

CEN429 GÄ¼venli Programlama Hafta-2

GeliÅtirme OrtamÄ± GÄ¼venliÄyi ve YazÄ±lÄ±m GeliÅtirme SÄ¼reÅleri

Yazar: Dr. Ä-Är. Äeyesi UÄur CORUH

İçindekiler

1 CEN429 GÄ¼venli Programlama	1
1.1 Hafta-2	1
1.1.1 Outline	2
1.2 YazÄ±lÄ±m GeliÅtirme SÄ¼reci ve DeÄyiÅlik YÄnetimi (Software Development Flow and Change Management)	2
1.2.1 1. YazÄ±lÄ±m GeliÅtirme AkÄ±Å (Flow)	2
1.2.2 2. KonfigÄ¼rasyon Sabitleme (Baseline the Configuration)	2
1.2.3 3. DeÄyiÅlik BaÅlatma (Initiate the Change)	2
1.2.4 4. DeÄyiÅlikliÄyi SÄ±nÄ±flandÄ±rma (Classify the Change)	3
1.2.5 5. DeÄyiÅlikliÄyi Onaylama ve Planlama (Approve and Schedule the Change)	3
1.2.6 6. DeÄyiÅlikliÄyi YayÄ±nlama (Release the Change)	3
1.2.7 7. DeÄyiÅlikliÄyi DoÄrülama ve Änceleme (Validate and Review the Change)	3
1.3 YazÄ±lÄ±m GeliÅtirme OrtamlarÄ± ve SÄ¼rÄ¼m Kontrol Sistemleri (Software Development Environments and Source Code Version Control System)	3
1.3.1 1. YazÄ±lÄ±m GeliÅtirme OrtamlarÄ± (Development Environments)	3
1.3.2 2. SÄ¼rÄ¼m Kontrol Sistemleri (Version Control Systems)	4
1.3.3 3. GeliÅtirme Sitesi ve Kaynak Kod Sunucu GÄ¼venliÄyi (Development Site and Source Code Server Security)	4
1.3.4 4. Sunucu OdasÄ± ve GeliÅtirme BilgisayarlarÄ± GÄ¼venliÄyi (Development Office and Server Room Security)	4
1.4 HaftanÄ±n Äzeti ve Gelecek Hafta	4
1.4.1 Bu Hafta:	4
1.4.2 Gelecek Hafta:	4

Åekil Listesi

Tablo Listesi

1 CEN429 GÄ¼venli Programlama

1.1 Hafta-2

1.1.0.1 GeliÅtirme OrtamÄ± GÄ¼venliÄyi ve YazÄ±lÄ±m GeliÅtirme SÄ¼reÅleri Ändir

- PDF¹
- DOC²

¹pandoc_cen429-week-2.pdf

²pandoc_cen429-week-2.docx

- SLIDE³
- PPTX⁴

1.1.1 Outline

- Geliştirme Ortamı Güvenli ve Yazılım Geliştirme Sürüşleri
- Yazılım Geliştirme Süreci
 - Yazılım Geliştirme Akışı
 - Konfigürasyon Sabitleme
 - Değişiklik Başılatma ve Sınanma
 - Değişiklikli Onaylama ve Yayınlama
- Yazılım Geliştirme Ortamları
 - Geliştirme Ortamı Güvenli
 - Sürüş Kontrol Sistemleri
 - Kaynak Kod Sunucu Güvenli
 - Sunucu Odası ve Geliştirme Bilgisayarları Güvenli

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

1.2.1 1. Yazılım Geliştirme Akışı (Flow)

1.2.1.1 Teorik Açıklama: Yazılım geliştirme sürüşleri, belirli akışlarla kontrol edilmelidir. Değişikliklerin doğru şekilde yönetilmesi, projenin başarıyla bir şekilde devam etmesini sağlar. Bu akış genellikle sürüş kontrol sistemleri, teknik ekipler ve proje yönetimi sürüşlerini içerir.

1.2.1.2 Uygulama:

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

1.2.2 2. Konfigürasyon Sabitleme (Baseline the Configuration)

1.2.2.1 Teorik Açıklama: Konfigürasyon sabitleme, bir sürüş veya sistemin belirli bir sürüşü belirleyerek, bu sürüşten itibaren yapılacak tüm değişikliklerin izlenebilir olmasıdır. Bu, geliştirme ve değişiklik yönetimi sürüşlerinin temel adımlarından biridir.

1.2.2.2 Uygulama:

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

1.2.3 3. Değişiklik Başılatma (Initiate the Change)

1.2.3.1 Teorik Açıklama: Değişiklik talepleri (RFC), projeye eklenecek yeni özellikler veya düzeltilen hataların yapımıdır. Bu sürüş, geliştirme başlamadan önce tüm gereksinimlerin belirlendiği ve teknik toplantılarla değerlendirildiği bir adımdır.

³cen429-week-2.pdf

⁴cen429-week-2.pptx

1.2.3.2 Uygulama:

- **Uygulama:** Bir deÄYiÄYiklik talebi (RFC) oluÄYturun ve bu talebin proje ekibine nasÄ±l iletil-diÄYini simÄ¼le edin. ToplantÄ±lar ve teknik incelemeler Ä¼zerinden karar verildiÄYini gÄ¼sterecek bir senaryo oluÄYturun.

1.2.4 4. DeÄYiÄYikliÄYi SÄ±nÄ±flandÄ±rma (Classify the Change)

1.2.4.1 Teorik AÄŞÄ±klama: DeÄYiÄYiklik talepleri, maliyet, sÄ¼re ve teknik gereksinimler aÄŞÄ±sÄ±ndan sÄ±nÄ±flandÄ±rÄ±lÄ±r. EÄYer finansal veya teknik bir engel yoksa, Ä¼rÄ¼n sahibi (product owner) bu talebin geliÄYtirilmesi iÄŞin teknik ekibe onay verir.

1.2.4.2 Uygulama:

- **Uygulama:** Bir deÄYiÄYiklik talebini inceleyin ve bu talebin nasÄ±l sÄ±nÄ±flandÄ±rÄ±ldÄ±ÄYÄ±nÄ±, hangi koÄYullara gÄ¼re onaylandÄ±ÄYÄ±nÄ± gÄ¼sterecek bir sÄ¼reci uygulamalÄ± olarak yÄ¼netin.

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

1.2.5.1 Teorik AÄŞÄ±klama: GeliÄYtirme sÄ¼recine baÄYlamadan Ä¼nce, deÄYiÄYiklik talebi onaylanÄ±r ve bir proje planÄ± oluÄYturulur. Bu plan, sprintler ve gÄ¼rev daÄYÄ±lÄ±mÄ±nÄ± iÄŞerir.

1.2.5.2 Uygulama:

- **Uygulama:** Bir sprint planlama toplantÄ±sÄ± dÄ¼zenleyin ve deÄYiÄYiklik talebine uygun olarak gÄ¼revleri belirleyin. Planlama araÄŞlarÄ± (Jira, Trello vb.) kullanarak bu sÄ¼reci organize edin.

1.2.6 6. DeÄYiÄYikliÄYi YayÄ±nlama (Release the Change)

1.2.6.1 Teorik AÄŞÄ±klama: GeliÄYtirilen deÄYiÄYiklik, testler tamamlandÄ±ktan sonra Ä¼retim ortamÄ±na gÄ¼nderilir. Bu adÄ±mda deÄYiÄYikliÄYin baÄYarÄ±lÄ± bir ÄYekilde uygu-landÄ±ÄYÄ±ndan emin olunur.

1.2.6.2 Uygulama:

- **Uygulama:** GeliÄYtirilen bir deÄYiÄYikliÄYi sÄ¼rÄ¼m kontrol sisteminden ÄŞekin ve Ä¼retim or-tamÄ±na yayÄ±nlayÄ±n. YayÄ±n sÄ¼recindeki adÄ±mlarÄ± ve test sonuÄŞlarÄ±nÄ± kaydedin.

1.2.7 7. DeÄYiÄYikliÄYi DoÄYrulama ve Ä°nceleme (Validate and Review the Change)

1.2.7.1 Teorik AÄŞÄ±klama: DeÄYiÄYiklik yayÄ±nlandÄ±ktan sonra, deÄYiÄYikliÄYin doÄYru bir ÄYekilde uygulandÄ±ÄYÄ± ve beklentilere uygun olduÄYu doÄYrulanÄ±r. Teknik ve kullanÄ±cÄ± incelemeleri yapÄ±lÄ±r.

1.2.7.2 Uygulama:

- **Uygulama:** YayÄ±nlanan bir deÄYiÄYikliÄYi test edin ve kullanÄ±cÄ± geri bildirimlerini top-layÄ±n. DeÄYiÄYikliÄYin beklentilere uygun olup olmadÄ±ÄYÄ±nÄ± kontrol edin.

1.3 YazÄ±lÄ±m GeliÄYtirme OrtamlarÄ± ve SÄ¼rÄ¼m Kontrol Sistemleri (Software Development Environments and Source Code Version Control System)

1.3.1 1. YazÄ±lÄ±m GeliÄYtirme OrtamlarÄ± (Development Environments)

1.3.1.1 Teorik AÄŞÄ±klama: YazÄ±lÄ±m geliÄYtirme, farklı ortamlar arasÄ±nda gerÄŞikleÄYtirilir: geliÄYtirme, test ve Ä¼retim ortamlarÄ±. Her ortam, farklı gÄ¼venlik Ä¼nlemleri ve konfigÄ¼rasyonlar gerektirir.

1.3.1.2 Uygulama:

- **Uygulama:** Geliştirme ve test ortamlarını kurun. Her ortam için farklı güvenlik yapılandırmaları oluşturarak güvenli bir uygulama geliştirin.

1.3.2 2. Sürüm Kontrol Sistemleri (Version Control Systems)

1.3.2.1 Teorik Açıklama: Sürüm kontrol sistemleri (Git, SVN vb.), yazılım geliştirme süreçlerinin takip edilmesi ve değişikliklerin geri alınabilmesi için kullanılır. Her değişikliğin kaydedildiği ve versiyonlar arasında geçiş yapılabilir.

1.3.2.2 Uygulama:

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

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

1.3.3.1 Teorik Açıklama: Geliştirme ortamını fiziksel ve dijital güvenliğini sağlamak önemlidir. Kaynak kod sunucularını ve izleme sistemlerinin korunması, yazılımın güvenliğini bütünlüğünü sağlar.

1.3.3.2 Uygulama:

- **Uygulama:** Bir geliştirme ortamında, kaynak kod sunucusunun güvenliğini altın alana kadar güncelleyin. İzleme ve erişim kontrol sistemlerini uygulamaları olarak yapılandırın.

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

1.3.4.1 Teorik Açıklama: Sunucu odaları ve geliştirme bilgisayarları, yazılımın güvenliğini sağlamak için fiziksel güvenliğini sağlamak önemlidir. Erişim kontrolleri, izleme ve fiziksel güvenlik bu süreçlerin bir parçasıdır.

1.3.4.2 Uygulama:

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

1.4 Haftanın Özeti ve Gelecek Hafta

1.4.1 Bu Hafta:

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

1.4.2 Gelecek Hafta:

- Veri Güvenliği ve Kriptografi
- Güvenli İletişim ve Anahtar Yönetimi