

15.3. ISO/IEC ve IEEE/EIA Standart 10207

IEEE/EIA 10207, yazılım süreç yönetimiyle ilgili IEEE/EIA(Institute of Electrical and Electronics Engineers/Electronic Industries Alliance) tarafından geliştirilen bir üst standarttır. ISO'nun (International Organization for Standardization) yazılım süreçlerinin yönetimiyle ilgili önerdiği standartlar, IEEE/EIA tarafından ilave ve açıklamalarla geliştirilerek 1995 yılında yayınlanmıştır. Diğer IEEE yazılım standartları 10207'ye bağlı olarak geliştirilir. Bu standart iyi tanımlanmış bir terminoloji kullanarak yazılım yaşam döngüsü için bir altyapı oluşturur. Bu sayede yazılım dünyasından referans edilebilir. Kendi içerisinde üç alt kısma sahiptir.

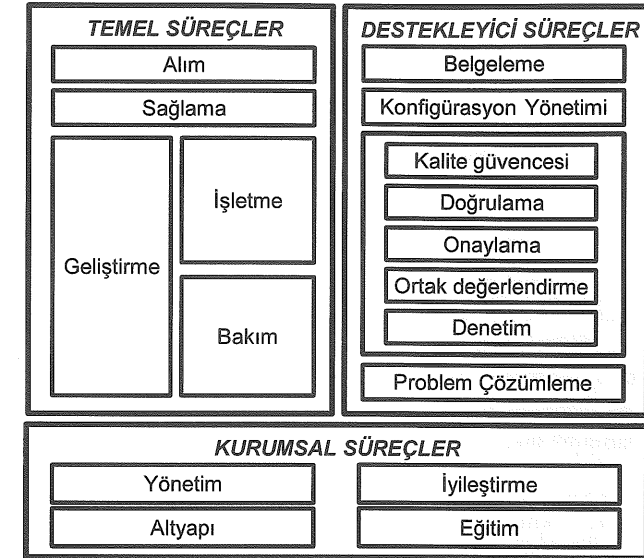
- **10207.0** (Bilgi teknoloji standartları, Yazılım Yaşam Döngüsü Süreçleri) Standardı açıklayan temel belgedir [IEEE-10207.0]. ISO/IEC 12207 standardının orijinal formunu içerir ve ayrıca altı ek bulunur: Temel kavramlar, uyumluluk, yaşam döngüsü süreç hedefleri, ilişkiler ve yanlış doğru cetveli, standardın farklı süreçlere nasıl uyarlanabileceğiyle ilgili ek açıklamalar.
- **10207.1** (Bilgi teknoloji standartları, Yazılım yaşam Döngüsü Süreçleri, Yaşam Döngüsü Verisi) Yaşam döngü verileri için ek kılavuzlar içerir[IEEE-10207.1].
- **10207.2** (Bilgi teknoloji standartları, Yazılım yaşam Döngüsü Süreçleri, Uygulamada dikkat edilecekler) standardın iş hayatında nasıl uygulanabileceğine dair bilgiler ve en iyi uygulama örnekleri içerir [IEEE-10207.2]⁶.

IEEE/EIA 10207, kapsamlı bir standarttır. Kurum, hedeflerine göre sadece kendilerine uygun alt kısımları seçebilir. Bu standart öncelikle kurum, proje ve uygulamalara uyarlanmak üzere tasarlanmıştır. Ancak tek başına veya büyük bir sistemin parçasını oluşturan gömülü yazılımlarda da kullanılabilir. Standardın projeye nasıl uyarlanabileceği 10207 içinde özellikle gösterilmiştir. 10207 standardı yazılım yaşam döngüsündeki görev ve işlemlerin “nasıl olması gerektiğini” değil “ne olması gerektiğini” gösterir! Bu standart aşağıdaki süreçlerde kullanılabilir:

1. Yazılım alma, sağlama, geliştirme, yürütme ve bakım
2. Birinci maddedeki fonksiyonları kalite güvence, konfigürasyon yönetim ve ortak gözden geçirme gibi faaliyetlerle desteklemek
3. Kurumun süreç ve insan kaynağını yönetmek ve geliştirmek
4. İş hedeflerine hizmet etmek için düzenlenmiş ve uyarlanmış, yaşam döngüsü süreçlerine dayanan, yazılım yönetim ve mühendislik ortamı kurmak
5. Yazılım yaşam döngüsünde yer alan müşteri, sağlayıcı ve diğer kurumlar arasında karşılıklı anlayışın gelişmesini teşvik etmek
6. Yazılımda milletlerarası ticareti kolaylaştırmak

⁶ Konu açıklanırken bu kaynaktan birçok noktada faydalanılmıştır.

10207 standardında süreçler, temel, destekleyici ve kurumsal olmak üzere üç kategori altında incelenir; bkz. Şekil 15.10.



Şekil 15.10. Milletlerarası IEEE/EIA 10207 Standardı yapısı

15.3.1. Temel Yaşam Döngüsü Süreçleri

Temel süreçler her firmada var olması ve yürütülmesi gerekli işlemleri gösterir. Standart içinde her sürecin sorumlusu belirtilmektedir.

1. Alım

Alım, satın alan kurumun faaliyetlerini tanımlar. Alım süreci müşteri tarafından bir sistem, yazılım ürünü veya yazılım servisi kapsam ve ihtiyacının tanımlanmasıyla başlar. Sistem gereksinimleri ve alım alternatifleri ayrıntılı analiz edilir. Süreç, şartname hazırlığı, sözleşme, kontrolle devam eder, kabul ve tamamlamayla sonlanır.

Alıcı ve sağlayıcı taraflar arasında uzlaşma sağlanmalıdır. Ayrıca son kullanıcıların iyileştirmenin kapsamını anlaması ve uyum göstermesi gerekir. Alıcı, alım planı içinde yazılım geliştirme faaliyetlerinin ölçülmesi için gerekli programı oluşturmaktadır. Doğrudan satın alma, yeni geliştirme veya mevcuda ilave yapma alım şekilleridir.

Alım şartları şartnamede toplanır. Şartname ihtiyaçlar, kapsam, isteklilere yapılan açıklamalar, yazılım ürünleri listesi, terimler ve şartlar, altyüklenici kontrol yöntemi ve teknik kısıtlar hakkında bilgiler içermelidir.

Seçim kriterleri belirlendikten sonra seçilen firmayla sözleşme yapılır. Sözleşmede doğrudan kullanıma hazır yazılım ürünleriyle ilgili mülkiyet, kullanım, garanti ve lisans hakları yer almalıdır.

Kontrol süreci ortak gözden geçirme ve denetime dayanır. Buna ilave olarak *satıcı*, onaylama ve doğrulama süreçlerini alıcının izleyebilmesini sağlamalıdır. *Satıcı*, gerekli tüm bilgileri zamanında vermekten sorumludur.

Alıcı, kabul stratejisi ve kriterlerine göre kabul şartlarını hazırlamalıdır. Tüm şartların sağlandığı, satıcıyla birlikte gözden geçirilir. Alıcının, satıcının hazırladığı kılavuzları kullanarak, yazılımı konfigürasyon yönetimine eklemesi ve hedef ortama kurması öngörülmüştür. Alıcının ürünü kendi kurması önemlidir, çünkü eğer destek anlaşması yoksa sonraki kurulumları kendisi yapacaktır. Bu aşamada satıcıdan da destek alınabilir.

2. Sağlama-Temin

Sağlama, satın alana hizmet sağlayan kurumun, yani sağlayıcının faaliyetlerini tanımlar. Hazırlık, cevap hazırlama, sözleşme, planlama, uygulama ve kontrol, gözden geçirme ve değerlendirme, teslimat ve tamamlama aşamalarından oluşur. *ISO 9001 standardında* satın alma ve satış süreçleri ikiye ayrılmış, her ikisi için de ayrı planlar hazırlaması önerilmiştir. Sağlayıcı için çok daha ayrıntılı bir plan önerilmiş, alıcı için ise planlama faaliyetleri kontrol ağırlıklı bırakılmıştır.

Sağlayıcı, alıcının şartnamesindeki gereksinimleri kendi kurumsal politikalarına göre değerlendirir. Teklif verme veya sözleşme yapma konusunda bir karar verir. Alıcıyla görüşerek şartnamenin bazı şartlarında değişiklik isteyebilir. Yapılacakları belirleyen bir sözleşme imzalandıktan sonra sağlayıcı ayrıntılı bir plan hazırlar. Sağlayıcının hazırladığı genel yazılım geliştirme planında, alıcının sorumlulukları da kurumsal yapıyı ve gözden geçirme şekillerini içerecek şekilde ayrıntılı belirtilmelidir.

Sağlayıcının diğer kontrol faaliyetlerine ek olarak alıcıya değerlendirme, gözden geçirme, denetleme, test ve problem çözümü raporları sağlaması önerilir. Planda belirtilen gözden geçirmeler için satıcının bazı imkânlarından alıcının da faydalanması sağlanmalıdır. Alt yüklenici kullanılıyorsa herbiri için sadece kendi yapacakları çalışmaları içeren ayrı bir plan hazırlanır. Ana ve alt yüklenici arasında da alıcı ve sağlayıcı ilişkisi vardır. Sağlayıcı, sözleşmede belirtilen şartlara uygun yazılım ürün veya hizmetini teslim etmelidir. Sağlayıcı destek konusunda alıcıya yardım edebilir.

3. Geliştirme

Yazılım ürününü tanımlayan (tasarlayan) ve geliştiren kurumun faaliyetlerini tanımlar. Yazılım geliştirme sürecinin izlenmesi için 13 faaliyet önerir:

- | | |
|-------------------------------|--|
| 1. Süreç gerçekleştirme | 8. Yazılım bütünleştirme |
| 2. Sistem ihtiyaç analizi | 9. Yazılım yeterlilik (<i>qualification</i>) testi |
| 3. Sistem mimari tasarımı | 10. Sistem entegrasyonu |
| 4. Yazılım ihtiyaç analizi | 11. Sistem yeterlilik testi |
| 5. Yazılım mimari tasarımı | 12. Yazılım kurulumu |
| 6. Yazılım ayrıntılı tasarımı | 13. Yazılım kabul desteği |
| 7. Yazılım kodlama ve test | |

Sistem donanım, yazılım ve elle yürütülen işleri kapsar. *Analiz* sistemden başlayarak yazılım ihtiyaçlarına doğru ayrıntılı hale getirilir. Sistemin tümüyle ilgili ihtiyaçlar çok hızlı değiştiğinden standart içinde kısaca bahsedilmiştir. Kalite karakteristiklerini içeren bir ihtiyaç analizi yapılmalıdır. Her karakteristik kendisiyle ilgili tarafsız testler tanımlanabilecek şekilde ifade edilmelidir. Kalite karakteristiklerinin içerdiği konular (*ISO/IEC 9126* standardı yardımıyla):

- Fonksiyonel ve kapasite özellikleri: Performans, fiziksel karakteristikler, yazılımın altında çalıştığı çevre şartları gibi
- Yazılım bileşenlerine yönelik harici ara yüzlerin kalitesi
- Yeterlilik şartları
- İş emniyeti yönetmelikleri: Yürütme ve bakım ile ilgili faaliyet, çevresel etki ve personel yaralanmalarını önlemeyi içerir
- Hassas bilgilerle ilgili uzlaşmaları içeren güvenlik yönetmelikleri
- İnsan faktörü mühendisliği (*ergonomiler*): Elle yürütülen işlemler, insan ekipman etkileşimi, personel üzerindeki kısıtlar, kişilerin hata ve eğitimleriyle ilgili yoğun insan dikkati gerektiren alanlar
- Veri tanımları ve veritabanı gereksinimleri
- Teslim edilen yazılımın çalışma ve bakım ortamına, kurulum ve kabulü
- Kullanıcı belgeleri
- Kullanıcı çalışma, işleri yürütme ve bakım ihtiyaçları

Sürüm temelli (yinelemeli) geliştirme için genel bir plan ve her sürüm için ayrıntılı birer plan olması önerilir. Geleceğe yönelik planın mevcut bilgilerin elverdiği ayrıntıda yapılabileceği hatırlatılır.

Yazılım ekibinin satın alana karşı sorumlulukları üzerinde hassasiyetle durulur. *Geliştirici*, kullanıcıyı etkileyen her konuda kullanıcının görüşünü almalıdır. Bunu sağlamak için kullanıcı belgesinin ön sürümü, sistem yazılım tasarımıyla birlikte hazırlanır. Geliştirici, alıcının kabul gözden geçirmeleri yapmasını ve yazılımı test etmesini sağlamalı, yapılan sözleşmede belirtildiği şekilde yazılımı tamamlamalıdır. Kullanım başladığında ve sonrasında, sürekli eğitim ve bakım sağlamalıdır.

4. İşletme

İşletme, bilgisayar sistemine çalışma ortamında işletme hizmeti veren *operatör* kurumun faaliyetlerini tanımlar. Yazılımın işlemesi için bazı rutin işlerin yapılma ve son kullanıcıya işletme desteği sağlanmasını kapsar. Yazılımın işletilmesi altyapıdaki sistemle bütünleşik yapıldığından, *sistem düzeyindeki* bir faaliyet olarak kabul edilir. *İşletme* son kullanıcı için program çalıştıran operatörler kullanan sistemler için geçerli bir hizmettir. Sistemin bazı özel ayarlarını yapan yönetici, rapor ve diğer çıktıları üreten operatör *işletme hizmetine* örnektir. Son kullanıcıya destek ve uzmanlık sağlamak da operatörün bir işidir. *PMI* ve *CMMI* doğrudan işletme sürecini adreslemez.

İşletme faaliyetleri planlanmalı ve plana göre yürütülmelidir. Özellikle sürüm geçişlerinde tüm işletme faaliyetlerinin tekrar test edilmesi önerilir. Çıkan sorunlar bakım sürecine aktarılır.

5. Bakım:

Bakım, yazılımı güncel ve işleme uygun tutma, başka bir ortama taşıma ve devreden çıkarma gibi hizmetleri veren kurumun faaliyetlerini tanımlar. Hata veya iyileştirme sebebiyle yazılımın kodu ve ilgili belgelerde değişiklik yapılmasıyla bakım süreci aktifleşir. Temel hedef, mevcut yazılımı bütünlüğünü koruyarak değiştirmektir.

Tüm belgeler güncelse, bakım çalışmalarında hangi belgenin nasıl değişeceğini belirlenir. Ancak birçok projede bilgiler güncel olmadığından, önce ne kadar yeni bilgi oluşturulacağı sonra da mevcutta değişecek kısımlar belirlenmelidir. Bakım sürecinde çıkan değişiklikleri gerçekleştirmek için *geliştirme* sürecine girilir. Değişiklik için gerekli test planını sağlama ve yapılanları test etmek, bakım sorumlusunun görevidir.

15.3.2. Destekleyici Süreçler

Destekleyici süreç, diğer (temel) bir süreçle özel bir amaçla bütünleşir ve onu destekleyen bir parça olur. Böylece yazılım projesinin başarısına ve kalitesine katkı sağlar.

1. Belgeleme

Belgeleme yaşam döngüsü sürecinde, üretilen bilgilerin nasıl saklanacağını tanımlar. *Belgeleme süreci* tüm tarafların ihtiyaç duyduğu belgelerin; planlama, tasarım, geliştirme, üretme, değiştirme, dağıtım ve bakım aşamalarından oluşur. Yazılım yaşam döngüsünde oluşan belgeleri tanımlayan bir plan geliştirilmeli ve yürütülmelidir. Planda belgeyle ilgili başlık, tür, amaç, hedef kullanıcılar, belge üzerindeki işlemlerle ilgili yönetmelik ve sorumluluklar, ara ve son sürümlerin zamanlaması bulunmalıdır.

Belgeler, belgeleme standartlarına uygun olarak tasarlanır. Giriş verisinin kaynağı ve uygunluğu doğrulanmalıdır. Bilgileri belgelemek için otomatik belge hazırlayan araçlar kullanılabilir. Yapılan çalışmalar, yetkili çalışanlar tarafından gözden geçirilir.

Belgeler plana uygun şekilde üretilir veya sağlanır. *Belge saklanırken* kayıt yaşam döngüsüne (*retention*), güvenlik, bakım ve yedekleme gereksinimlerine uygun hareket edilmelidir. Belge güncellenme, konfigürasyon yönetim süreciyle yönetilir.

2. Konfigürasyon Yönetimi

Konfigürasyon Yönetimi aşağıdaki amaçları sağlamak için yazılım yaşam döngüsü üzerine yönetici ve teknik yöntemler uygulamaktır.

- Yazılım bileşenlerini saptamak, tanımlamak ve taban hattı oluşturmak,
- Bileşenlerin değişimini ve sürümleri kontrol etmek
- Bileşenin ve değişim isteğinin durumunu, kayıt altına almak ve raporlamak
- Bileşenlerin bütünlüğü, tutarlılığı ve doğruluğu güvenceye almak
- Bileşenlerin depolanmasını, işlemlerini ve teslimatını kontrol etmek

Konfigürasyon yönetimini gerçekleştirmek için her yazılım bileşenine tekil bir tanımlayıcı atanmalı ve tüm bileşenleri içeren bir şema hazırlanmalıdır.

Değişim yönetimi, 10207 standardında konfigürasyon yönetimi konfigürasyon kontrol başlığında ele alınmıştır. *Buradaki işlemler* değişik isteklerini tespit ve kaydetme, analiz ve değerlendirme, onay veya ret, gerçekleştirme, doğrulama ve değişen bileşenleri devreye alma şeklindedir. Değişimleri sebep ve yetki açısından izleyebilmek amacıyla her değişim için bir denetleme yöntemi belirlenmelidir. Konfigürasyonunun tutarlılığı hem ihtiyaçlarla yapılanlar arasındaki tutarlılık, hem de yazılımın bileşenlerinin fiziksel bütünlüğü açısından değerlendirilmelidir.

3. Kalite güvencesi

Kalite güvencesi proje yaşam döngüsü içerisindeki yazılım süreç ve ürünlerinin, belirlenen özel şartlara uygun geliştirildiğini ve oluşturulan plana bağlı kaldıklarını, güvence altına alma sürecidir. Sürecin tarafsız olabilmesi için kalite güvencesi faaliyetini yürüten kişiler, doğrudan yazılımı geliştirilen veya sürecini yürüten kişilerden kurumsal bağımsızlığa ve yetkiye sahip olmalıdır. Bu kişiler sorunların çözüm yöntemlerini belirleme ve uygulanmasını sağlamaktan sorumludur. *Kalite güvence süreci* süreç ve ürün kalite delillerinin, sağlayıcı veya alıcının yönetim kademelerine gösterilip gösterilmeyeceğine göre dâhili veya harici olarak yürütülebilir.

Kalite güvence faaliyetleri bir plana göre yürütülmelidir. Bu planda kalite güvence standart yöntem ve yönetmelikleri, diğer paydaşlarla birlikte yapılacak gözden geçirme ve koordinasyon faaliyetleri, kaliteyle ilgili kayıtların saklanması ve yönetimiyle ilgili yönetmelikler, kaynak kullanım ve zamanlamalar, doğrulama ve onaylama gibi diğer süreçlerden seçilen kalite güvence işlemleri bulunmalıdır.

Kalite güvencesi diğer süreçlerle koordineli çalışmalıdır. Böylece gereksiz ve zarar verici mükerrer değerlendirmeler olmaması, planlı değerlendirmelerin tüm taraflar için yararlı ve gerçekçi olması, ilişkili süreçlerdeki çakışan raporların çözülmesi sağlanır. Bu süreç onaylama, doğrulama, ortak gözden geçirme, denetim ve problem çözümleme alanlarının sonuçlarını kullanabilir.

4. Doğrulama

Doğrulama süreci, bir işlemin çıktısı olan yazılım ürününün, o işlem için önceki işlemler tarafından belirlenen, ihtiyaç veya şartları yerine getirip getirmediğini belirleme sürecidir. Örneğin tasarım süreci sonunda, analiz sürecinden tasarıma aktarılan ihtiyaçların ürün mimarisine eklenmesi doğrulanır. Doğrulama planlı yürütülmelidir.

Doğrulama süreci, sağlama ve geliştirme gibi süreçler tarafından kullanılır. Maliyet ve performans etkinliği için *doğrulama* ile *doğrulama* kullanan diğer süreçler mümkün olduğunca erken bütünleştirilmelidir. Analiz, gözden geçirme ve test gibi süreçler doğrulama süreçleridir. Doğrulama kalite güvencesi gibi bağımsız yürütülmelidir. Bu süreçte tespit edilen sorunlar, problem çözümleme sürecine aktarılır. Paydaşın yeterliliği, süreç, plan, ihtiyaç, tasarım, kod, bütünleşme ve belgeler doğrulanmalıdır.

5. Onaylama

Onaylama süreci, ihtiyaçlar ve son ürünün (geliştirilen yazılım veya sistemin) planlanan kullanım amacını yerine getirip getirmediğini belirleme sürecidir. *Onaylama*da asıl olan ürünün istenilenleri karşılayıp karşılamadığıdır. Kalite güvencesi gibi bağımsız yürütülmelidir. Bulunan sorunlar, *problem çözümleme* sürecine aktarılır. Onaylama süreci aşağıdaki işlemleri içerecek şekilde planlanmalıdır.

- Yükleme, sınır ve tek değer giriş testleri yapmak
- Yazılımın hatayı yalıtma ve etkisini en küçük dairede sınırlama yeteneğini doğrulanmak, hangi durumlarda operatör yardımı gerektiğini belirlemek
- Geliştirilen yazılım kullanılarak belli işlemlerin yapılıp yapılamadığını, kullanıcı temsilcisiyle birlikte incelenmek
- Yazılımın genel amacına uygun çalışıp çalışmadığını belirlemek ve hedeflenen ortamın çeşitli alanlarında yazılımı test etmek

6. Ortak gözden geçirme

Ortak gözden geçirme, projedeki işlemlerin durumu ve ürünlerin uygunluğunu değerlendirme sürecidir. Gözden geçirilen ve gözden geçiren şeklinde iki sorumlu vardır. Sözleşme süresince proje yönetimi ve teknik düzeyde yürütülür. Bulunan sorunlar, *problem çözümleme* sürecine aktarılır. Gözden geçirmeler plana bağlı yürütülmelidir.

Proje yönetim gözden geçirmeleri işlemlerin plana göre yürütülmesi için mevcut durumun değerlendirilmesini, kaynakların yeterliliğinin genel kontrolünü, proje yönünün değiştirilmesi için gerekenleri ve proje başarısını etkileyen riskleri kapsar. Gözden geçirmeler yürütülürken projenin durumu, seçimler, teknik anlaşmalar ve riskler konusunda üst yönetimin bilgilendirilmesine özellikle dikkat edilmelidir. Ayrıca *üst yönetim yetkisindeki* risk ve problem çözümlerinin belirlenmesi de gerekir.

Teknik gözden geçirmeler yazılımın bütünlüğünü, standartlara uygunluğunu, değişimlerin doğru ve sadece istenen kısımlarda yapıldığını, plana uygunluğu ve süreçte bir sonraki aşamaya geçmeye hazır olup olunmadığının kontrolünü kapsar.

7. Denetim

Denetim ihtiyaç, plan ve sözleşme arasındaki uyumun belirlenmesi sürecidir. Denetleyen ve denetlenen şeklinde iki ekiple de bir plana bağlı gerçekleştirilir. Diğer gözden geçirmeler gibi bağımsız bir ekiple yürütülmeli ve çıkan sorunlar, problem çözümleme sürecine aktarılmalıdır.

Sözleşmedeki standartlara uyum denetimin temel konusudur. Denetlenmesi gereken konular: Geliştirilen yazılım, kod, test verisi, kullanıcı belge ve yapılan işlemlerin; (sözleşmede) belirlenmiş standart ve planlara uygunluğudur.

8. Problem çözümleme

Problem çözümleme geliştirme, sağlama, alım gibi süreçleri gerçekleştirirken karşılaşılan hataları, sebepleri ne olursa olsun, analiz etme ve ortadan kaldırma sürecidir.

Önce olası tüm problemlerin çözümüyle ilgili bir altyapı oluşturulur, sonrasında problem çıktığında belirlenmiş yöntemlerle çözülmeye çalışılır. Süreçteki işlemler:

- Kapalı döngü şeklinde sürece giren tüm problemleri tespiti etme, inceleme, ilgililere bildirme, sebeplerini tespit etme ve mümkünse ortadan kaldırma, problemi çözümlenme, süreci izlenmek
- Problemleri sınıflamak ve öncelik belirlenmek
- Hata eğilim analizleri ve raporlanması
- Problem çözüm ve eğilimleri belirlenmek, yazılımda gerekli değişikliklerin doğru şekilde yapıldığı ve yeni hata oluşumunu değerlendirilmek

Üründe yazanın kontrolünde bulunan problemleri, yazan kendisi çözmelidir. Yazanın kontrolü dışında bulunan sorunlar, *problem çözümleme* sürecine aktarılır.

15.3.3. Kurumsal Yaşam Döngüsü Süreçleri

Kurumsal yaşam döngüsü süreçleri birbiriyle ilişkili yaşam döngüsü süreçleri ve kurum çalışanlardan oluşan bir altyapının inşa edilmesi, uygulanması ve sürekli geliştirilmesi amacıyla kurumun yürüttüğü süreçlerdir. Tipik olarak özel bir proje ve sözleşme alanını aşan daha geniş bir kapsamda çalışırlar. Burada öğrenilen dersler kurumun gelişmesine katkı sağlar.

1. Yönetim

Yönetim süreci, kendisiyle ilgili süreçleri yönetmesi gereken herhangi bir ekip tarafından kullanılan işlem ve görevleri içerir. Bir yönetici, projeyi veya süreçlerin belli görevlerini yönetebilir. Proje yönetimi için başlangıç ve kapsam belirleme, planlama, uygulama ve kontrol, gözden geçirme ve değerlendirme, kapatma işlemleri önerilmiştir. Yöneticinin görevleri, sayılan tüm işlemlerin yapılması ve kontrol edilmesidir. Yönetici süreci izler ve çıkan sorunları çözer.

Planlama yönetim kısmında en kapsamlı açıklanan işlemdir. *Plan en az* görev tamamlanma süre ve harcanacak emek tahmini, görev için gerekli kaynaklar, görev dağılımı, sorumluluk atama, görev ve süreçle ilgili riskler, kalite kontrol ölçütleri, süreci gerçekleştirme maliyeti, sağlanacak çevre ve altyapı bilgilerini içermelidir.

2. Altyapı

Altyapı süreci, diğer süreçler için gereken altyapıyı oluşturma ve sürdürme sürecidir. Bu süreç geliştirme, çalıştırma ve bakım işlemlerinin sağlıklı yürütülmesi için gerekli donanım, yazılım, araç, teknik ve standartlar sağlamayı içerir. Altyapı ihtiyaçları belirlenir ve bunlar sürecin çalışması için gerekli zamanda hazır edilir.

3. İyileştirme

İyileştirme süreci bir yazılım yaşam döngüsü sürecini inşa, değerlendirme, ölçme, kontrol ve iyileştirme işlemleri için yürütülen kurumsal süreçtir. Süreci inşa, değerlendirme ve iyileştirme aşamalarını içerir.

Süreci inşa, yazılım yaşam döngüsü süreçlerinin kurum faaliyetlerine uygulanmasına benzer şekilde, yazılım süreçlerine uygulanacak kurumsal süreç paketleri oluşturmak, belgelemek ve nasıl kontrol edilecekleri belirlemektir. *Sürecin değerlendirme* şekli belirlenmeli, belgelenmeli ve gözden geçirilmelidir. *Süreci iyileştirme* aşamasında iyileştirmeler değerlendirilir ve gerekli görüldüğü anda uygulanır. Belgeler buna göre düzenlenir. Tarihsel veriler toplanarak süreçlerin kuvvetli-zayıf yönleri bu verilere göre analiz ve değerlendirilir. Kalite için yapılan harcamalar kaydedilir, ilerdeki problemlerin fark edilmesi ve çözümünde bu veriden faydalanılır.

4. Eğitim

Eğitim süreci eğitimli personel sağlama ve idame ettirme sürecidir. Temel süreçleri yürütebilmek büyük ölçüde bilgili ve vasıflı personel bulunmasına dayanır. Eğitimin erken planlanması ve gerçekleştirilmesi, yazılım süreçlerinde gerek duyulduğu anda eğitimli personele sahip olmayı sağlar.

Eğitim, için öncelikle eğitim süreci oluşturulur. Bu süreçte eğitim ihtiyaçları paydaşlarla görüşülür. Kişilere gerekli eğitim türleri ve seviyeleri belirlenir. Bunların nasıl karşılanacağı planlanır. Sonrasında eğitim malzemeleri hazırlanır. En sonunda eğitim gerçekleştirilir, yapılanlar kaydedilir ve değerlendirilir.

15.3.4. Kuruma Uyarlama Süreci

Uyarlama, milletlerarası standartları kuruma uyarlama için yapılması gereken temel işlemleri tanımlayan süreçtir. 10207 içerisinde kuruma göre düzenlemelerin nasıl yapılacağını anlatan bir kısım bulunmaktadır. *Uyarlama sürecinin*, ilk aşamasında uyarlama yapacak kurumun iş sahasına (kamu veya telekom gibi) uygun düzenlemeler önerilmektedir. İkinci aşamada geliştirilen projeye özel düzenlemeler yapılır. Düzenleme için standardın kullanılmayacak kısımların çıkartılması yöntemi önerilmiştir. Sonrasında standarttaki hükümlerin nasıl uygulanacağı sözleşmeye göre yorumlanır.

15.4. IEEE/EIA Standart 1058

IEEE 1058, yazılım proje planının içeriğini ve şeklini(formatını) belirlemeyi hedefleyen resmi bir standarttır [IEEE-1058]. Örnek bir plan veya kesin bir teknik önermez. Her yazılım proje yönetim planında olması gereken (en az) özellikleri açıklar. *IEEE/EIA 12207* ile uyumludur ve ana standarttaki işlerin nasıl plana dönüşeceğini açıklar.

Bu standarda uygun geliştirilen planlarda, giriş ve projeyi açıklama kısmı bulunur. *Giriş kısmında* başlık, imza, değişim tarihçesi, önsöz, şekil ve tablo listesi sayfaları vardır. *Başlık sayfası* yayın tarihi, benzersiz bir tanımlayıcı (taslak ve sürüm numarası) ve yayınlayan kurum bilgileri içerir. *İmza sayfasında*, kabulden sorumlu kişilerin imzalarını bulunur. *Değişim tarihçesi kısmında* proje ismi, tarih, sürüm numarası, değişen sayfa listesi, ilgili sürümde yapılan değişikliği tanımlayan özet ifade, önceki sürüm numarası ve yayın tarihi bulunur.

Proje açıklamaları kısmında aşağıdaki bilgiler bulunur.

1. **Genel bakış:** Proje özeti, varsayımlar, projenin teslimatları (çıktıları) ve zaman- maliyet özetleri bu kısımda yer alır.

1.1 **Proje özeti:** Projenin ve ortaya çıkacak ürünün amaç, hedef ve kapsam gibi konularında özet bilgi vermek için kullanılan kısımdır. Çalışan diğer proje ve süreçler ile iletişim ve bütünleşmenin nasıl sağlanacağı da bu kısımda açıklanır.

1.1.1 **Amaç hedef ve kapsam:** Proje hedeflerinin kısa net özetini verir. İş ve sistem hedeflerini karşılayacak, ürün ve yöntemi tanımlar. Ürün ve proje kapsamı dışında bırakılacak kısımları da açıklar.

1.1.2 **Varsayımlar ve kısıtlar:** Varsayımlar doğruluğu ilk anda kanıtlanamayan ön kabullerdir. “*Müşterilerin proje geliştirme sürecine tam zamanlı katılımını sağlama beklentisi*”, “*Yazılım testi için bağımsız firmadan ayrı bir ekip görevlendirilme ihtimali*” gibi ilk anda kesin bilinmeyen konular bu varsayımlara örnektir. Kısıtlar projedeki zaman, bütçe, kaynak, ara yüz ve teknoloji gibi konularda projeye dayatılan şartlardır.

1.1.3 **Teslimatlar:** Alıcıya sağlanacak ürün, teslim tarihi, teslim yeri, proje sözleşmesindeki terimleri karşılamak için gerekli miktar, teslim aracı, ürünün paketlenme ve kullanım şekliyle ilgili özel açıklamalar bu kısımda açıklanır. İhtiyaç listesi ve ürün listesi şeklinde ayrı olarak hazırlanıp ana plandan referans da edilebilir.

1.1.4 **Zamanlama ve bütçe özeti:** Yazılım projesindeki ana işlemler ve destek faaliyetleriyle ilgili, zamanlama ve bütçe özeti bu kısımda açıklanır. Ana işlemler, iş bölümleme yapısının en üst seviyesindeki işleri işaret eder.

1.2 **Planın gelişimi:** Bu kısımda yazılım proje planının programlı veya programsız değişimleri belirtilir. Değişimlerin kullanıcılara neşredilmesi için kullanılacak yöntemlerden bahsedilir. Planın, konfigürasyon yönetimine nasıl ekleneceği ve gelecekte plandaki değişikliklerin hangi yöntemle kontrol edileceği de bu kısımda tanımlanır.

2. **Referanslar:** Referanslar kısmında proje planından referans edilen diğer belge ve kaynakların listesi bulunur. Her belge için başlık, numara, tarih, yazar, erişim yolu ve yayıncı gibi bilgiler saklanır. Bu bilgiler kullanılarak ilgili belgeye erişilebilmelidir.

3. **Tanımlar:** Planda açıkça anlaşılması gereken terim ve kısaltmaların tanımlandığı veya açıklayıcı referanslar sağlanan kısımdır.

4. Proje organizasyonu: Bu kısım, proje haricindeki kurumsal yapılarla proje arasında ara yüzler belirlemek, proje iç organizasyon yapısı, rol ve sorumlulukları tanımlamak için kullanılır. Organizasyon şeması gibi şekiller kullanılabilir.

4.1 Harici ara yüzler: Proje ve harici kurumlar arasındaki kurumsal sınır tanımlarıdır. Bunlar ana kurum, kurum dışında yazılım satın alan müşteri veya son kullanıcı olabilir. Bu kişi veya grupların isimleri, telefon, e-mail gibi iletişim bilgileri proje planında yer almalıdır.

4.2 İç ara yüzler: Proje ekibinin iç yapısının nasıl oluşturulacağını açıklar. Kurum içinde destekleyici süreçleri yürüten sistem analiz ekibi veya konfigürasyon yönetim grubu gibi diğer birim ve ekiplerle ilişkiler de bu kısımda açıklanır. Proje ekinin organizasyonu, ekipteki rol ve gruplar arasındaki çalışma şekli, iletişim hatları ve irtibat noktaları tanımlanmalıdır.

4.3 Rol ve sorumluluklar: Temel işlerin yapısı ve bu işlerden kurum içinde hangi biriminin sorumlu olduğunu Rol-sorumluluk matrisiyle açıklar.

5. Yönetim süreçleriyle ilgili planlar: Proje yönetim sürecinin belirlendiği kısımdır. Proje kapsamıyla uygun olmalıdır. Bu kısımda *başlangıç planı*, *çalışma planı*, *kontrol planı*, *risk yönetim planı* ve *kapatma planı* bulunmaktadır.

Tahminin de bir planı olması önerilmektedir. 1058 standardında *başlangıç planı* kısımda önerilen *proje tahmin planı*, *PMI* ve *CMMI* diğer gibi standartlardan farklılık arz eder. Bu planla projeyi yürütmek için gerekli maliyet ve zaman; tahmin yöntemi, araç ve teknikler; tahminlerin güvenilirlik düzeyi ve yenilenme periyodu belirlenir.

6. Teknik süreç planları: Bu kısımda geliştirme sürecinin modeli, ürün geliştirmede kullanılacak teknik yöntem, araç ve teknik belirtilir. Proje altyapısını sürdürmek için gerekli planlama ve *proje kabul planı* yapılır. *Geliştirme süreç planı*, *yöntem*, *araçlar*, *teknikler*, *altyapı planı* ve *ürün kabul planı* bu kısımda bahsi geçen planlamalardır.

7. Destek süreç planları: Proje süresine yayılan destekleyici süreçlerin planıdır. En azından *belgeleme*, *konfigürasyon yönetimi*, *kalite güvencesi*, *doğrulama*, *onaylama*, *ortak gözden geçirme*, *denetim* ve *problem çözümleme* alanlarında yapılması önerilir. Bunlardan biri yapılmayacaksa niçin yapılmadığı harici olarak değerlendirilmelidir. Bu planlar diğer ana ve ayrıntı kısımlarla uyumlu olmalıdır.

8. Ek planlar: İhtiyaçları veya sözleşmeyi sağlamak için bazı projelerde gerekli olan *kullanıcı eğitimi*, *ürün kurulumu* ve *bakım* gibi planları kapsar.

9. Ekler: Eğer içine eklenirse, proje planını karmaşıklştıracak yardımcı ayrıntıları içerir. Bu ayrıntıların dolaylı olarak referans adreslerini de içerebilir.

10. Dizin: Plan içinde kullanılan terim ve kısaltmaların dizini bulunması için kullanılır. İsteğe bağlı ancak faydalı bir kısımdır.

IEEE/EIA 1058 Standart Proje Planı Şablonu

Giriş

Başlık, Değişiklik Tarihçesi, Önsöz, Şekil Listesi, Tablo Listesi

1. Genel Bakış

1.1. Proje Özeti

- 1.1.1. Amaç, Hedef ve Kapsam
- 1.1.2. Varsayımlar ve Kısıtlar
- 1.1.3. Proje Çıktıları
- 1.1.4. Zamanlama ve Bütçe Özeti

1.2. Planın Gelişimi

2. Referanslar

3. Tanımlar

4. Proje Organizasyonu

- 4.1. Harici Ara Yüzler
- 4.2. İç Ara Yüzler
- 4.3. Roller ve Sorumluluklar

5. Yönetim Süreçleriyle İlgili Planlar

5.1. Başlangıç Planı

- 5.1.1. Proje Tahmin Planı
- 5.1.2. İnsan Kaynakları Temin Planı
- 5.1.3. Kaynak Temin Planı
- 5.1.4. Proje Ekibi Eğitim Planı

5.2. Çalışma Planı

- 5.2.1. İş Bölümleme Yapısı (WBS) ve İş Paketleri
- 5.2.2. Plan Bağımlılıkları
- 5.2.3. Kaynak Atamaları
- 5.2.4. Bütçe Atamaları

5.3. Kontrol Planı

- 5.3.1. İhtiyaçlar
- 5.3.2. Zaman Planı
- 5.3.3. Mali Kaynaklar
- 5.3.4. Kalite
- 5.3.5. Ölçümler
- 5.3.6. Raporlama

5.4. Risk Yönetim Planı

5.5. Kapatma Planı

6. Teknik Süreç Planları

- 6.1. Geliştirme Süreç Modeli
- 6.2. Metot, Araç ve Teknikler
- 6.3. Altyapı Planı
- 6.4. Ürün Kabul Planı

7. Destek Süreç Planları

- 7.1. Konfigürasyon Yönetimi Planı
- 7.2. Doğrulama ve Onaylama Planı
- 7.3. Belgeleme Planı
- 7.4. Gözden Geçirme ve İzleme Planı
- 7.5. Problem Çözümleme Planı
- 7.6. Alt Yüklenici Yönetim Planı
- 7.7. Süreç İyileştirme Planı

8. Ek Planlar

Ekler

Dizin