1. Amaç – Kapsam

Bu politikanın amacı, Necmettin Erbakan Üniversitesi tarafından kullanılan yazılımların uluslararası metodolojilere, bilgi güvenliği standartlarına ve Kurum güvenlik politikalarına uygun geliştirme ve yedeklenmesini sağlamaktır.

2. Sorumlular

Bu prosedürün hazırlanması ve yönetiminden Bilgi Güvenliği Ekibi ile Sistem, Yazılım ve Güvenlik Hizmetleri Birimi sorumludur. Tüm doküman taleplerinde Bilgi Güvenliği Sorumlusu yetkilidir.

3. Tanımlar

Program Kaynak Kodu: Programcılar tarafından yazılan ve çalıştırılabilir dosyalar oluşturmak için derlenen (compiling) kodlar

Veri Tabanı: Birbirleriyle ilişkili bilgilerin depolandığı alanlardır.

İmplementasyon: Uygulama

4. UYGULAMA

4.1 Genel

Kurum bünyesinde yazılım geliştirme faaliyetleri üç ana başlık altında incelenebilir. Bunlar proje özelliğinde yazılım geliştirme faaliyetleri, proje özelliğinde olmayan yazılım geliştirme faaliyetleri ve devam eden projelerde yazılım geliştirme faaliyetleridir. Yazılım talebi *Bilgi İşlem Daire Başkanlığı Yazılım Geliştirme Hizmeti Talep Formu şeklinde üniversite portalından* ve başvuru kanallarından yapılır. Bilgi İşlem Daire Başkanlığı tarafından değerlendirmeye alınan talebin, proje kapsamına alınıp alınmayacağının kararı verilir. Bu aşamadan sonra talebin kapsamına göre süreç başlatılır. Temin edilecek veya geliştirilecek olan uygulamalar için, güvenlik ihtiyaçları, temin veya geliştirim öncesi Yazılım Geliştirme Ekibi tarafından değerlendirilir.

4.2 Proje Kapsamında Yazılım Geliştirme Döngüsü

Üniversitemizde proje bazlı yürütülen yazılım geliştirme çalışmaları planlama, tasarım, geliştirme ve test yaşam döngüsünden oluşur. Bu aşamalar aşağıda detaylandırılmıştır.

4.2.1 Proje Planlama

Bu aşamada yazılım geliştirme projesinin analizinin gerçekleştirilmesi, önceliklendirmenin yapılması, proje planının hazırlanması, test stratejilerinin belirlenmesi ve yazılım geliştirme görevlerinin dağıtılması süreçleri bulunmaktadır. Projenin bu süreçlerinde Azure Devops kullanılmaktadır.

4.2.2 Ön Analiz

Proje Ekibi tarafından proje olarak kabul gören taleplerin önceliklendirmesinin yapılabilmesi için genel iş gereksinimlerinin ve ihtiyaç duyulan kaynakların belirlendiği aşamadır. Bu aşamada ortaya çıkan gereksinimlerin Kurumun hangi stratejik hedeflerine yönelik olduğu tespit edilir. Aynı zamanda proje sürecinde ihtiyaç duyulan kaynaklar Proje Ekibi tarafından belirlenir.

4.2.3 Değerlendirme ve Önceliklendirme

Yazılım Geliştirme birim yöneticisi ile Proje Ekibi değerlendirme yapar ve projenin önceliğini belirler. Proje düşük, normal veya yüksek öncelikli olabilir. Yüksek öncelikli projeler için fazla mesai veya ek insan kaynağı değerlendirilir. Projenin kapsamına göre proje sorumlusu belirlenir.

4.2.4 Analiz

Yazılım Geliştirme birim yöneticisi tarafından görevlendirilen personelin yapmış olduğu değerlendirmeye ve önceliklendirmeye bağlı olarak detaylı analiz aşamasına geçilir. İstek sahibi birimden iş ihtiyaçlarının alınması, alınan iş ihtiyaçlarına bağlı olarak fonksiyonel, fonksiyonel olmayan gereksinimlerin çıkarılması ve bunlara bağlı olarak güvenlik gereksinimlerinin analizi bu aşamada gerçekleştirilir.

Analiz sonucunda tasarım yavaş yavaş şekillenmeye başlar, kullanılacak teknoloji belirlenir (veri tabanı gereksinimi ortaya çıkar, dağıtık mimari/merkezi sunucu/yerel uygulama şeklinde mimari tercihleri değerlendirilir vb.).

4.3 Proje Planının Hazırlanması

Önceliklendirmesi yapılmış, iş ihtiyaçları analiz edilmiş, fonksiyonel gereksinimleri çıkarılmış proje için proje planı oluşturulur. Proje planı geliştirme aşamasından canlı sistemde uygulama aşamasına kadar tüm aktiviteleri içermelidir. Atanan proje sorumlusu proje planını hazırlar. Proje planının hazırlanmasının ardından istek sahibi, proje takım çalışanları, gerekli sistem, ağ ve veri tabanı yöneticileri ile bir başlangıç toplantısı yapılır. Bu toplantıda proje planı gözden geçirilir. Kaynakların durumu göz önünde bulundurularak proje planına uygun takvimlendirme yapılır. Bu proje takviminin gerçekçi oluşturulması ve bu plana uyulması başta proje sorumlusu olmak üzere tüm proje takımı çalışanlarının sorumluluğundadır.

4.4 Proje Rol ve Sorumlulukların Atanması

Proje planındaki faaliyetler ile ilgili sorumluluklar belirlenir. Bu sorumluluklar belirlenirken görevler ayrılığı ilkesi dikkate alınır. Yazılım geliştirme rol ve sorumlulukları değerlendirildiğinde yazılımın bir modülünü yazan kişinin aynı modülü test etmemesinin sağlanması diğer test modüllerinin test sorumluluğunun kendisine verilmesi Kurum yazılım geliştirme rol ve sorumluluklarında görevlerin ayrımı prensiplerini oluşturmaktadır. Geliştirilen yazılımın işletilmesi görev ve sorumluluğu yazılım geliştiren çalışanda olmaması sağlanır. Proje planında aşamalandırılmış yazılım geliştirme faaliyetleri için uygun bilgi ve tecrübeye sahip takım çalışanları seçilip bu takım çalışanlarına görev, sorumluluk ve yetkilerinin anlatılması sağlanır. Yazılım geliştirme görevleri önceliklendirmesine bağlı olarak ilgili sorumlulara atanır.

5. Yazılım Tasarımı ve Geliştirme

5.1 Tasarım

Proje kapsamında geliştirilecek yazılım ile ilgili planlama aşamasında tanımlanan gereksinimler tasarım aşamasının girdilerini oluşturur. Tasarım aşaması proje planına uygun şekilde ve ortak tasarım anlayışının geliştirilmesi ile yürütülür. Proje sorumlusu, tasarım için belirlediği ekipler ile toplantılar yapar.

5.2 Geliştirme, Test ve Gerçek Sistemlerin Ayrılması

Yazılım geliştirme projelerinde yazılım geliştirme ortamı, test ortamı ve canlı sistemler birbirinden ayrılır. Canlı sistemlerde yazılım geliştirme ve test faaliyetleri yapılmaz. Yazılım geliştirme ortamlarına (veri tabanı sunucusu, uygulama sunucusu vb.) erişim yetkisi sorumlu proje ekibine verilir. Bütün erişimler kayıt altına alınır. İstisnai durum olarak canlı sisteme eş bir test ortamı sağlanamaması durumunda iş ihtiyaçları göz önünde bulundurularak canlı ortamlarda test yapılmasına izin verilebilir. Bu izin Yazılım Geliştirme birim yöneticisi ya da birim yöneticisi tarafından görevlendirilen personel tarafından verilir. Bu durumlarda teste başlanılmadan olası sorunlara karşı sistem ve veri tam yedeğinin alınması, test ve geçiş çalışmalarının kesinlikle işi aksatmayacak gün ve saatlerde üst yönetim seviyesinde izin alınarak gerçekleştirilir.

5.3 Yazılım Geliştirme ve Özelleştirme

Yazılım geliştirme yaşam döngüsü aşağıdaki şemada açıklandığı şekilde gerçekleştirilir. Yazılım geliştirme süreci gereksinimlerin geliştirilmesi ile başlayıp uygulama ile ilgili verilen eğitim ve destek hizmetleriyle tamamlanmaktadır. Her bir aşamada sorun yaşanması veya tasarımın güncellenmesi / değiştirilmesi söz konusu olduğunda gerekli durumlarda bir veya daha fazla önceki seviyeye geri dönülür, güncellemeler gerçekleştirilip sonraki aşamaya doğru ilerlemeye devam edilir.

5.4 Yazılımların Test Edilmesi, Dokümantasyonu ve İmplementasyonu

5.4.1 Bileşen (Unit), Fonksiyon, Entegrasyon, Sistem ve Sistem Entegrasyon Testlerinin Gerçekleştirilmesi

Yazılım bileşenleri, en alt seviyeden en üst seviyeye kadar test edilir. Test sonuçları kayıt altına alınır. Her bir test için en az aşağıdaki detaylar yer almalıdır;

- Test edilen yazılım birimi
- Testi gerçekleştiren personel
- Gözlemlenen / ölçülen test sonucu

Test ortamındaki veriler gerçek ortamlardaki verilerin belli bir algoritma kullanılarak bozulması ile hazırlanır. Yani test ortamındaki veriler gerçek verilerden oluşmaz. Test verileri izole ve erişim kısıtlaması olan alanda tutulur.

Geliştirilen uygulamaların farklı ortamlara (işletim sistemleri, internet tarayıcılar vs) uyumluluğu da test edilir.

Yapılan sızma testleri ile geliştirilen yazılımlardaki güvenlik test edilir.

5.4.2 Teknik ve Kullanıcı Dokümantasyonunun Geliştirilmesi ve Güncellenmesi

Yazılım geliştirme süreci devam ederken dokümantasyon çalışmalarına başlanır. Yazılım geliştirme süreci içerisinde kabul testleri gerçekleşene kadar dokümanlar olgunlaştırılır. Yazılım ekibi ve ilgili birim tarafından kabulü gerçekleştirildiği tarihe kadar dokümantasyonun tamamlanması hedeflenir. Geliştirilen yazılımda bir ara yüz bulunmaması, sorumlu veya kullanıcı kavramının olmadığı durumlar için kılavuz doküman hazırlanmasına gerek yoktur.

5.4.3 Kurulum ve Kabul Testlerine Hazırlık

Kaynak kodların derlenerek kurulum dosyalarının hazırlanması sistem entegrasyon testlerinin ardından yapılması gereken bir iş adımıdır. Yazılımcı, kaynak kodları geliştirdikçe versiyonlarını tutmakla sorumludur. Geliştirilen yazılımlar, Proje Sorumlusu tarafından versiyon bilgileri belirtilerek saklanır. Proje sorumlusu sadece yazılım kaynak kodlarını değil yazılım geliştirme projesi ile ilgili tüm dokümantasyon versiyonlarının tutulmasını sağlar.

5.4.4 Canlı Sisteme Uygulama

Yazılım geliştirme ve test süreçlerinden geçip ürün haline gelmiş proje canlı sisteme geçirilirken aşağıdaki hususlara dikkat edilmelidir.

- Canlı sistemin ve verilerin tam yedeği mutlaka canlı sisteme geçişten önce alınmış olmalıdır.
- Olası sistem ve servis kesintileri için ilgili taraflar önceden bilgilendirilmeli ve izin alınmalıdır. Çalışmayı en az etkileyecek saatler seçilmelidir.
- Canlı sisteme geçişte yaşanan tüm problemler olay bildirimi olarak kayıt altına alınmalı ve sorun çözümleri bu kayıtlara eklenmelidir.

5.4.5 Acil Durumlar İçin Geliştirme Stratejisi

Acil durumlarda yazılım geliştirme için iki durum söz konusudur; bunlardan ilki geliştirme ortamının çalışılamayacak hale gelmesi ve iş gereksiniminin önceliklendirmesinin acil olduğu durumlar, ikinci olarak canlı sistemde çok yüksek öncelikli değişiklik gereksinimi olması durumunda ürünün gerçekleşme şekli canlı sistemde olmasını gerektirdiği durumlardır.

5.4.6 Aktivite Yönetimi

Yazılım geliştirme faaliyetleri sırasında üretilen ve iletilen talepler, hatalar ve diğer raporlar erişim kısıtlı bir alanda kayıt altına alınır ve izlenebilir hale getirilir. Yazılım geliştirme aktiviteleri proje sorumlusu ve ilgili birim yetkilileri tarafından izlenir.

5.4.7 Proje Seviyesinde Olmayan Yazılım Geliştirme Faaliyetleri

Proje seviyesinde olmayan yazılım geliştirme aktivitelerinde proje seviyesindeki eylemlerden farklı olarak aşağıdaki faaliyetlerin yürütülmesine gerek yoktur;

- Özel bir test stratejisinin geliştirilmesine
- Proje planı tanımlanmasına

Bunların haricinde proje seviyesinde olan yazılım geliştirme faaliyetlerinde olduğu gibi aşağıdaki faaliyetler yürütülmelidir;

- İş ihtiyaçlarının tanımlanması
- Analiz
- Önceliklendirme
- Geliştirme rol ve sorumluluklarının belirlenmesi
- Yazılım geliştirme ve özelleştirme
- Bileşen (Unit), fonksiyon, entegrasyon, sistem ve sistem entegrasyon testlerinin gerçekleştirilmesi
- Teknik ve kullanıcı dokümantasyonunun geliştirilmesi ve güncellenmesi
- Kaynak kodların kurulum için hazırlanması ve kabul testlerine hazırlık
- Canlı sisteme uygulama
- Aktivitelerin izlenmesi

5.4.8 Devam Eden Projelerde Yazılım Geliştirme Faaliyetleri

Birimler, mevcut yazılımın güncellenmesi ya da geliştirilmesi için taleplerini *üniversite portalından* yazılım ekibine iletir.

- Yazılım geliştirme süresi göz önünde bulundurulup yazılımda yapılacak değişiklikler ya da eklemeler Kurumdaki işleyiş dikkate alınmalıdır.
- Yazılımda istenen değişiklikler ve ekranlarda görünmesi istenen açıklamalar anlaşılır bir dille yazılım ekibine iletilmelidir.
- Yapılan tasarımlar ve geliştirilen kodlar talep sahibi birim yetkilileri tarafından test edilir ve onaylanır.

Bunların haricinde proje seviyesinde olan yazılım geliştirme faaliyetlerinde olduğu gibi aşağıdaki faaliyetler yürütülür;

- Analiz
- Yazılım geliştirme ve özelleştirme
- Bileşen (Unit), fonksiyon, entegrasyon, sistem ve entegrasyon testlerinin gerçekleştirilmesi
- Teknik ve kullanıcı dokümantasyonunun geliştirilmesi ve güncellenmesi
- Kaynak kodların kurulum için hazırlanması ve kabul testlerine hazırlık
- Canlı sisteme uygulama
- Aktivitelerin izlenmesi

6. GÜVENLİ YAZILIM GELİŞTİRME

6.1 Uygulama Güvenlik Gereksinimleri

Temin edilecek veya geliştirilecek olan uygulamalar için, güvenlik ihtiyaçları, temin veya geliştirim öncesi Yazılım Geliştirme tarafından değerlendirilir. Belirlenen güvenlik gereksinimleri üzerinde ilgili ekibin hem fikir olması sağlanır ve sonrasında Bilgi İşlem Daire Başkanı'nın onayı alınır. Uygulama alımı yapılırken karar aşamasında, uygulamanın güvenlik kriterlerine uyumluluğu, güvenilirliği bilinen bir üretici tarafından geliştirilmiş olması, şimdiye kadar tespit edilmiş güvenlik zafiyetlerinin az olması ya da faydasının zafiyetlerinden daha ağır basması, yaygın kullanılması, güvenli geliştirme yöntemleri

kullanılarak geliştirilmiş olması vb. kriterlere bakılır. Aşağıdaki kriterleri sağlıyor olması ilgili uygulamanın temini için tercih sebebi olarak kabul edilir.

- Arayüzün HTTPS üzerinden hizmet vermesi
- Two Factor Authenticationı desteklemesi
- SQL Injection saldırılarına korumalı olması

Uygulama temin edildikten sonra hizmete alınmadan önce, Yazılım Geliştirme tarafından fonksiyon/güvenlik testlerine tabi tutulur. Testlerden başarılı olarak geçen uygulamalar hizmete alınır. Başarılı olamayan uygulamalar için, güvenliği arttıracak fırsatlar araştırılır. Eğer bu mümkün olamıyorsa, ürünün kullanımına izin verilmez ve iade işlemi başlatılır.

6.2 Uygulamaların Doğru İşlemesi

Uygulamalarda hata, kayıp, yetkisiz erişim, yetkisiz değişiklikler ve bilginin yanlış kullanımı engellenmelidir. Bunun için aşağıdaki kontroller uygulanır.

6.2.1 Giriş Verisi Geçerleme

Sabit veri, iş aktivite verileri vb. veri girişlerinde giriş verisinin doğruluğu, bütünlüğü aşağıdaki kontroller ile yapılır.

- Belirlenen aralık, kriter dışında bulunup bulunmama
- Veri alanlarında geçersiz karakter kullanılamama
- Kayıp veya eksik veri önlemleri bulundurma
- Üst ve alt veri limitlerini aşmasına izin vermeme
- Yetkisiz ve tutarsız verileri doğrulama

Geçerleme hatalarını gidermek ve giriş verisinin makul olduğu testlerde kontrol edilmelidir. Veri giriş sürecinde tüm personelin sorumlulukları tanımlanmalı ve bu süreçteki aktivitelerin günlükleri (log) tutulmalıdır. Bu tür önlemler alınarak code injection ve buffer overflow saldırılarına karşı koruma arttırılmış olur.

6.2.2 İç İşleme Kontrolü

- Bilgi işleme hataları yüzünden bilgi bütünlüğünün kaybolmasını önlemek amacıyla uygulamaların tasarım ve implementasyon aşamalarında aşağıdaki kontroller uygulanmalıdır;
- Veri değişikliklerini ekle, değiştir, sil fonksiyonları ile yönetmek
- Programların yanlış sırada çalışmalarını önlemek için prosedürler oluşturmak
- Verilerin doğru işlediğini sağlamak ve böylece hatalardan korunmak için uygun programları kullanmak
- "Buffer overrun" ve "Buffer overflow"saldırılarına karşı koruma önlemleri almak
- İşlemede gerçekleştirilen aktiviteler ile ilgili günlük (log) tutmak
- Herhangi bir hata durumunda programın sonlandırılmasını sağlamak
- Uygulama programlarının doğru zamanda çalıştıklarını kontrol etmek
- Sistem tarafından üretilen verilerin geçerliliğinin kontrolü
- İşlem güncellemelerinden sonra veri dosyası uyumunun sağlanması için oturum ve grup kontrolü
- Programın kendisine özgü yapısından kaynaklanan diğer güvenlik kontrolleri

6.2.3 Mesaj Bütünlüğü

Uygulamaların, diğer katman ve bileşenler ile iletişimi sırasında iletilen verinin doğruluğunun tanınabilir olması ve bütünlüğünün korunması verinin şifreli iletilmesi, şifreli iletimin sağlandığı ağların ayrıştırılması ve bu ağlara erişimlerin kısıtlanması ile sağlanır.

6.2.4 Çıkış Verisi Geçerleme

Veri işlendikten sonra üretilen farklı türdeki verilerin doğruluklarının geçerliliği kontrol edilmelidir. Bunun kontrolü aşağıdakilerle sağlanır;

- Çıkış verisinin makul olduğunun kontrolü
- Tüm verilerin işlendiğinin kontrolü
- Okuyucu veya sonraki aşamada kullanılacağı sistem için gerekli bilginin sağlandığının, doğruluğunun, kesinliğinin, bütünlüğünün ve sınıfının geçerliliğinin kontrolü
- Personellerin veri çıkış sürecindeki sorumluluklarını belirleme ve kontrol etme
- Çıkış verisi süreci ile ilgili aktivite günlükleri (log) oluşturma

6.3 Geliştirme ve Test

Geliştirilen uygulamaların, değiştirilen, güncellenen veya yükseltilen bilgi sistemlerinin testleri üretim veya gerçek sistem ortamında değil, farklı ve ayrılmış bir ortamda gerçekleştirilmelidir. Uygulamaların ve bilgi sistemlerinin üzerinde gerçekleştirilecek bu testlerden, test ortamlarının belirlenmesinden ve hazırlanmasından Yazılım Geliştirme sorumludur.

6.3.1 Yazılım Geliştirme Kuralları

Programlama yapılırken kod modülü, tablo, uygulama isimlendirme, açıklama metni yazma, belirlenen programlama kurallarına uygun yapılmalıdır. Yazılım üzerinde yapılacak geliştirmeler sırasında aşağıdaki kurallara uyulmalıdır.

- Kodun okunabilirliğini arttırmak,
- Tutarlılığı sağlamak,
- Anlaşılırlığı arttırmak,
- Yeni personelin adaptasyonunu kolaylaştırmak,
- Sistemin açık olan orijinal kodunun içerisinde özel geliştirmelerin birbirinden mümkün olduğunca kolay ayırt edilebilmesini sağlamak.

6.3.2 Değişiklik Talebi Açma/Kapama Kuralları

Yazılım kodları üzerinde yapılan değişikliklerin ne amaçla ve hangi konu ile ilişkili yapıldığının takibi yapılır. Bilgi İşlem Dair Başkanlığına üniversite portalından iletilmiş olan bugfix ve onaylı yeni geliştirme istekleri ilgili yazılım sorumlusuna havale edilir. Geliştirme yapılıp devreye alındıktan sonra, konu ile ilgili değişiklik talebi kapatılır.

6.3.3 Geliştirme Ortamının Güvenliği

- Yazılım kodları lokal bilgisayarlarda tutulur. Günlük/haftalık/Canlı yayım öncesi adımlarında git üzerinde versiyonlanır.
- Test Veritabanı: Geliştirilen kodların test edildiği veriler bu veritabanındadır. Bu veritabanı canlı sistemin kopyasının alınması ile otomatik olarak oluşturulur. Bu sayede canlı sistemdeki yetkiler aynen geçerlidir. Local yada serverda ayrıca bir veritabanı üzerinde olabilir.
- Canlı Veritabanı: Canlı sistemin verileri bu veritabanında tutulur.

6.3.4 Test Ortamı Güvenliği ve Test Verisi

Yazılım geliştirmelerin ve hata çözümlerinin yapılabilmesi için test veri tabanına ihtiyaç vardır. Canlı sistemin yedeğinin alınıp verilerin belli bir algoritmaya göre bozularak geri yüklenmesi ile test ortamı oluşturulur. Test sistemi canlı sistemin bire bir kopyasından oluşturulduğu için canlı sistemdeki yetkiler aynen geçerli olarak gelir. Yazılım geliştirme sonucunda hata çözümleri için yapılan testlerde gerçek test verisinin kullanılmamasına dikkat edilmektedir. Gerçek test verisi kullanılacak ise test verisi anonimleştirme yöntemi ile ihlal vb. olaylar için önlemler alınır.

6.3.5 Veritabanı Güvenliği

Yazılım Geliştirme personelinin canlı sistemin veri tabanında hem okuma hem de yazma yetkileri vardır. Yazılım Geliştirme personeli hassas bilgiler dışındaki bütün bilgileri görebilir. Yazılım Geliştirme jenerik hesaplar yerine kendilerine ait kullanıcı hesapları ile veritabanına bağlanır. Veritabanında yeni bir kullanıcı açılması, onay mekanizmalarından geçerek gerçekleştirilir.

6.3.6 Uygulama Geliştirme Platformu

Backend için .NET Core, FrontEnd için Angular ve Mobil için Flutter yazılım geliştirme platformları kullanılır.

6.3.7 Yazılım Kaynak Kodları

Geliştirilen bütün yazılımların kaynak kodları, DEVOPS kod deposu üzerinde saklanır.

- Yazılım Geliştirme, yaptıkları geliştirmeyi Azure DEVOPS GIT deposuna aktarmak ile sorumludur. Her canlı ortama almadan önce ve hafta bitiminde git deposuna aktarmaktan sorumludur.
- Azure DEVOPS üzerinde Yazılım Geliştirme personelinin yetkisi vardır.
- DEVOPS sisteminde yeni bir kullanıcı açılması, uygun şekilde onay mekanizmalarından geçerek gerçekleştirilir.
- Sadece bu GIT kod deposunda bulunan kodlar canlı ortama taşınabilir. Devreye alma sırasında GIT kaynak olarak kullanılır, diğer dışarıdaki kodlar devreye alınamaz.
- Devops Server' ın yedeklerini almak sistem biriminin kontrolünde günlük yapılmalıdır.

6.3.8 Web Tabanlı Intranet Uygulamalarının Güvenliği

Kullanıcı kimlik doğrulaması için TEK ŞİFRE sistemi kullanılır. Bunun dışında kullanıcı ve şifre tanımı yapısı kullanılamaz. Geliştirilen yeni uygulamalar, BGYS varlık envanterine eklenir. Bir uygulama devreye alınmadan önce, yetki erişim listesinin güncellenmesi için BGYS ekibine bildirilir.

6.3.9 Veritabanı Seviyesinde Programlama

Veritabanı seviyesindeki programlama yöntemleri (stored procedure, trigger, fonksiyon) desteklenmez.

Bunların kullanımı aşağıdaki durumlar ile sınırlıdır:

- Trigger: İzleme & log tutma amaçlı olarak kullanılır.
- Stored Procedure: DBA sistem yönetimi için kullanılır.
- Fonksiyon: Rapor almayı kolaylaştırmak için kullanılır.

6.3.10 Veritabanı Kullanıcı Güvenliği

Veritabanı kullanıcı güvenliği kuralları aşağıda belirtildiği şekildedir: Yazılım Geliştirmenin yaptığı veri değişiklikleri loglanır. Bu sayede Yazılım Geliştirmenin yaptığı değişiklikler raporlanabilir olur. Bu loglara ihtiyaç halinde bakılır. Bu loglar geçmişe dönük olarak en az 1 yıl boyunca saklanır.

6.3.11 Veritabanında Kayıt Değişiklik Bilgisi Saklama

Geliştirilen uygulamalarda kaydın kim tarafından ne zaman oluşturulduğu ve kim tarafından ne zaman değiştirildiği tutulur. Bu yüzden yeni yaratılacak olan veri tabanı tablolarında ekleyen, ekleme tarihi, değiştiren, değiştirilme tarihi alanları yer alır. Uygulamalar bu 4 alanı uygun şekilde doldurur.

6.3.12 Uygulamalar Hata Kontrolü

Geliştirilen programlar, çalışma zamanında ortaya çıkan beklenmedik hatalarla karşılaştığında aşağıdaki şekilde davranır:

- Hata detayı yazılım bilgi sistemi veri tabanının hata log tablosuna loglanır.
- Sistem güvenliği açısından kullanıcı ara yüzünde hatanın detayları gösterilmez. Bunun yerine standart bir hata görüntüsü ve hatanın Yazılım Geliştirme tarafından incelenebilmesi için hata log numarası gösterilir.
- Sql injection saldırılarından korunmak için veri tabanı sorgularında string birleştirme yöntemi kullanılmaz. Bunun yerine .netCore apileri veya Entitiy Framework, parametreli

sorgu çağırma yeteneği kullanılır.

- Cross site scripting ve Url injection saldırılarından korunmak için şu kurallara uyulur:
- Web uygulaması, gelen parametrelerin içinde zararlı script olup olmadığını kontrol eder.
- Web uygulamalarının güvenliğini sağlamak için, yeni geliştirilen uygulamalar HTTPS protokolü üzerinde çalıştırılır.

6.4 Dışarıdan Geliştirilen Yazılımlar

Dışarıdan geliştirilen yazılımların teknik şartnamelerinde bilgi güvenliği ile ilgili konular belirtilir. Bu sayede bu uygulamaların bilgi güvenliği üzerindeki etkisi kontrol altına alınır.

Aşağıda belirtilen maddeler şartnamelerde aranır:

- Bu dokümanda belirtilen Veritabanında Kayıt Değişiklik Bilgisi Saklama maddesine uygunluk.
- Bu dokümanda belirtilen Uygulamalar Hata Kontrolü maddesine uygunluk.
- Bağımsız güvenlik Kurumları tarafından uygulanan güvenlik testlerine uygunluk.

6.5 Yazılım Paketlerindeki Değişikliklerdeki Kısıtlamalar

Yazılım paketlerine yapılacak değişiklikler, gerek duyulanlar hariç önlenmeli ve tüm değişiklikler sıkı bir biçimde kontrol edilmelidir. Bir yazılım paketinin değiştirilmesi gerekli görüldüğünde, aşağıdaki noktalar göz önünde bulundurulmalıdır.

- Yerleşik kontrollerin ve bütünlük proseslerinin ele geçirilme riski,
- Üreticinin onayının alınmasının gerekip gerekmediği,
- Üreticiden gerekli değişikliklerin standart program güncellemeleri olarak temin edilebilme olasılığı,
- Değişikliklerin sonucu olarak, kuruluşun yazılımın gelecekteki bakımı konusunda yükümlülük altında kalması halinde oluşacak etki.

7. YAPTIRIM

Bu politikanın ihlal edilmesi durumunda BGYS yöneticisi tarafından gerekli personel desteği de alınarak ihlal nedeni incelenir. İhlal kasıtsız olup personelin eğitim vb. bir eksikliğinden kaynaklanıyorsa problemin kaynağını oluşturan eksikliği kapatmak için çalışma yapılır. Personel BGYS Temsilcisi tarafından e-posta üzerinden yazılı olarak uyarılır. Tüm çalışanlar, güvenlik ihlali olaylarını ve bu politikanın ihlallerini, birim amirinin bilgisi dahilinde BGYS Ekibi'ne en kısa sürede bildirme sorumluluğundadır.

HAZIRLAYAN	KONTROL EDEN	ONAYLAYAN
BİLGİ İŞLEM DAİRE BAŞKANLIĞI	KURUMSAL KALİTE GELİŞTİRME VE AKREDİTASYON KOORDİNATÖRLÜĞÜ	KURUMSAL YETKİLİ