

CEN429 Güvenli Programlama

Hafta-2

Geliştirme Ortamı Güvenliği ve Yazılım Geliştirme Süreçleri

İndir

- PDF
- DOC
- SLIDE
- PPTX



- Geliştirme Ortamı Güvenliği ve Yazılım Geliştirme Süreçleri
- Yazılım Geliştirme Süreci
 - Yazılım Geliştirme Akışı
 - Konfigürasyon Sabitleme
 - Değişiklik Başlatma ve Sınıflandırma
 - Değişikliği Onaylama ve Yayınlama
- Yazılım Geliştirme Ortamları
 - Geliştirme Ortamı Güvenliği
 - Sürüm Kontrol Sistemleri
 - Kaynak Kod Sunucu Güvenliği

Yazılım Geliştirme Süreci ve Değişiklik Yönetimi (Software Development Flow and Change Management)

1. Yazılım Geliştirme Akışı (Flow)

Teorik Açıklama:

Yazılım geliştirme süreçleri, belirli akışlarla kontrol edilmelidir. Değişikliklerin doğru şekilde yönetilmesi, projenin başarılı bir şekilde devam etmesini sağlar. Bu akış genellikle sürüm kontrol sistemleri, teknik ekipler ve proje yönetim süreçlerini içerir.

Uygulama:

- **Uygulama:** Basit bir yazılım projesi başlatın ve değişiklik taleplerini (RFC) nasıl yöneteceğinizi gösteren bir süreç oluşturun. Her adımda bir onay mekanizması kurun ve sürüm kontrol sistemi üzerinden projeyi yönetin.

2. Konfigürasyon Sabitleme (Baseline the Configuration)

Teorik Açıklama:

Konfigürasyon sabitleme, bir ürün veya sistemin belirli bir sürümünü belirleyerek, bu sürümden itibaren yapılacak tüm değişikliklerin izlenebilir olmasını sağlar. Bu, geliştirme ve değişiklik yönetimi süreçlerinin temel adımlarından biridir.

Uygulama:

- **Uygulama:** Bir GIT deposu oluşturun ve ilk yapılandırma ayarlarını sabitleyin. Bu sürümdeki tüm değişikliklerin izlenmesini sağlayacak bir yapı kurun.

3. Değişiklik Başlatma (Initiate the Change)

Teorik Açıklama:

Değişiklik talepleri (RFC), projeye eklenecek yeni özellikler veya düzeltilecek hatalar için yapılır. Bu süreç, geliştirme başlamadan önce tüm gereksinimlerin belirlendiği ve teknik toplantılarla değerlendirildiği bir adımdır.

Uygulama:

- **Uygulama:** Bir değişiklik talebi (RFC) oluşturun ve bu talebin proje ekibine nasıl iletildiğini simüle edin. Toplantılar ve teknik incelemeler üzerinden karar verildiğini gösterecek bir senaryo oluşturun.

4. Değişikliği Sınıflandırma (Classify the Change)

Teorik Açıklama:

Değişiklik talepleri, maliyet, süre ve teknik gereksinimler açısından sınıflandırılır. Eğer finansal veya teknik bir engel yoksa, ürün sahibi (product owner) bu talebin geliştirilmesi için teknik ekibe onay verir.

Uygulama:

- **Uygulama:** Bir değişiklik talebini inceleyin ve bu talebin nasıl sınıflandırıldığını, hangi koşullara göre onaylandığını gösterecek bir süreci uygulamalı olarak yönetin.

5. Değişikliği Onaylama ve Planlama (Approve and Schedule the Change)

Teorik Açıklama:

Geliştirme sürecine başlamadan önce, değişiklik talebi onaylanır ve bir proje planı oluşturulur. Bu plan, sprintler ve görev dağılımını içerir.

Uygulama:

- **Uygulama:** Bir sprint planlama toplantısı düzenleyin ve değişiklik talebine uygun olarak görevleri belirleyin. Planlama araçları (Jira, Trello vb.) kullanarak bu süreci organize edin.

6. Değişikliği Yayınlama (Release the Change)

Teorik Açıklama:

Geliştirilen değişiklik, testler tamamlandıktan sonra üretim ortamına gönderilir. Bu adımda değişikliğin başarılı bir şekilde uygulandığından emin olunur.

Uygulama:

- **Uygulama:** Geliştirilen bir değişikliği sürüm kontrol sisteminden çekin ve üretim ortamına yayınlayın. Yayın sürecindeki adımları ve test sonuçlarını kaydedin.

7. Değişikliği Doğrulama ve İnceleme (Validate and Review the Change)

Teorik Açıklama:

Değişiklik yayınlandıktan sonra, değişikliğin doğru bir şekilde uygulandığı ve beklentilere uygun olduğu doğrulanır. Teknik ve kullanıcı incelemeleri yapılır.

Uygulama:

- **Uygulama:** Yayınlanan bir değişikliği test edin ve kullanıcı geri bildirimlerini toplayın. Değişikliğin beklentilere uygun olup olmadığını kontrol edin.

Yazılım Geliştirme Ortamları ve Sürüm Kontrol Sistemleri (Software Development Environments and Source Code Version Control System)

1. Yazılım Geliştirme Ortamları (Development Environments)

Teorik Açıklama:

Yazılım geliştirme, farklı ortamlar arasında gerçekleştirilir: geliştirme, test ve üretim ortamları. Her ortam, farklı güvenlik önlemleri ve konfigürasyonlar gerektirir.

Uygulama:

- **Uygulama:** Geliştirme ve test ortamlarını kurun. Her ortam için farklı güvenlik yapılandırmalarını gösterecek bir uygulama geliştirin.

2. Sürüm Kontrol Sistemleri (Version Control Systems)

Teorik Açıklama:

Sürüm kontrol sistemleri (Git, SVN vb.), yazılım geliştirme süreçlerinin takip edilmesi ve değişikliklerin geri alınabilmesi için kullanılır. Her değişiklik kaydedilir ve versiyonlar arasında geçiş yapılabilir.

Uygulama:

- **Uygulama:** GIT kullanarak bir yazılım geliştirme sürecini yönetin. Farklı dallar (branch) arasında geçiş yapın ve bir değişikliği geri alın.

3. Geliştirme Sitesi ve Kaynak Kod Sunucu Güvenliği (Development Site and Source Code Server Security)

Teorik Açıklama:

Geliştirme ortamının fiziksel ve dijital güvenliği çok önemlidir. Kaynak kod sunucuları ve izleme sistemlerinin korunması, yazılımın bütünlüğünü sağlar.

Uygulama:

- **Uygulama:** Bir geliştirme ortamında, kaynak kod sunucusunun nasıl güvenlik altına alınacağını gösterin. Şifreleme ve erişim kontrol sistemlerini uygulamalı olarak yapıştırın.

4. Sunucu Odası ve Geliştirme Bilgisayarları Güvenliği (Development Office and Server Room Security)

Teorik Açıklama:

Sunucu odaları ve geliştirme bilgisayarları, yazılımın güvenliğini sağlamak için güvenlik önlemleriyle korunmalıdır. Erişim kontrolleri, şifreleme ve fiziksel güvenlik bu süreçlerin bir parçasıdır.

Uygulama:

- **Uygulama:** Bir sunucu odasının erişim kontrollerini simüle edin. Geliştirme bilgisayarlarında güvenlik yazılımlarını yapılandırın ve olası saldırılara karşı önlemler alın.

Haftanın Özeti ve Gelecek Hafta

Bu Hafta:

- Yazılım Geliştirme Akışı ve Değişiklik Yönetimi
- Konfigürasyon Sabitleme ve Değişiklik Onaylama
- Geliştirme Ortamları ve Sürüm Kontrol Sistemleri
- Fiziksel ve Dijital Güvenlik

Gelecek Hafta:

- Veri Güvenliği ve Kriptografi
- Güvenli İletişim ve Anahtar Yönetimi

2.Hafta – Sonu