

CEN429 G venli Programlama Hafta-2

Geli tirme Ortam  G venli i ve Yaz l m Geli tirme S re  leri

Yazar: Dr.   r.   yesi U  ur CORUH

  indekiler

1 CEN429 G�venli Programlama	1
1.1 Hafta-2	1
1.1.1 Outline	2
1.2 Yaz�l�m Geli�tirme S�reci ve De�i�iklik Y�netimi (Software Development Flow and Change Management)	2
1.2.1 1. Yaz�l�m Geli�tirme Ak���� (Flow)	2
1.2.2 2. Konfig�rasyon Sabitleme (Baseline the Configuration)	2
1.2.3 3. De�i�iklik Ba�latma (Initiate the Change)	2
1.2.4 4. De�i�ikli�i S�n�fland�rma (Classify the Change)	3
1.2.5 5. De�i�ikli�i Onaylama ve Planlama (Approve and Schedule the Change)	3
1.2.6 6. De�i�ikli�i Yay�nlama (Release the Change)	3
1.2.7 7. De�i�ikli�i Do�rulama ve �nceleme (Validate and Review the Change)	3
1.3 Yaz�l�m Geli�tirme Ortamlar� ve S�r�m Kontrol Sistemleri (Software Development Environments and Source Code Version Control System)	3
1.3.1 1. Yaz�l�m Geli�tirme Ortamlar� (Development Environments)	3
1.3.2 2. S�r�m Kontrol Sistemleri (Version Control Systems)	4
1.3.3 3. Geli�tirme Sitesi ve Kaynak Kod Sunucu G�venli�i (Development Site and Source Code Server Security)	4
1.3.4 4. Sunucu Odas� ve Geli�tirme Bilgisayarlar� G�venli�i (Development Office and Server Room Security)	4
1.4 Haftan�n �zeti ve Gelecek Hafta	4
1.4.1 Bu Hafta:	4
1.4.2 Gelecek Hafta:	4

 ekil Listesi

Tablo Listesi

1 CEN429 G venli Programlama

1.1 Hafta-2

1.1.0.1 Geli tirme Ortam  G venli i ve Yaz l m Geli tirme S re  leri  ndir

- PDF¹
- DOC²

¹pandoc_cen429-week-2.tr_doc.pdf

²pandoc_cen429-week-2.tr_word.docx

- SLIDE³
- PPTX⁴

1.1.1 Outline

- Geliştirme Ortamı Geliştirme ve Yazılım Geliştirme Süreçleri
- Yazılım Geliştirme Süreci
 - Yazılım Geliştirme Akışı
 - Konfigürasyon Sabitleme
 - Değişiklik Baskılatma ve Sınırlama
 - Değişiklikli Onaylama ve Yayınlama
- Yazılım Geliştirme Ortamları
 - Geliştirme Ortamı Geliştirme
 - Süreç Kontrol Sistemleri
 - Kaynak Kod Sunucu Geliştirme
 - Sunucu Odası ve Geliştirme Bilgisayarları Geliştirme

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 süreçleri, belirli akışlarla kontrol edilmelidir. Değişikliklerin doğru şekilde yönetilmesi, projenin başarıyla bir şekilde devam etmesini sağlar. Bu akış genellikle süreç kontrol sistemleri, teknik ekipler ve proje yönetimi süreçlerini içerir.

1.2.1.2 Uygulama:

- **Uygulama:** Basit bir yazılım projesi baskılatma ve değişiklik taleplerini (RFC) nasıl yönetileceğini gösteren bir süreç oluşturulur. Her adımda bir onay mekanizması kurulur ve süreç kontrol sistemi üzerinden projeyi yönetin.

1.2.2 2. Konfigürasyon Sabitleme (Baseline the Configuration)

1.2.2.1 Teorik Açıklama: Konfigürasyon sabitleme, bir süreç veya sistemin belirli bir süreçten itibaren yapılacak tüm değişikliklerin izlenebilir olmasıdır. Bu, geliştirme ve değişiklik yönetimi süreçlerinin temel adımlarından biridir.

1.2.2.2 Uygulama:

- **Uygulama:** Bir GIT deposu oluşturulur ve ilk yapılandırma ayarları sabitleyin. Bu süreçteki tüm değişikliklerin izlenmesini sağlayacak bir yapı kurulur.

1.2.3 3. Değişiklik Baskılatma (Initiate the Change)

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

³[cen429-week-2.tr_slide.pdf](#)

⁴[cen429-week-2.tr_slide.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 gÄ¶sterecek 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 gÄ¶re onaylandÄ±ÄYÄ±nÄ± gÄ¶sterecek bir sÄ¼reci uygulamalÄ± olarak yÄ¶netin.

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

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

1.2.5.2 Uygulama:

- **Uygulama:** Bir sprint planlama toplantÄ±sÄ± dÄ¼zenleyin ve deÄYiÄYiklik talebine uygun olarak gÄ¶revleri 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 gÄ¶nderilir. Bu adÄ±mda deÄYiÄYikliÄYin baÄYarÄ±lÄ± bir ÄYekilde uygulandÄ±ÄYÄ±ndan emin olunur.

1.2.6.2 Uygulama:

- **Uygulama:** GeliÄYtirilen bir deÄYiÄYikliÄYi sÄ¼rÄ¼m kontrol sisteminden ÄŞekin ve Ä¼retim ortamÄ±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 toplayÄ±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, farklÄ± ortamlar arasÄ±nda gerÄŞekleÄYtirilir: geliÄYtirme, test ve Ä¼retim ortamlarÄ±. Her ortam, farklÄ± gÄ¼venlik Ä¶nlemleri ve konfigÄ¼rasyonlar gerektirir.

1.3.1.2 Uygulama:

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

1.3.2 2. Sürüm Kontrol Sistemleri (Version Control Systems)

1.3.2.1 Teorik Aşıklama: 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.

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 Aşıklama: Geliştirme ortamının fiziksel ve dijital güvenliğini sağlamak önemlidir. Kaynak kod sunucularını ve izleme sistemlerinin korunması, yazılımın güvenliğini bütünlüğü sağlar.

1.3.3.2 Uygulama:

- **Uygulama:** Bir geliştirme ortamında, kaynak kod sunucusunun güvenliğini altın alacakları için 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 Aşıklama: Sunucu odaları ve geliştirme bilgisayarları, yazılımın güvenliğini sağlamak için güvenliğini sağlamak için güvenlik önlemleriyle korunmalıdır. Erişim kontrolleri, izleme ve fiziksel güvenlik bu süreçlerin bir parçasıdır.

1.3.4.2 Uygulama:

- **Uygulama:** Bir sunucu odasının erişim kontrollerini simüle edin. Geliştirme bilgisayarlarında güvenlik yazılımları yapılındır 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