



YMT 312-Yazılım Tasarım Ve Mimarisi Doğrulama ve Geçerleme

Fırat Üniversitesi Yazılım Mühendisliği Bölümü

Bölüm-7

Bu Haftaki Konular

Doğrulama Ve Geçerleme.....5

Sinama Kavramları.....21

Doğrulama ve Geçerleme Yaşam Döngüsü.....27

Sinama Yöntemleri.....29

Sinama ve Bütünleştirme Stratejileri.....34

Sinama Planlaması.....47

Sinama Belirtileri.....48

Yaşam Döngüsü Boyunca Sinama Etkinlikleri.....50

Amaçlar

- Doğrulama ve Geçerleme Kavramlarını Öğrenmek
- Doğrulama Tekniklerini Öğrenmek
- Sınama Kavramları Hakkında Bilgi Edinmek
- Doğrulama ve Geçerleme Yaşam Döngüsünü Kavramak
- Bütünleştirme Sınamasında "koçan" Kullanımı
- Yazılım Yaşam Döngüsü Boyunca Sınama Etkinliklerini Öğrenmek



Giriş

- Geliştirilecek bilgi sistemi yazılımının doğrulanması ve geçerlenmesi, üretim süreci boyunca süren etkinliklerden oluşur. Söz konusu etkinlikler:
 - Yazılım belirtilmelerinin ve proje yaşam sürecindeki her bir etkinlik sonunda alınan çıktıların, tamam, doğru, açık ve önceki belirtilmeleri tutarlı olarak betimler durumunda olduğunun doğrulanması.
 - Proje süresince her bir etkinlik ürününün teknik yeterliliğinin değerlendirilmesi ve uygun çözüm elde edilene kadar aktivitenin tekrarına sebep olması.
 - Projenin bir aşaması süresince geliştirilen anahtar belirtilmelerin önceki belirtilmelerle karşılaştırılması.
 - Yazılım ürünlerinin tüm uygulanabilir gerekleri sağladığının gerçekleşmesi için sınamaların hazırlanıp yürütülmesi.

biçiminde özetlenebilir.

Doğrulama ve Geçerleme

Doğrulama

Doğru ürünü mü üretiyoruz ?

Ürünü kullanacak kişilerin isteklerinin karşılanıp karşılanmadığına dair etkinliklerden oluşur.

Geçerleme

Ürünü doğru olarak mı üretiyoruz?

Ürünün içsel niteliğine ilişkin izleme ve denetim etkinliklerinden oluşur.

Doğrulama ve geçerleme işlemleri temel olarak çeşitli düzeylerde sına, gözden geçirme, denetim ve hata giderme süreçlerinden oluşur.

Bağımsız Doğrulama ve Geçerleme

Tüm D&G işlemleri bir başka şirket tarafından yapılır. Kurum içi başka bir bölüm/grup tarafından da yapılabilir.

Avantajları

- Konusunun uzmanları tarafından yapılır
- Kurum içi siyasetten etkilenmez

Dezavantajları

- Daha pahalıdır
- Daha sıkı koordinasyon gerektirir

Doğrulama Teknikleri

➤ Gözden Geçirme

- Yönetim
- Teknik
- Arkadaş
- Masaüstü

➤ Üstünden Geçme

➤ Denetleme

➤ İnceleme

Gözden Geçirme - Yönetim

Amaç: [IEEE1028-97]

- Durum izlemek
- Planların ve takvimin durumlarını belirlemek
- Gereksinimleri ve sistem kaynaklarını onaylamak
- Yönetim şeklinin hedefe uygunluğunu değerlendirmek

Yöntem:

- Proje Yöneticisi veya Lideri tarafından yöneticilere sunum

Katılımcılar:

- Karar verme yetkisi olan yöneticiler

Neler:

Raporlar (Denetleme, Durum, Sonuçlar, ...)

Planlar (Risk, Proje, Konfigürasyon, ...)

Beklenen Sonuçlar:

Yapılması planlanan değişikliklere ve iyileştirme kararlarına destek ve onay almak

Plan, takvim, ve gereksinimlerin yeterliliğini belirlemek

Projenin yolunda gidip gitmediğini belirlemek

Gözden Geçirme - Teknik

Amaç: [IEEE1028-97]

- Ürünün kullanıma uygunluğunu değerlendirmek
- Ürünün onaylanmış gereksinimlere uymayan yanlarını belirlemek

Yöntem:

- Teknik Lider tarafından sunum

Katılımcılar:

- Karar verme yetkisi olan yöneticiler,
- Gözden geçirme sorumlusu,
- Kaydedici,
- Teknik Uzmanlar

Neler:

Ürüne ait:

- Amaç ve hedefler
- Proje yönetim planı
- Problem listesi

Beklenen Sonuçlar:

- Ürünün beklentilere ve standartlara uygun olup olmadığını yöneticilere göstermek
- Değişiklikleri kontrol etmek,
- Devam edip etmeme kararını almak

Gözden Geçirme - Arkadaş

Amaç:

- Ürünün kullanıma uygunluğunu değerlendirmek
- Ürünün onaylanmış gereksinimlere uymayan yanlarını belirlemek.

Yöntem:

- Ürün konuya hakim bir takım arkadaşına verilir ve bulunan hatalar/düzeltilmeler ürün sahibine açıklanır

Katılımcılar:

- Konusunda uzman takım elemanı
- Ürün sahibi

Neler:

- Gereksinimler, tasarım dokümantasyonu, kaynak kodu
- Planlar (Proje, Yazılım Geliştirme, Test, ...)

Beklenen Sonuçlar:

- Bulunan hatalar/düzeltilmeler/tavsiyeler belgenin üzerine yazılır ve yazara açıklanır

Gözden Geçirme – Masa Üstü

Amaç:

- Ürünün kullanıma uygunluğunu değerlendirmek
- Ürünün onaylanmış gereksinimlere uymayan yanlarını belirlemek.

Yöntem:

- Ürün konuya hakim uzmanlara dağıtılır ve bulunan hatalar/düzeltilmeler ürünün bir kopyası üzerine kaydedilir

Katılımcılar:

- Uzmanlar
- Belgenin müşterileri
- Belgenin sahibi

Neler:

- Gereksinimler, tasarım dokümantasyonu, kaynak kodu
- Planlar (Proje, Yazılım Geliştirme, Test, ...)

Beklenen Sonuçlar:

- Bulunan hatalar/düzeltilmeler/tavsiyeler belgenin üzerine yazılır ve yazara açıklanır

Üstünden Geçme

Amaç: [IEEE1028-97]

- Ara veya son ürünü değerlendirmek,
- Katılımcıları eğitmek

Yöntem:

- Konu ile ilgili proje elemanlarına sunum

Katılımcılar:

- Yazar (Sunucu)
- Konusunda uzmanlar
- Ürünün müşterileri

Neler:

Sistem Gereksinim Belirtileri (SRS)
Tasarım
Proje Planı

Beklenen Sonuçlar:

Bulgu listesi
İyileştirme ve alternatif tavsiyeleri

Denetleme

Amaç: [IEEE1028-97]

- Ürünün ve süreçlerin bağımsız uzmanlar tarafından değerlendirilip regülasyonlara, standartlara, kılavuzlara, planlara ve protokollere uygunluğunun belirlenmesi.

Yöntem:

- Bağımsız bir uzman takımı tarafından değerlendirme

Katılımcılar:

- Bağımsız uzmanlar

Neler:

- Gereksinimler, Sistem Mimarisi, Tasarımlar
- Proje Planı

Beklenen Sonuçlar:

- Bulgu listesi
- İyileştirme ve alternatif tavsiyeleri

İnceleme

Amaç: [IEEE1028-97]

- Üründeki hata ve anormallikleri belirlemek.

Yöntem:

- İnceleme toplantısı düzenlenir
- Katılımcılar toplantıdan önce ürünü detaylı olarak değerlendirir
- Toplantı sırasında ürün satır satır okunarak bulgular tartışılır

Katılımcılar : (Yöneticiler katılmaz)

- İnceleme sorumlusu (Uzlaştırıcı)
- Yazar,
- Okuyucu,
- Kayıtcı,
- İnceleyiciler
 - Tekrar incelenmesi gerekir

Neler:

Gereksinimler, tasarım ve kaynak kod

Beklenen Sonuçlar:

Hata listesi

Kabul Durumu:

Olduğu gibi veya önemi düşük düzeltmelerle kabul
Önemli değişikliklerle kabul

İncelemenin Önemi

- İnceleme maliyeti artırır. Faydası nedir?
- Sonraki aşamalarda, özellikle Test ve Müşteri tarafından bulunan hataların onarılması maliyeti daha fazla etkiler
- Bu hataları giderirken, başka hataların sisteme girme olasılığı artar
- Test ve Müşteri tarafından hataların düzeltilmesi proje süresini artırır
- Müşterinin ürün ve firma hakkındaki fikirleri zedelenir
- Proje elemanlarının işlerinden tatmini azalır

İnceleme Yapmayan Bir Proje

Proje büyüklüğü aşağıdaki rakamlarla verilmiştir:

- Gereksinimler : 300 sayfa
- Tasarım : 150 sayfa
- Kaynak Kodu : 10 000 satır

Varsayımlar:

- Gereksinim, Tasarım ve Kodlama aşamaları 100'er hata yapıyor
- Tüm hatalar Test ve Müşteri tarafından bulunuyor
- Testin Hata Bulma Etkinliği (HBE) %75 ve geriye kalan hataların hepsi müşteri tarafından bulunuyor
- Test tarafından bulunan hataların onarılması Gereksinim, Tasarım ve Kodlama hataları için sırasıyla 3, 2.5, ve 2 gündür
- Müşteri tarafından bulunan hataların onarılması Gereksinim, Tasarım ve Kodlama hataları için sırasıyla 4, 3.5, ve 3 gündür

İnceleme Yapmayan Bir Proje (Devam)

Aşama	Hata	Tst	Mlyt	Mşt	Mlyt	Top Mlyt
Gerek	100	75	225	25	100	325
Tasar	100	75	188	25	88	275
Kod	100	75	150	25	75	225
Toplam	300	225	563	75	263	825

Tüm Hataların Onarılma Maliyeti

825 adam-gün

Aynı Proje – Gereksinimler İncelenirse

İnceleme için varsayımlar:

- Gereksinim İncelemesi hızı 15 sayfa/saat
- İnceleme Takımı 5 kişi
- İncelemenin HBE %75
- İncelemede bulunan hataların onarılma süresi ½ adam-gün
- İş Günü 8 saat

İncelemenin maliyeti

$$(300/15) \times 5 = 100 \text{ saat} = 12.5 \text{ gün}$$

İncelemede bulunan hataları onarma süresi

$$75 \times 0.5 = 37.5 \text{ gün}$$

Toplam

$$12.5 + 37.5 = \underline{\underline{50 \text{ gün}}}$$

Aynı Proje – Gereksinimler İncelenirse (Devam)

Test ve Müşteri Tarafından Bulunan Hataların Onarılma Maliyeti

Aşama	Hata	Tst	Mlyt	Mşt	Mlyt	Top Mlyt
Gerek	25	19	57	6	24	81
Tasar	100	75	188	25	88	275
Kod	100	75	150	25	75	225
Toplam	225	169	395	56	187	581

Tüm Hataların Toplam Maliyeti

581+50 = 631 adam-gün

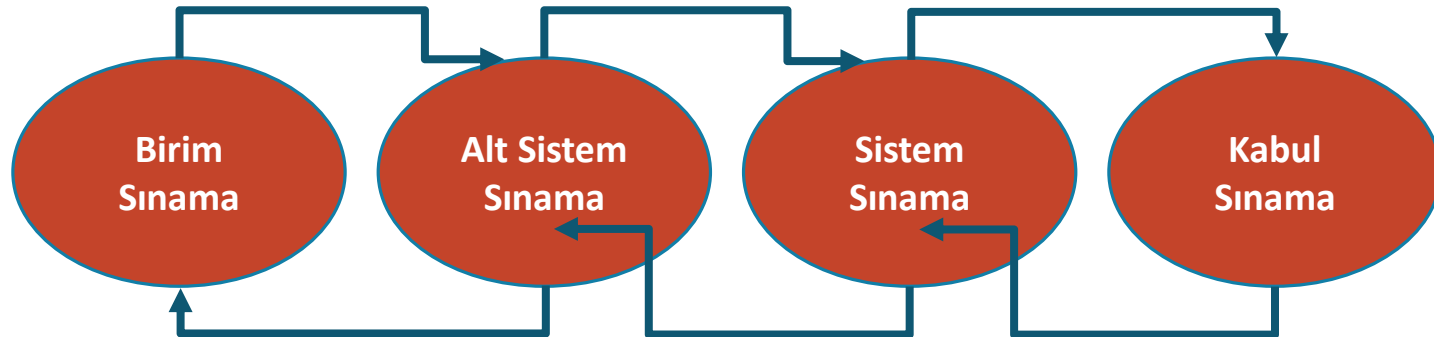
İncelemenin faydaları

- Hataların Onarılması:
 - İncelemesiz : 825 adam-gün
 - İnceleme ile : 631 adam-gün
 - Kazanılan zaman : 194 adam-gün
 - 39 adam-hafta
 - 9 adam-ay
- Maliyet azaldı
- Takvim kısaldı
- Kalite arttı
- Müşteri memnun (Neden?)
- Proje elemanları memnun (Neden?)
- Yöneticiler memnun (Neden?)

Sinama Kavramları

- Sinama ve bütünleştirme işlemlerinin bir strateji içinde gerçekleştirilmesi, planlanması ve tekniklerin seçimi gerekmektedir. Bütünleştirme işlemi, en küçük birimlerden başlanarak sistem düzeyine çıkılmaktadır.
- Bu değişik düzeylere hitap edecek sinama yöntemleri olmalıdır.
- Sınamalar, hatalardan kurtulmanın bir güvencesi değildir.
- Hatalardan bütünüyle arınıldığı gibi bir kanı edinilmemelidir.
- Yalnızca, sinamalar uzadıkça hata bulma sıklığı azalır, daha zor bulunacak hatalar gizli kalmağa devam eder.
- Ne kadar hata sıklığına erişildiğinde sinama işleminin durdurulacağı, maliyet ve kalite arasında bir en iyileştirme yapma gereğini öne çıkarır.
- Aynı zamanda vakit de önemli bir unsurdur.
- Daha uzunca süreler vakitler harcanarak daha az hatalar bulunmaya devam eder.
- Yazılımın kritiklik düzeyine göre, sinamaya ayrılan süre ve çaba artar.

Sinama Kavramlari



Birim Sınama

- Bağlı oldukları diğer sistem unsurlarından bütünüyle soyutlanmış olarak birimlerin doğru çalışmalarının belirlenmesi amacıyla yapılır.
- Birimler, ilişkili yapıtaşlarının bütünleştirilmesinden oluşurlar.

Alt Sistem Sınama

- Alt sistemler, modüllerin bütünleşmesi ile ortaya çıkar. Yine bağımsız olarak sınamaları yapılmalıdır.
- Bu düzeyde en çok hata arayüzlerde bulunmaktadır, arayüz hatalarına yönelik sınamalara yoğunlaşılmalıdır.

Sistem Sınama

- Üst düzey bileşenlerin sistem ile olan etkileşimlerinde çıkacak hatalar aranmaktadır.
- Ayrıca belirtilen ihtiyaçların doğru yorumlandıkları da sınanmalıdır.

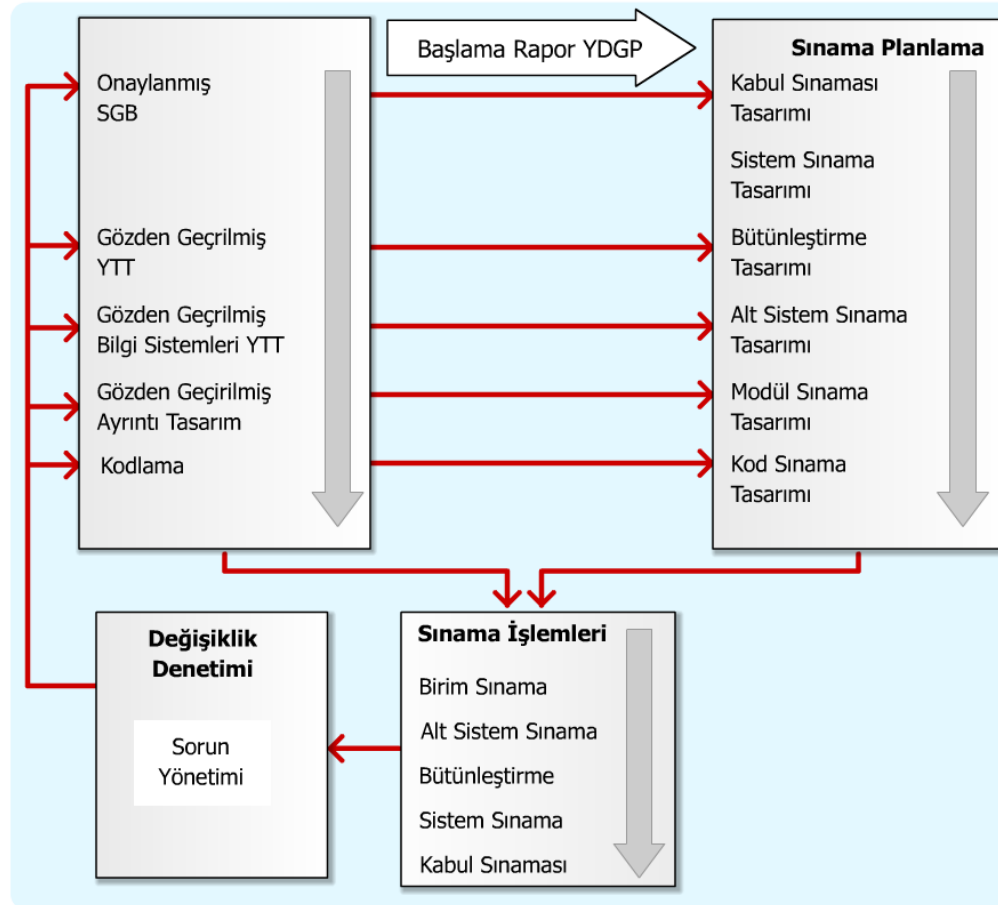
Kabul Sınama

- Çalıştırılmadan önce sistemin son sınavasıdır. Artık yapay veri yerine gerçek veriler kullanılır.
- Bu sınama türü alfa sınavası ve beta sınavası olarak ta bilinir.
- Alfa sınamada, tanımında, sinamanın geliştirici organizasyonun yerleşkesinde, kullanıcıların da gelerek katkıda bulunması içerilir.
- Daha sonra ürünün pazarlama işlemi sırasında beta sınama denilen, sınama kullanıcının kendi yerleşkesinde geliştirici gözetiminde yapılır.

Doğrulama ve Geçerleme Yaşam Döngüsü

- Doğrulama ve geçerleme işlemleri yazılım üretim yaşam döngüsünün tüm süreçlerinde ve bu süreçlere koşut olarak sürer.
- Gerçekleştirim aşamasına kadar olan süreçlerde doğrulama ve geçerleme işlemlerinin planlaması yapılır.
- Planlama, genel olarak, birim, alt sistem, bütünleştirme, sistem ve kabul sınamalarının tasarımlarını içerir. Gerçekleştirim aşamasının sonunda ise söz konusu planlar uygulanır.
- Uygulama sonucu elde edilen bulgular, Yazılım Doğrulama ve Geçerleme raporları biçiminde sürekli olarak raporlanır.
- Bu bilgiler, değişiklik denetim sistemi ve sorun yönetim sistemlerinde girdi olarak kullanılır.
- Değişiklik Denetim sistemi, sınama süresince elde edilen bulguların izlenmesi amacıyla oluşturulan bir sistemdir.
- Bu sistemde, sınama sonucu elde edilen bulgular ve bunlara karşı sorun yönetimi tarafından alınan önlemler izlenir.

Doğrulama ve Geçerleme Yaşam Döngüsü



Beyaz Kutu Sınaması

Beyaz kutu sınaması tasarlanırken, birimin süreç belirtiminden yararlanır.

Yapılabilecek denetimler arasında:

- Bütün bağımsız yolların en azından bir kere sınanması,
- Bütün mantıksal karar noktalarında iki değişik karar için sınamaların yapılması,
- Bütün döngülerin sınır değerlerinde sınanması,
- İç veri yapılarının denenmesi

bulunur.

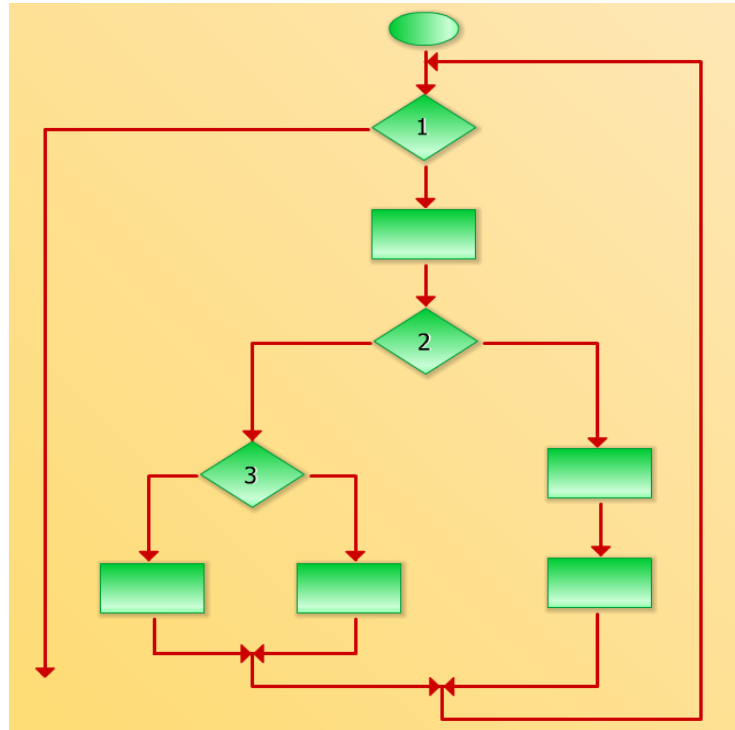
Bunun için hataların bazı özelliklerinin bilinmesinde yarar vardır:

- Bir program kesiminin uygulamada çalıştırılma olasılığı az ise o kesimde hata olması ve bu hatanın önemli olması olasılığı fazladır.
- Çoğu zaman, kullanılma olasılığı çok az olarak kestirilen program yolları, aslında çok sıkça çalıştırılıyor olacaktır.
- Yazım hataları rasgele olarak dağılır. Bunlardan bazılarını derleyiciler bulur, bazıları da bulunmadan kalır.

Temel Yollar Sınaması

- Daha önce çevrimsellik karmaşıklığı konusunda gördüğümüz hesap yöntemi ile bir programdaki bağımsız yollar bulunduktan sonra, bu kadar sayıda sınama yaparak programın her birimini bir şekilde sınamalara dahil etmiş oluruz.
- Bağımsız yolların saptanması için önce, program çizgesel bir biçime çevrilir.
- Bunu yapmak için ise, program iş akış şemaları diyagramları iyi bir başlangıç noktasıdır.

Temel Yollar Sınaması



Bir Programın İş Akış Şeması

Temel Yollar Sınaması

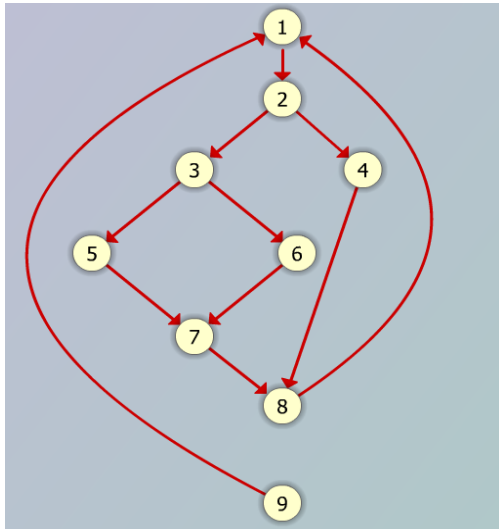
- Program akış diyagramları matematiksel titizlik ile tanımlanmayan daha serbestçe program yapılarını alt düzeyde modelleyen çizimlerdir. Akış diyagramları ise Çizge Teorisi dalında kabul görecektir bir "Çizge"dir.
- Her çizgede olduğu gibi burada da düğümler ve dallar (veya kırımlar) bulunur.
- Program akış diyagramından akış diyagramına geçmek için süreç kutuları ortadan kaldırılır, koşul baklavaları yerine düğümler çizilir ve her koşul düğümlerine karşı düşecek birleştirme düğümleri eklenir.
- Birleştirme düğümleri, koşul kollarının kapandığı noktaya konur.
- Artık bu çizge üzerinden temel yol sayısını da verecek olan çevrimsellik karmaşıklığı sayısını hesaplayabiliriz:

$$E - N + 2$$

E: toplam dal sayısı
N: toplam düğüm sayısı

- Bağımsız temel yol sayısı kadar temel yolları çizge üzerinde (sonunda programa yansıtılmak üzere) veya program üzerinde işaretlenir. Sonra bu yolların hepsinin oluşturulacağı sınav programları tasarlanır.

Temel Yollar Sınaması



Programın Grafik Diyagramı

Formüle göre bağımsız yol sayısı: $11 - 9 + 2 = 4$

Bunun anlamı da şöyle açıklanabilir: Çizgedeki her dal sınaama işleminde kapsanmalıdır. Bu, sınaama sırasında her işlemin çalıştırılması demektir. Her dalın en az bir kere kapsanacağı ve en az sayıda yollar bulunmalıdır. Programa ortadan giremeyeceğimize göre de bu yollar başlangıç noktasından bitiş noktasına kadar uzanmalıdır. Son olarak da minimum sayıda yol ile bu şartları sağlamalıyız. Bu sayının 4 olduğu, daha önce yukarıdaki formülde hesaplanmıştı.

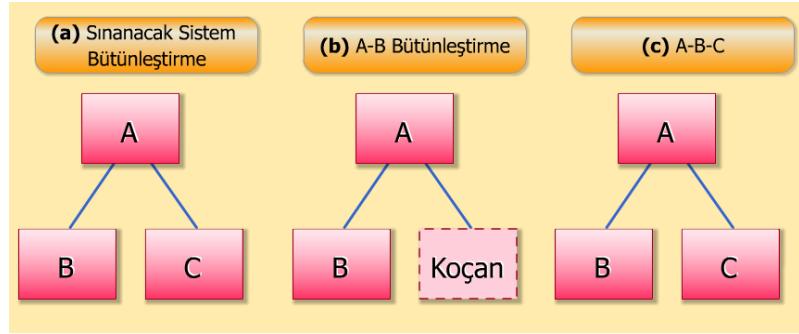
Sinama ve Bütünleştirme Stratejileri

- Genellikle sinama stratejisi, bütünleştirme stratejisi ile birlikte değerlendirilir. Ancak bazı sinama stratejileri bütünleştirme dışındaki tasaları hedefleyebilir.
- Örneğin, yukarıdan aşağı ve aşağıdan yukarı stratejileri bütünleştirme yöntemine bağlıdır.
- Ancak işlem yolu ve gerilim sinamaları, sistemin olaylar karşısında değişik işlem sıralandırmaları sonucunda ulaşacağı sonuçların doğruluğunu ve normal şartların üstünde zorlandığında dayanıklılık sınırını ortaya çıkarır.

Yukarıdan Aşağı Sınama ve Bütünleştirme

- Yukarıdan aşağı bütünleştirmede, önce sistemin en üst düzeylerinin sınanması ve sonra aşağıya doğru olan düzeyleri, ilgili modüllerin takılarak sınanmaları söz konusudur.
- En üst noktadaki bileşen, bir birim/modül/alt sistem olarak sınıandıktan sonra alt düzeye geçilmelidir.
- Ancak bu en üstteki bileşenin tam olarak sınanması için alttaki bileşenlerle olan bağlantılarının da çalışması gerekir.
- Alt bileşenler ise bu stratejiye göre henüz hazırlanmış olamazlar.
- Bunların yerine üst bileşenin sınaması için kullanılmak üzere 'koçan' programları yazılır.
- Koçanlar, bir alt bileşenin, üst bileşen ile ara yüzünü temin eden, fakat işlevsel olarak hiç bir şey yapmayan, boş çerçeve programlarıdır.
- Üst bileşenin sınanması bittikten sonra bu koçanlar, içleri doldurularak kendi kodlama ve birim sınama işlemlerini tamamladıktan sonra üst bileşen ile yeniden sınanırlar.

Bütünleştirme Sınamasında "koçan" Kullanımı



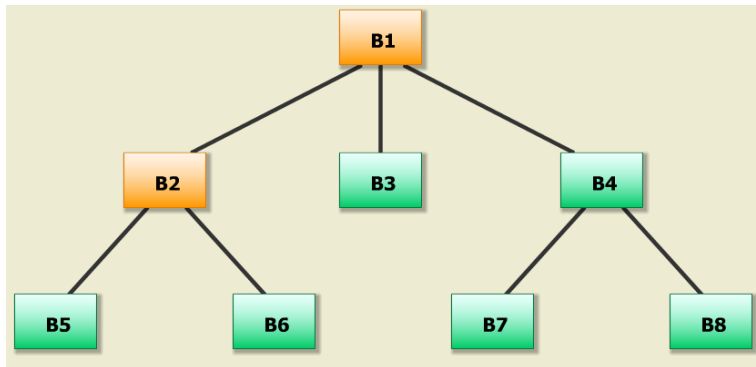
A, B, C birimlerinden oluşan ve birim şeması (a) şıkkında belirtilen bir sistemin bu tür koçan kullanılarak sınanması (b) şıkkı ve şekil (c) şıkkında belirtilmektedir.

İlk adımda A ve B birimleri bütünleştirilir; C için bir "koçan" yazılır. İkinci adımda ise "koçan" kaldırılır ve C ile yer değiştirilerek A-B-C bütünleştirilir.

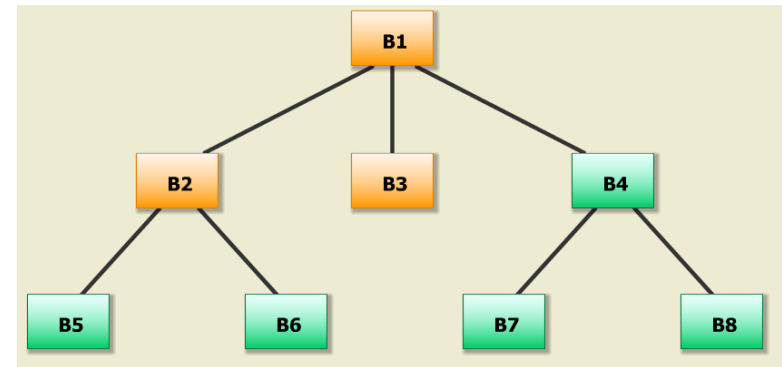
Yukarıdan aşağıya doğru bütünleştirme işleminde iki yaklaşım izlenebilir:

- 1. Yaklaşım:** Düzey Öncelikli Bütünleştirme (En üst düzeyden başlanır, öncelikle aynı düzeylerdeki birimler bütünleştirilir.)
- 2. Yaklaşım:** Derinlik Öncelikli Bütünleştirme (En üst düzeyden başlanır. Birim şemasında bulunan her dal soldan sağa olma üzere ele alınır. Bir dala ilişkin bütünleştirme bitirildiğinde diğer dalın bütünleştirilmesi başlar.)

1. Yaklaşım: Düzey Öncelikli Bütünleştirme

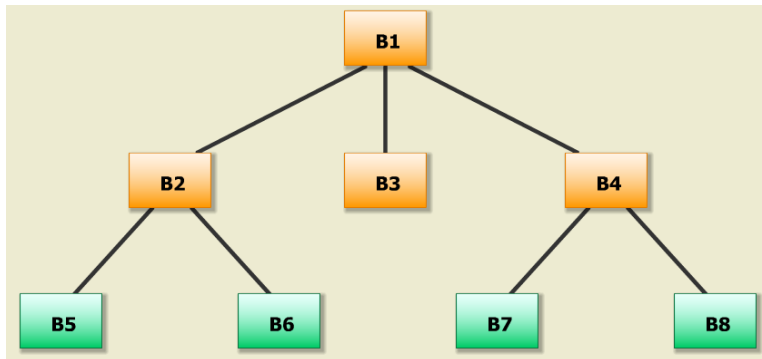


1. Adım: B1-B2 (KoçanB3 - KoçanB4- KoçanB5 - KoçanB6)

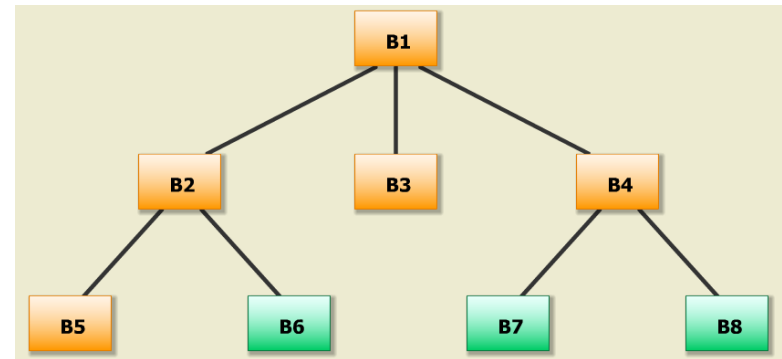


2. Adım: B1-B2-B3 (KoçanB4- KoçanB5 - KoçanB6)

1. Yaklaşım: Düzey Öncelikli Bütünleştirme

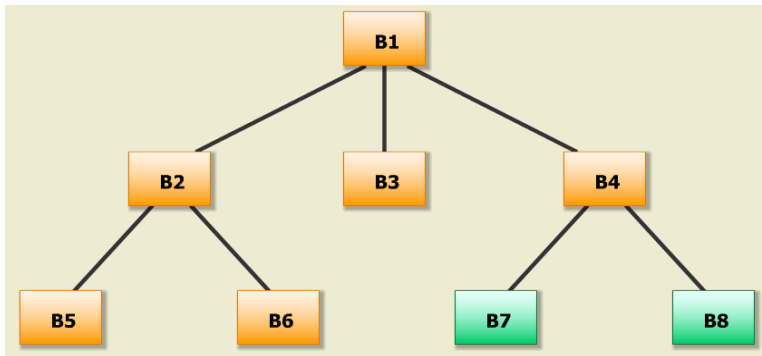


3. Adım: B1-B2-B3-B4 (KoçanB7 - KoçanB8- KoçanB5 - KoçanB6)

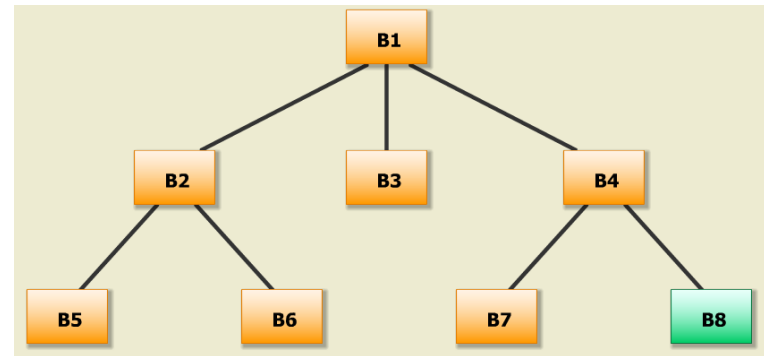


4. Adım: B1-B2-B3-B4-B5 (KoçanB7 - KoçanB8- KoçanB6)

1. Yaklaşım: Düzey Öncelikli Bütünleştirme

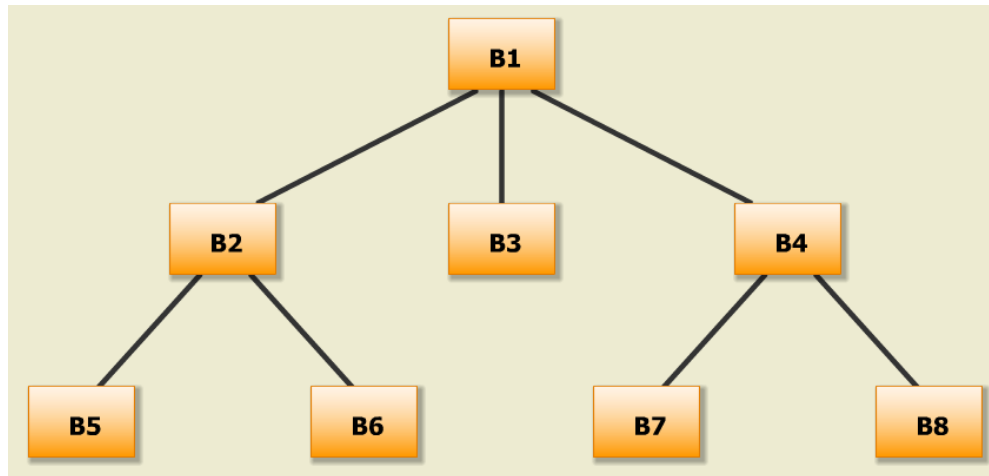


5. Adım: B1-B2-B3-B4-B5-B6 (KoçanB7 - KoçanB8)



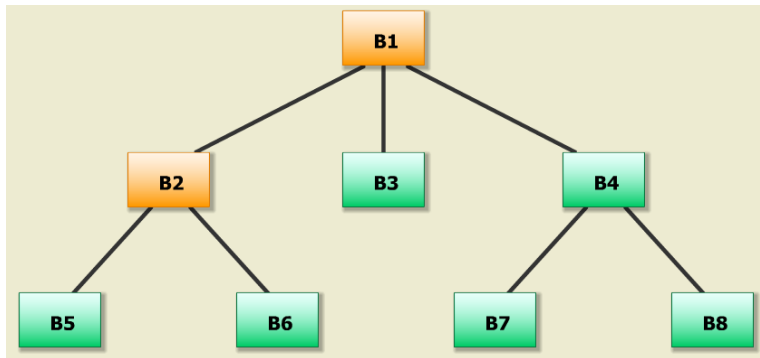
6. Adım: B1-B2-B3-B4-B5-B6-B7 (KoçanB8)

1. Yaklaşım: Düzey Öncelikli Bütünleştirme

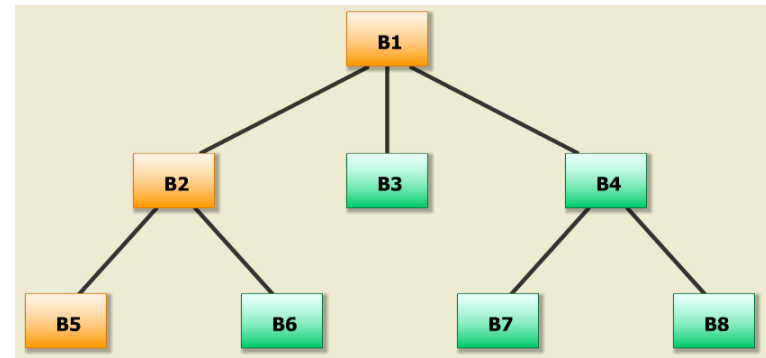


7. Adım: B1-B2-B3-B4-B5-B6-B7-B8

2. Yaklaşım: Derinlik Öncelikli Bütünleştirme

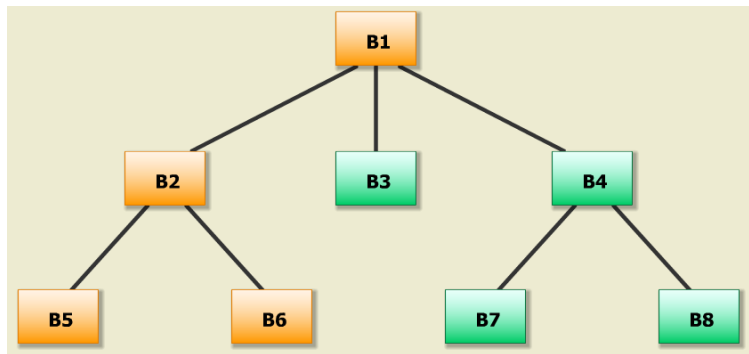


1. Adım: B1-B2 (KoçanB3 - KoçanB4- KoçanB5 - KoçanB6)

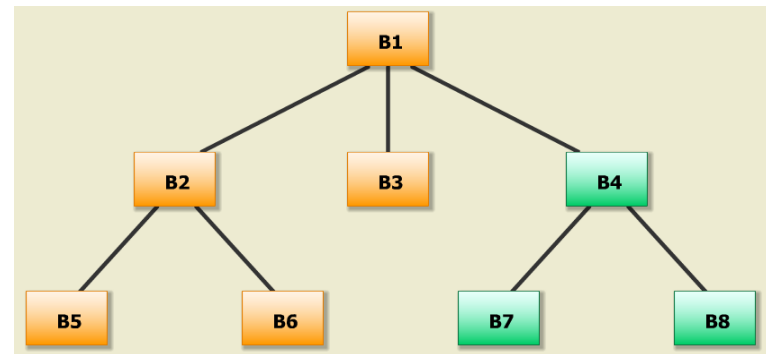


2. Adım: B1-B2 - B5 (KoçanB3- KoçanB4 - KoçanB6)

2. Yaklaşım: Derinlik Öncelikli Bütünleştirme

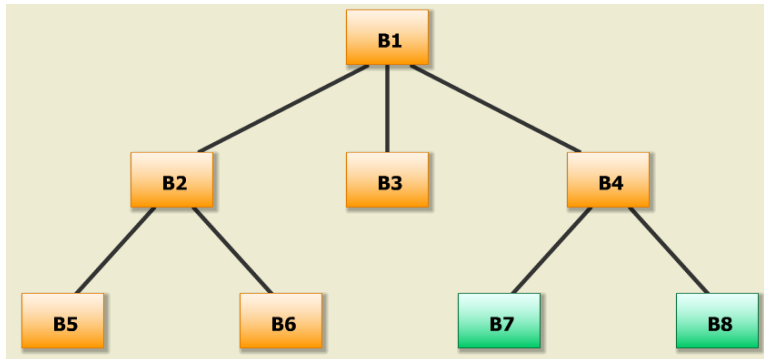


3. Adım: B1-B2-B5-B6 (KoçanB3 - KoçanB4)

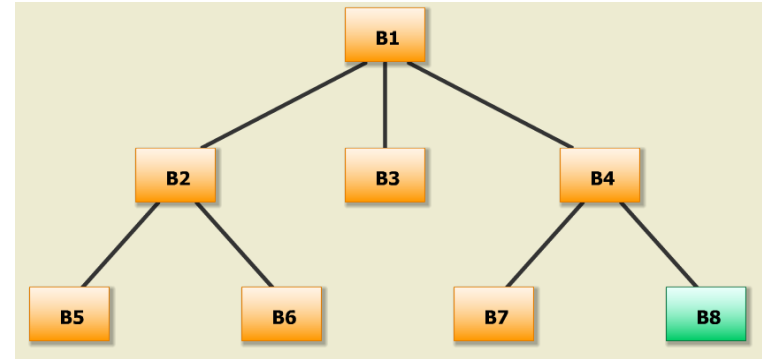


4. Adım: B1-B2-B5-B6-B3 (KoçanB4)

2. Yaklaşım: Derinlik Öncelikli Bütünleştirme

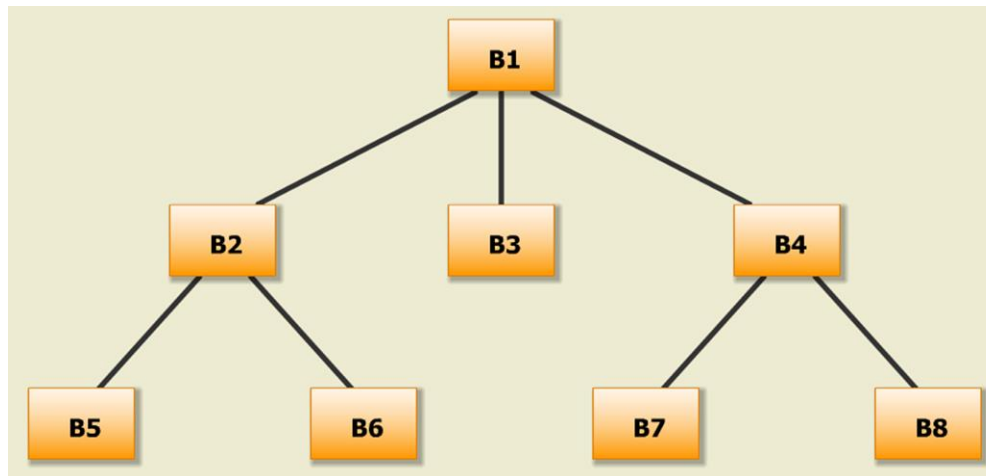


5. Adım: B1-B2-B5-B6-B3-B4 (KoçanB7 - KoçanB8)



6. Adım: B1-B2-B5-B6-B3-B4-B7 (KoçanB8)

2. Yaklaşım: Derinlik Öncelikli Bütünleştirme

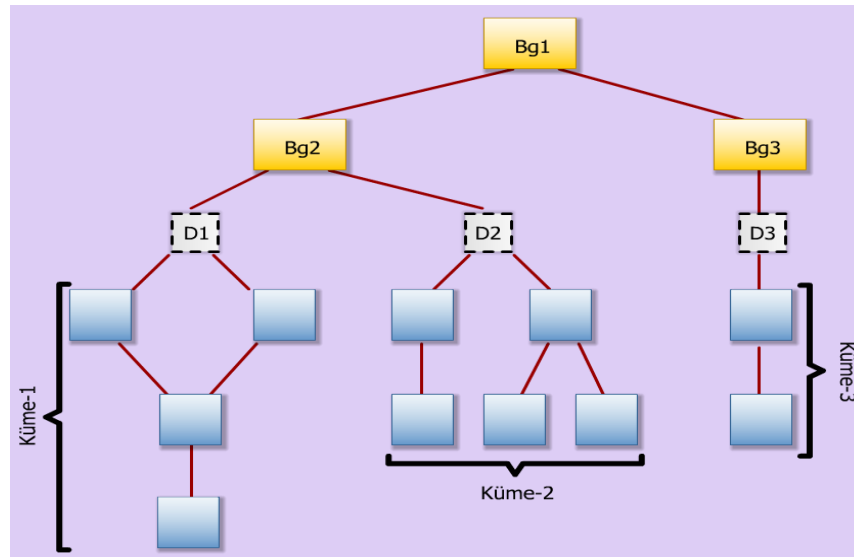


7. Adım: B1-B2-B5-B6-B3-B4-B7-B8

Aşağıdan Yukarıya Sınama ve Bütünleştirme

- Aşağıdan yukarı bütünleştirmede ise, önceki yöntemin tersine uygulama yapılır.
- Önce en alt düzeydeki işçi birimleri sınanır ve bir üstteki birimle sınama edilmesi gerektiğinde bu üst bileşen, bir 'sürücü' ile temsil edilir.
- Yine amaç, çalışmasa bile arayüz oluşturacak ve alt bileşenin sınanmasını sağlayacak bir birim edinmektir.
- Bu kez kodlama, bütünleştirme ve sınama aşağı düzeylerden yukarı düzeylere doğru gelişir ve yukarı düzeylerde önce sürücü olarak yazılan birimler sonra gerçekleriyle yer değiştirerek o düzeyin birimleri/alt sistemleri olurlar.
- Bütünleştirme yukarı doğru yapıldıkça daha az sürücü gereği duyulur.
- Uygulamada, hem aşağıdan yukarıya, hem de yukarıdan aşağıya sınama stratejilerinin iki stratejinin birleştirildiği de olur. 'Sandviç' adı verilen bu karma yaklaşımda hem üstten hem alttan sınama etkinliği sürer.

Aşağıdan Yukarıya Sınama ve Bütünleştirme



Şekilde;

1. Belirli bir yazılım alt işlevini gören alt düzey birimler **kümeler** biçiminde oluşturulurlar,
2. Denetim amaçlı bir **sürücü** programı sınama işlemi için girdi ve çıktı oluşturmak amacıyla yazılır,
3. Sürücüler aşağıdan yukarı kaldırılır ve gerçek birim ya da birim kümeleriyle değiştirilerek sınama işlemi sürdürülür.

Sinama Planlaması

- Sinama işlemi çok kapsamlıdır.
- Bir plan güdümünde gerçekleştirilmelidir.
- Böyle bir planın temel bileşenleri önceki sayfalarda belirtilmiştir.
- Yazılım yaşam döngüsünün süreçlerine koşut olarak, farklı ayrıntı düzeylerinde birden fazla sinama planı hazırlanır. Sinama planları;
 - Birim (Modül) Sinama Planı,
 - Alt Sistem Sinama Planları,
 - Bütünleştirme Sinama Planları,
 - Kabul Sinama Planları,
 - Sistem Sinama Planları biçimindedir.

Her sinama planı, sinama etkinliklerinin sınırlarını, yaklaşımını, kaynaklarını ve zamanlamasını tanımlar. Plan neyin sinanacağını, neyin sinanmayacağını, sorumlu kişileri ve riskleri göstermektedir. Sinama planları, sinama belirtilmelerini içerir.

Sinama Belirtilimleri

➤ Sinama belirtilimleri, bir sinama işleminin nasıl yapılacağına ilişkin ayrıntıları içerir. Bu ayrıntılar temel olarak:

- sinanan program modülü ya da modüllerinin adları,
- sinama türü, stratejisi (beyaz kutu, temel yollar vb.),
- sinama verileri,
- sinama senaryoları

türündeki bilgileri içerir.

➤ Sinama verilerinin elle hazırlanması çoğu zaman kolay olmayabilir ve zaman alıcı olabilir. Bu durumda, otomatik sinama verisi üreten programlardan yararlanılabilir.

➤ Sinama senaryoları, yeni sinama senaryosu üretebilmeye yardımcı olacak biçimde hazırlanmalıdır.

➤ Zira sinama belirtilimlerinin hazırlanmasındaki temel amaç, etkin sinama yapılması için bir rehber oluşturmaktır. Sinama işlemi sonrasında bu belirtilimlere,

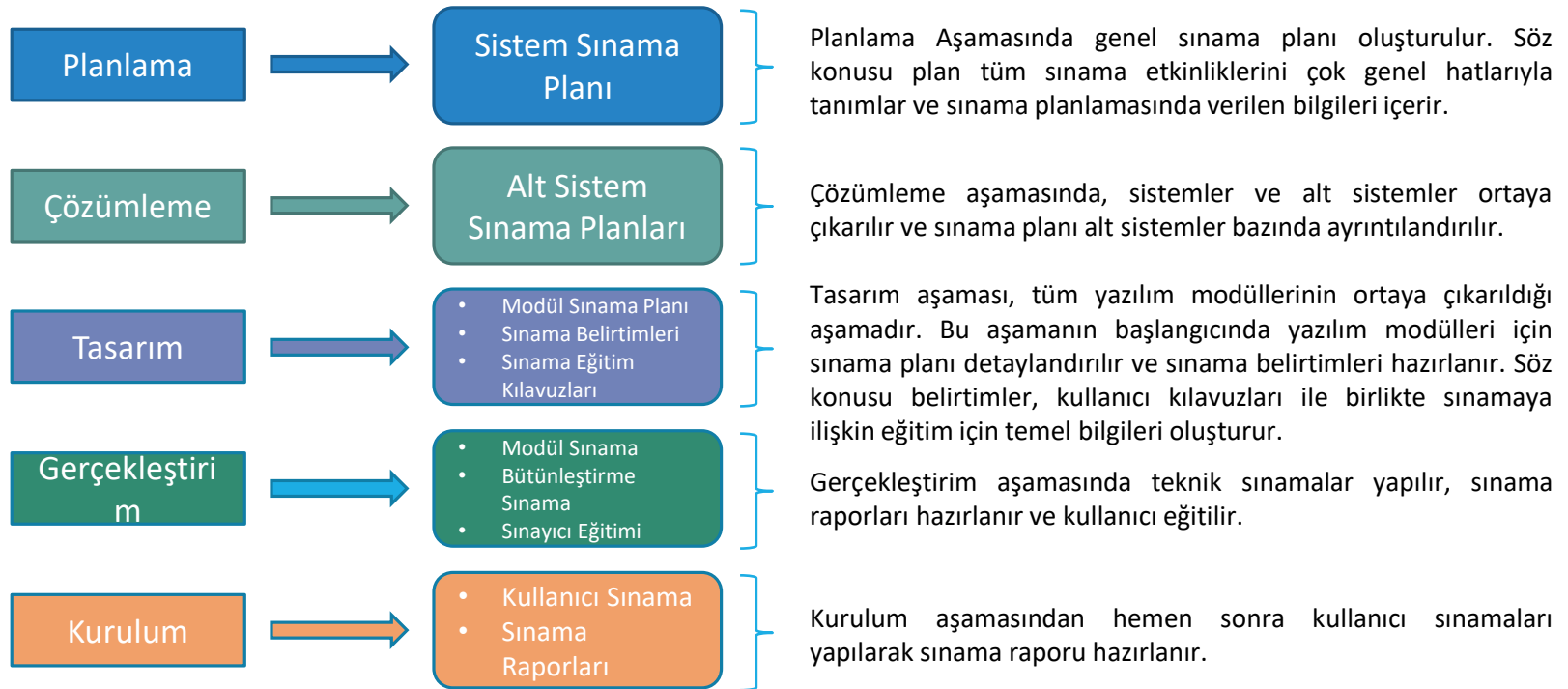
- sinamayı yapan,
- sinama tarihi,
- bulunan hatalar ve açıklamaları

türündeki bilgiler eklenerek sinama raporları oluşturulur.

Sinama Belirtilimleri

- Sinama raporları, sinama bitiminde imzalanır ve yüklenici ile iş sahibi arasında resmi belge niteliği oluşturur.
- Sinama planları; Birim (Modül) Sinama Planı, Alt Sistem Sinama Planları, Bütünleştirme Sinama Planları, Kabul Sinama Planları, Sistem Sinama Planları biçimindedir.
- Her sinama planı, sinama etkinliklerinin sınırlarını, yaklaşımını, kaynaklarını ve zamanlamasını tanımlar.
- Plan neyin sinanacağını, neyin sinanmayacağını, sorumlu kişileri ve riskleri göstermektedir. Sinama planları, sinama belirtilmelerini içerir.

Yazılım Yaşam Döngüsü Boyunca Sınama Etkinlikleri



Yazılım Yaşam Döngüsü Boyunca Sınama Etkinlikleri

- Hazırlanan sinama raporları, doğrulama ve geçerleme yaşam döngüsü işlemleri gereği "sorun yönetimi "ne iletilir. Bu bölümde hatalar kaydedilir ve bulunan hatalara karşı yapılacak işlemler planlanır.
- Sınamalar sırasında bulunan her bulgu ya da hata olarak belirtilen her durum gerçekte hata olmayabilir.
- Farklı sınavıcılarından biri, bir durumu hata olarak nitelerken diğeri aynı durumu doğru olarak değerlendirebilir.
- Bu nedenle sinama raporlarında hata olarak bildirilen her durum hemen düzeltilmek üzere ele alınmaz.
- Önce çözümlenir, kullanıcı çelişkileri giderilir ve gerçekten hata olduğuna karar verilirse düzeltilir. Söz konusu karar kullanıcı temsilcileri ile birlikte alınır.
- Sınama sırasında bulunan her hata için, değişiklik kontrol sistemine (DKS), "Yazılım Değişiklik İsteği" türünde bir kayıt girilir.

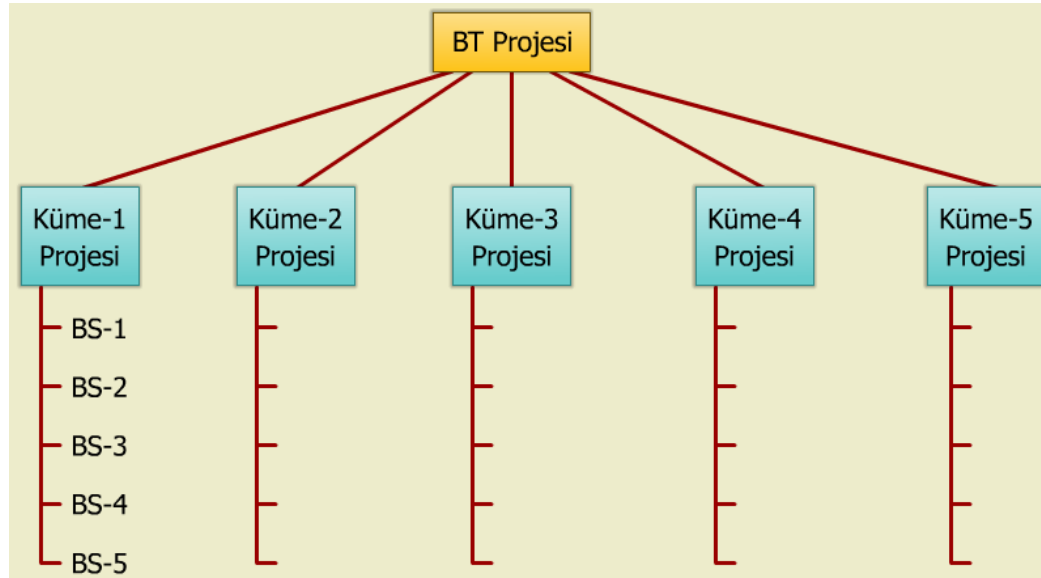
Yazılım Yaşam Döngüsü Boyunca Sınama Etkinlikleri

- Hatalar, DKS kayıtlarında aşağıdaki gibi gruplara ayrılabilir:
 - **Onulmaz Hatalar:** BT projesinin gidişini bir ya da birden fazla aşama gerileten ya da düzeltilmesi mümkün olmayan hatalardır.
 - **Büyük Hatalar:** Projenin kritik yolunu etkileyen ve önemli düzeltme gerektiren hatalardır.
 - **Küçük Hatalar:** Projeyi engellemeyen, ve giderilmesi az çaba gerektiren hatalardır.
 - **Şekilsel Hatalar:** Heceleme hatası gibi önemsiz hatalardır.

Uygulama : Görsel Yazılım Geliştirme Ortamında Sınama

- Bu kısımda, gerçek yaşam ortamında, **Oracle Designer CASE** aracı ve **Developer** görsel yazılım geliştirme platformu kullanılarak geliştirilen yazılım modüllerinin sınanması işleminin nasıl yapılacağı ve buraya kadar açıklanan sinama yöntemlerinin nasıl uygulandıkları bir örnek üzerinde anlatılmaktadır.
- Oracle Developer kullanılarak geliştirilen her yazılım formlardan oluşur. Bir form, bir ekran ve bu ekranda yapılan işlemlere karşılık gelen PL/SQL kodları biçiminde tanımlanır.
- Bu örnekte elimizde, sinama işlemine koşulacak ve uygulamanın çeşitli işlevlerine ilişkin bir dizi form olduğunu düşünebiliriz.
- Bu örnekte söz edilen uygulama, 2000'den fazla kullanıcısı olan, ülkenin çeşitli yörelerine dağılmış birimlerde çalışacak biçimde tasarlanmış ve 1000'den fazla Developer formundan oluşmaktadır.
- Uygulamanın sinama aşamasına gelmesi, 2 yıllık bir süre ve yaklaşık 100 kişi-yıl'lık bir iş gücü gerektirmiştir. Uygulama beş ana kümeye bölünmüş ve her küme belirli sayıda bilgi sistemini içermektedir.
- Toplam olarak 30 bilgi sistemi bulunmaktadır.
- Uygulama sıra düzeni bir sonraki slaytta gösterilmektedir.

Uygulama : Görsel Yazılım Geliştirme Ortamında Sınama



- Şimdi böyle kapsamlı bir uygulamanın sınama aşamasında kullanılan yöntemleri ve metodolojiyi bir sonraki slaytta inceleyelim.

Sinama Ortamı Oluşturulması

- Üretimin etkilenmemesi amacıyla, yalnızca sanayicilerin kullanacakları ve ayrı bilgisayarlardan oluşan bir sinama ortamı oluşturuldu.
- Oluşturulan sinama ortamı ile üretim ortamının bir bir aynı olması sağlandı.
- Üretimi biten yazılım parçaları, bir kayıt düzeni içerisinde sinama ortamına alındı.

Sinama Yöntemlerine Karar Verilmesi

- Uygulamada kullanılacak sinama yöntemleri aşağıdaki gibidir:
 - Teknik Sinama
 - Biçimsel Sinama
 - İşletimsel Sinama
 - Senaryo Sinama
 - Kullanıcı Sinama

Teknik Sınama

- Üretim ortamında yapılacak sınama olarak karar verildi. Bu sınama, modül sınaması ve bütünleştirme sınaması olarak üretim ekipleri tarafından gerçekleştirildi. Modül sınama yöntemi olarak 'Beyaz kutu' sınama yöntemi uygulandı. Tüm program deyimleri en az bir kez, tüm döngüler en az 10 kez yinelenecek biçimde sınama yapıldı.
- Bütünleştirme sınama yöntemi olarak 'yukarıdan aşağıya sınama yöntemi' ve 'derinlik öncelikli bütünleştirme' stratejisi uygulandı.

Biçimsel Sınama

- Üretim ekiplerinden bağımsız olarak, sınama ekipleri tarafından yapılan sınamadır. Bu sınama, Developer formları üzerinde görsel olarak yapıldı. Amaç, formların, önceden kararlaştırılan standartlara uygunluğunun saptanmasıydı.
- Örneğin,
 - form alanları, kararlaştırılan uzunlukta mı?
 - Başlıklar istenilen gibi koyu mu?
 - Yardım düğmesi hep aynı yerde mi vb.
- Sınama, formlar işletilmeden yapılır. Tüm formlar tek tek incelenir ve standartlara uygun olmayanlar belirlenip, düzeltilmek üzere üretim ekibine geri iletilir.
- Biçimsel sınamaların yapılması amacıyla denetim listeleri hazırlanır ve sınama sırasında bu listeler kullanılır. Listelere kaydedilen her sonuç DKS'ye aktarılır.
- Bu yolla üretim ekiplerinin performansı izlenebilir.

İşletimsel Sınama

- Üretim ekiplerinden bağımsız olarak, sınama ekipleri tarafından yapılan sınamadır.
- Biçimsel sınama işlemi bittikten sonra yapılır.
- Bu sınamada her form ayrı ayrı çalıştırılarak işlem yapılır.
- Amaç, formun çalışıp çalışmadığının belirlenmesidir.
- Form alanlarının sınır değerlerle çalışıp çalışmadığı, aykırı değer verildiğinde uygun hata iletisi alınıp alınmadığı vb. belirlenmeye çalışılır.

Senaryo Sınama

- Sınama ekipleri tarafından yapılan sınamadır.
- Ancak, senaryoların hazırlanması sırasında üretim ekipleri ile birlikte çalışılır.
- Amaç, birden fazla formun bir arada sınanmasıdır. Bu amaçla, 'senaryo'lar hazırlanır.
- Her senaryo, çözümleme aşamasında belirlenen bir iş fonksiyonuna karşılık gelecek biçimde hazırlanır.

Kullanıcı Sinaması

- Kullanıcılar tarafından yapılması öngörülen sinamadır.
- Senaryo sinamasının kullanıcı tarafından yapılan biçimi olarak düşünülebilir.

Kullanıcı Sınama Eğitimi

- Sınama yapılacak kullanıcı sinayıcılarına, sinamaların nasıl yapılacağına ilişkin eğitim verilmesi gerekmektedir.
- Eğitim kitapçıklarının hazırlanması amacıyla, senaryo sinamalarında kullanılan "senaryo"lar ve kullanıcı kitapçıkları kullanılır.

Sınamaların Yapılması

- Sınamalar sırasında yapılan her işlem, DKS'de izlenir. Özellikle kullanıcı sınamalarının izlenmesi ve ortaya çıkabilecek tartışmaların önlenmesi bu yolla sağlandı.
- Yaklaşık 100 kullanıcı sınavıcısının, birbirinden farklı yerlerde yaptıkları sınamalar için "haftalık sinama sonuçları" formları toplandı.
- Yerinde destek elemanları, sürekli olarak kullanıcı sınavıcılarını ziyaret ederek sinama sonuçlarının düzenli olarak toplanmasını sağladı.
- Bu formlar DKS'ye aktarılarak, daha sonra Yazılım Doğrulama ve Geçerleme Raporları için önemli girdiler oluşturdu.

Özet

Doğrulama: Doğru ürünü mü üretiyoruz ?

- Ürünü kullanacak kişilerin isteklerinin karşılanıp karşılanmadığına dair etkinliklerden oluşur.

Geçerleme: Ürünü doğru olarak mı üretiyoruz?

- Ürünün içsel niteliğine ilişkin izleme ve denetim etkinliklerinden oluşur.

Doğrulama ve geçerleme işlemleri temel olarak çeşitli düzeylerde sınaama, gözden geçirme, denetim ve hata giderme süreçlerinden oluşur.

Doğrulama Teknikleri: Gözden Geçirme, Üstünden Geçme, Denetleme ve İncelemedir.

Sınaama planları;

- Birim (Modül) Sınaama Planı,
- Alt Sistem Sınaama Planları,
- Bütünleştirme Sınaama Planları,
- Kabul Sınaama Planları,
- Sistem Sınaama Planları biçimindedir.

Özet

Hatalar, DKS kayıtlarında aşağıdaki gibi gruplara ayrılabilir:

- **Onulmaz Hatalar:** BT projesinin gidişini bir ya da birden fazla aşama gerileten ya da düzeltilmesi mümkün olmayan hatalardır.
- **Büyük Hatalar:** Projenin kritik yolunu etkileyen ve önemli düzeltme gerektiren hatalardır.
- **Küçük Hatalar:** Projeyi engellemeyen, ve giderilmesi az çaba gerektiren hatalardır.
- **Şekilsel Hatalar:** Heceleme hatası gibi önemsiz hatalardır.

Uygulamada kullanılacak sınaama yöntemleri aşağıdaki gibidir:

- Teknik Sınama
- Biçimsel Sınama
- İşletimsel Sınama
- Senaryo Sınama
- Kullanıcı Sınama

Sorular

1. Doğrulama ile Geçerleme arasındaki farklılıkları belirtiniz. Birer örnekle açıklayınız.
2. Sınama Yöntemlerini açıklayınız.
3. "Beyaz Kutu" sınama ile "Temel Yollar Sınama" yöntemleri arasındaki farklılıkları belirtiniz.
4. Sınama Yöntemleri ile sınama belirtilimleri arasındaki farkı belirtiniz.
5. Yukarıdan aşağıya doğru bütünleştirme ve aşağıdan yukarıya bütünleştirme yöntemlerinin zorluklarını ve kolaylıklarını belirtiniz.
6. Sınama belirtilimlerinin önemi nedir.
7. Kullanıcı sinaması sırasında yaşanabilecek sorunları belirtiniz.

Kaynaklar

- “Software Engineering A Practitioner’s Approach” (7th. Ed.), Roger S. Pressman, 2013.
- “Software Engineering” (8th. Ed.), Ian Sommerville, 2007.
- “Guide to the Software Engineering Body of Knowledge”, 2004.
- ” Yazılım Mühendisliğine Giriş”, TBİL-211, Dr. Ali Arifoğlu.
- ”Yazılım Mühendisliği” (2. Basım), Dr. M. Erhan Sarıdoğan, 2008, İstanbul: Papatya Yayıncılık.
- Kalıpsız, O., Buharalı, A., Biricik, G. (2005). Bilgisayar Bilimlerinde Sistem Analizi ve Tasarımı Nesneye Yönelik Modelleme. İstanbul: Papatya Yayıncılık.
- Buzluca, F. (2010) Yazılım Modelleme ve Tasarımı ders notları (<http://www.buzluca.info/dersler.html>)
- Hacettepe Üniversitesi BBS-651, A. Tarhan, 2010.
- Yazılım Proje Yönetimi, Yrd. Doç. Dr. Hacer KARACAN

Ödev

- Yazılım Mimarileri Hakkında Araştırma Yapınız.
- İki Katmanlı ve Üç Katmanlı Mimarilerle yapılmış örnek sistemleri araştırınız.

