri tespit etmek ve anlamak için kapsamlı bir risk analizi yapmaktır. Bu, sistemin performansını, güvenilirliğini ve bütünlüğünü etkileyebilecek olası tehditleri ve zayıflıkları değerlendirmeyi içerir.

Risk Kategorilerinin Belirlenmesi:
Belirlenen riskler genelde belirli kategorilere ayrılır. Bu adım, farklı risk türlerini ve etkilerini gruplandırmak amacıyla risk kategorilerini tespit etmeyi hedefler.

Risk Derecelendirme:
Tespit edilen risklerin, şiddeti ve olasılığı derecelendirilir. Bu, risklerin etkisini ve olasılığını nicel veya nitel bir doğrultuda değerlendirmeyi içerir.

Risk Kabul Kriterlerinin Belirlenmesi:
Risk kabul kriterleri, belirli bir risk düzeyinin kabul edilebilir olup olmadığını ayırt etmek için kullanılır. Bu kriterler, projenin hedeflerine, bütünlük seviyelerine ve güvenlik gereksinimlerine dayanabilir.

Bütünlük Seviyelerinin Belirlenmesi:
Risk analizi sonuçlarından yola çıkarak, belirli bütünlük seviyeleri belirlenir. Bu seviyeler, sistemin güvenilirliği ve bütünlüğünü sağlamak için gerekli olan minimum gereksinimleri sembolize eder.

Risk Azaltma Stratejilerinin Geliştirilmesi:
Belirlenen risklere yönelik azaltma veya kontrol stratejileri ortaya konulur.. Bu stratejiler, risklerin kabul edilebilir düzeylere indirilmesini veya yönetilmesini sağlamak amacıyla oluşturulur.

Risk İzleme ve Değerlendirme:
Projenin ilerlemesi boyunca risklerin takibi ve değerlendirilmesi devam eden bir süreçtir. Yeni risklerin ortaya çıkması veya mevcut risk durumlarının değişmesi durumunda güncelleme yapılır.

4. Kaynakça

Kaynak 1:
https://www.iso.org/standard/21208.html

Kaynak 2:
https://medium.com/@jilvanpinheiro/software-development-life-cycle-sdlc-phases-40d46afbe384

Kaynak 3:
Al-Qutaish, R. E., & Al-Sarayreh, K. (2008). Software Process and Product ISO Standards: A Comprehensive Survey. European Journal of Scientific Research, 19(2), 289-303.

Kaynak 4:
Tripp, L. L. (1996). International Standards on System and Software Integrity. StandardView, Vol. 4, No. 3, 146-150.

Kaynak 5:
Aydan, U., Yilmaz, M., Clarke, P. M., & O'Connor, R. V. (2017). Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach. Computer Standards & Interfaces, 54(3), 129-138.

Kaynak 6:
Joannou, P., & Wassyng, A. (November 2014). "Understanding Integrity Level Concepts" Computer Standards & Interfaces, 54(3), 99-101.

Kaynak 7:
Takai, T., & Takamura, H. (Yıl). "IFIP WG10.4 Dependability Standards and our Challenge to Establish a New Standard for Open Systems." AIST, JAPAN.

Kaynak 8:
ISO/IEC/IEEE. (2019). ISO/IEC/IEEE 15026-1:2019(E) - Systems and software engineering — Systems and software assurance — Part 1: Concepts and vocabulary. Geneva, Switzerland: ISO/IEC.