

YAZILIM MÜHENDİSLİĞİ – Şubat 2012
Yrd.Doç.Dr. Yunus Emre SELÇUK

YAZILIM SINAMA TEKNİKLERİ

1

YAZILIM SINAMA TEKNİKLERİ

GENEL BİLGİLER

- Yazılım geliştirme karmaşık bir süreç olduğundan, hataların ortaya çıkması kaçınılmazdır.
- Yazılım, yaşam döngüsünün her aşamasında, hatalara karşı sınılanır.
 - Gereksinimler arasındaki tutarsızlıklar,
 - Çözümleme şeması ile uygulama alanı arasındaki uyumsuzluklar,
 - Tasarım hataları,
 - Çalışma anı hataları,
 - vb.
- Yaşam döngüsünde ilerledikçe, hataların düzeltilmesi zorlaşacaktır.
- İyi bir sinama yaklaşımı, hataların erken belirlenmesine katkıda bulunacak ve yazılımın kalitesini arttıracaktır.
- Bir sinama yaklaşımı şu bileşenleri içermelidir:
 - Planlama
 - Tasarım
 - Çalıştırma
 - Bilgi toplama ve değerlendirme

2

YAZILIM SINAMA TEKNİKLERİ

GENEL BİLGİLER

- Sınama yaklaşımlarının genel özellikleri:
 - Sınamalar bileşen düzeyinde başlar ve sistem düzeyinde sonlanır.
 - Ürün tek başına bilgisayar yazılımından ibaret olmayabilir.
 - Sistemin bütününde çeşitli algılayıcılar, harici sistemler, vb. yer alabilir.
 - NYP'de sınıfların bireysel sorumluluklarından başlayıp, yazılımın üst düzey sorumluluklarına kadar tüm işlevler sınanır.
 - Yazılım yaşam döngüsünün değişik aşamalarında değişik sınama teknikleri uygun olacaktır.
 - Projenin değişik aşamalarında değişik sınama teknikleri uygun olacaktır.
 - Sınamalar hem ilgili bileşeni oluşturan kişi tarafından, hem de bağımsız kişiler tarafından yapılır.
 - Büyük ölçekli kurumlarda ayrı bir sınama ekibi bulunabilir.
 - Sınama ve hata ayıklama ayrı işlerdir, ancak hata ayıklama her sınama yaklaşımının önemli bir parçasıdır.
 - Resmi teknik değerlendirmeler sayesinde sınama öncesinde de hatalar belirlenebilir.
 - Formal technical reviews, ileride değinilecek.

3

YAZILIM SINAMA TEKNİKLERİ

SINAMA AMAÇLI YAPILANMA

- Bir bileşen ilk olarak onu hazırlayan programcı tarafından sınanır.
 - Bir programı en iyi olarak bilen, onu yazandır.
- Her bileşen aynı zamanda yazarı dışındaki kişilerce sınanır.
 - Yazılım evinin bağımsız bir sınama ekibi bulunabilir (ITG: Independent Test Group).
 - Aksi halde farklı projelerde çalışan ekipler, veya aynı projenin farklı alt grupları, birbirlerinin çalışmalarını sınar.
 - Programcıların çıkarları, kendi yazdıkları programın hatasız, zamanında ve bütçe dahilinde tamamlanmasını gerektirir.
 - Psikolojik açıdan bakıldığında sınama 'yıkıcı' bir eylemdir.
 - Bu nedenlerle geliştiriciler kendi programlarının hatalarını bulmaya yönelik değil, doğru çalıştığını ispatlamaya yönelik sınamalara eğilim gösterebilir.
- Yazılım teknik olmayan kişiler tarafından da sınanır (ileride değinilecek).
- Sonuç:
 - Müşteriler hataları önünde sonunda bulacaklardır. İyisi mi siz onlardan önce davranın! Aksi halde prestijiniz sarsılacaktır.

4

YAZILIM SINAMA TEKNİKLERİ

SINAMA YAKLAŞIMININ BELİRLENMESİ

- Göz önünde bulundurulması gereken konular:
 - Sınama çalışmaları, en çok çaba gerektiren yazılım mühendisliği etkinlikleri arasında yer almaktadır.
 - Sınama çalışmaları, olası tüm hataların yakalanacağı bir 'güvenlik ağı' olarak düşünülmemelidir.
 - Sınama için ne kadar çaba gösterilirse gösterilsin, tespit edilemeyen hatalar olacaktır.
 - Hedefler açıkça belirlenmelidir:
 - Testlerin kapsama alanı,
 - Sınama çalışmalarına ne kadar kaynak ayrılacağı,
 - zaman, kişi, bütçe
 - Yazılımın türüne göre belirlenecek diğer çeşitli ölçütler
 - İki hata arası ortalama süre (MTBF), hassasiyet, vb.
 - Sınamalar otomatikleştirilmelidir.
 - Sınama süreci ölçülmeli ve iyileştirilmelidir.

5

YAZILIM SINAMA TEKNİKLERİ

SINAMA TÜRLERİ

- Yaklaşım tarzlarına göre sınama türleri:
 - Kara kutu sınaması (Black-box testing): Sınanacak birimin iç işleyişi bilinmez, sadece birimin beklenen girdilere karşı beklenen çıktıları üretip üretilmediğine bakılır.
 - Beyaz kutu sınaması (White-box testing): Sınanacak birimin iç işleyişi bilinir ve yapılacak sınamalar buna göre belirlenir.
- Yürütülme sıralarına göre sınama türleri:
 - Doğrulama Sınamaları (verification tests): Yazılım ekibi tarafından yapılır.
 - Birim sınamaları
 - Tümleştirme sınamaları
 - Geçerleme Sınamaları (validation tests): Son kullanıcılar tarafından yapılır.
 - Alfa sınaması
 - Beta sınaması

6

YAZILIM SINAMA TEKNİKLERİ

DOĞRULAMA SINAMALARI

- Birim sınamaları (Unit testing)
 - En küçük yazılım bileşeninin sınanmasıdır.
 - NYP'de bireysel sınıfların sınanmasıdır.
 - Ne zaman tasarlanır?
 - Kodlamadan önce (çevik yaklaşım),
 - kodlama sırasında,
 - veya kodlamanın ardından.
 - Kodlama sırasında veya kodlamanın ardından yürütülebilir.
 - Bir sınıfın tek başına yürütemediği sorumlulukların sınanması için, bu sınıfın ihtiyaç duyduğu diğer sınıfların yerine geçecek kod gerekebilir.
 - Vekil, sahte, yalancı kod/sınıf, vb.
 - Stub, dummy, surrogate, proxy, vb.
 - Vekil sınıflar, sadece ihtiyaç duyulan sınıflar gerçekleştirilene dek kullanılır.
 - Vekil sınıfların basit tutulması, ek kodlama yükünü azaltır.
 - Bu mümkün değilse, ortaklaşa yürütülen sorumlulukların sınanması tümleştirme sınamalarına bırakılır.

7

YAZILIM SINAMA TEKNİKLERİ

DOĞRULAMA SINAMALARI

- Birim sınamalarında aranabilecek hata türleri:
 - Farklı veri tiplerinin karşılaştırılması veya birbirinin yerine kullanımı.
 - NYP'de: Çokbiçimliliğin yan etkileri
 - Mantıksal işleçlerin yanlış kullanımı
 - İşleçlerin önceliklerinin gözden kaçırılması
 - Değişkenlerin karşılaştırılmasındaki hatalar
 - Döngülerin hatalı sonlanması veya sonsuz döngüler
 - Değişkenlere hatalı değerler atanması
 - vb.
- Yaptığınız hatalardan ders çıkarın:
 - Yapılabilecek tüm kodlama hataları öngörülemez, ancak kariyeriniz boyunca her hata yaptığınızda bu hatanızı 'aranabilecek hata türleri' listenize ekleyin.

8

YAZILIM SINAMA TEKNİKLERİ

DOĞRULAMA SINAMALARI

- Tümlleştirme sinamaları (Integration testing)
 - Sınıflar birim sinamalarını geçmişlerse, bir araya getirildiklerinde de doğru çalışmazlar mı?
 - Yazılım geliştirme sürecinin her aşamasında her ayrıntının açıkça belirlenmesi beklenemez.
 - Aynı ayrı programcılar, belirlenmemiş ayrıntılar üzerinde kendi karar verme yetkilerini (initiative) kullanabilir.
 - Aynı kişinin farklı ayrıntılar hakkında verdiği kararlar bile birbiri ile uyumlu olmayabilir.
 - NYP'de birim ve tümlleştirme sinamalarını birbirinden ayıran kesin çizgiler yoktur.

9

YAZILIM SINAMA TEKNİKLERİ

DOĞRULAMA SINAMALARI

- Tümlleştirme sinamaları türleri:
 - Tahribat sinaması (smoke testing)
 - Yüzeysel ancak başarısız olma durumunda tüm sistemin çalışmasının olanaksız olacağı sinamalardır (show-stopper errors).
 - Diğer tümlleştirme sinamalarından önce yapılır.
 - Çeşitli bileşenler tüm gerekli yazılım elemanlarını içeren (kod, yapılandırma dosyaları, dış kütüphaneler, vb.) parçalar bir araya getirilir (build) ve günlük olarak sinanırlar.
 - Geriye dönük sinama (regression testing)
 - Yazılıma yeni bir işlev veya bileşen eklendiğinde, tüm sinamaların yenilenmesidir.
 - Hangi ölçekte bir eklentinin geriye dönük sinamayı başlatacağının kararını vermek gerekir.
 - Ölçek düştükçe sinama sıklığı ve masraf artar.
 - Otomatik sinama gereçleri kullanılarak masraflar azaltılabilir.

10

YAZILIM SINAMA TEKNİKLERİ

GEÇERLEME SINAMALARI

- Geçerleme sınamaları (validation testing):
 - Gereksinimler belgesinde yazılmış olan işlevsellikten yola çıkılarak, kullanıcı tarafından yapılır.
 - Son kullanıcıların yapabileceği beklenmedik davranışların tümünü, teknik ekip önceden bilemez.
 - 'Fincan tutacağıının çalışmaması', 'Pencereyi açmak', ...
 - Bir şeyi ne kadar çok kişi incelerse, ondaki kusurlar o kadar çabuk bulunur ve düzeltilir.
 - Eric Raymond: "given enough eyeballs, all bugs are shallow"
 - Alfa ve Beta sınaması olmak üzere iki türü vardır.

11

YAZILIM SINAMA TEKNİKLERİ

GEÇERLEME SINAMALARI

- Alfa sınaması:
 - Yazılım firması içerisinde, kullanıcı tarafından yapılır.
 - Yazılım geliştirme ekibinin denetiminde ve izlemesi ile yapılır.
 - Yazılımın doğal kullanım ortamına en yakın koşullarda yürütülür.
- Beta sınaması:
 - Müşterinin kendi yeri içerisinde, gerçek kullanım ortamında yapılır.
 - Yazılım geliştirme ekibi müdahil olmaz.
 - Bulunan hatalar yazılım geliştirme ekibine düzenli aralıklarla ve resmi bir biçimde bildirilir.

12

YAZILIM SINAMA TEKNİKLERİ

SİSTEM SINAMALARI

- Yazılım tek başına sistemin bütünü oluşturmayabilir.
 - Gömülü uygulamalar, ara katman yazılımları, vb.
- Sistemin tümünü her yönüyle incelemeye yönelik sınamalardır:
 - Kurtarma (recovery) sınaması:
 - Hatalara dayanıklı (fault tolerant) sistemler için geçerlidir.
 - Bir hata ortaya çıktığında sistemin kendini toparlayarak doğru çalışmaya devam edip edemediği sınanır.
 - Kurtarma işlemi belli bir süre içerisinde tamamlanmalıdır (MTTR: Mean Time To Repair).
 - Güvenlik (security) sınaması:
 - Tek kural: Kural yok!
 - Her güvenlik önünde sonunda aşıılır!
 - Zorlama (stress) sınaması:
 - Normalin dışında yüklenme durumunda, sistemin nereye kadar dayanabileceğinin sınanması
 - Başarım (performance) sınaması:
 - Gerçek zamanlı uygulamalarda özel öneme sahiptir.
 - Program kendisinden bekleneni doğru yapabilir ama zamanında yapamayabilir.

13

YAZILIM SINAMA TEKNİKLERİ

HATA AYIKLAMA (DEBUGGING)

- Yapılan türlü sınamaların sonucunda bulunan hatalar düzeltilmelidir.
- Sınama ve hata ayıklama çalışmaları kimileri tarafından 'angarya', 'ayak işi', 'sıkıcı', 'ikinci sınıf' olarak nitelendirilebilir.
 - Sınamanın önemini gördük
 - Yazılımınızdaki hatalar size ve kurumunuza prestij kaybettirir.
 - Düzeltilmeyen veya düzeltilmesi uzun süren hatalar ise daha çok prestij kaybettirir.
 - Sınama ve güvenlik açıklarının belirlenmesi için alışlagelmiş düşünce biçiminin dışına çıkabilmek gerekir ki bu da özel bir yetenektir.
 - Saygı duyulan ve ünlü programcılar, hata ayıklamanın kodlama yapmaktan daha zor ve daha çok yetenek isteyen bir iş olduğu görüşünde birleşmektedirler.
- Bir noktada takıldığınızda, biraz ara verip sorunu yeniden incelemeniz, başarı şansınızı arttıracaktır.
- Geliştirme ortamının hata ayıklama yeteneklerinden sonuna kadar yararlanın.

14

YAZILIM SINAMA TEKNİKLERİ

YAZILIM GÖZDEN GEÇİRME EYLEMİ

- Yazılım gözden geçirme: Software review
 - Yazılım sınavından önce, sınama eylemlerinden daha az masrafla, yazılım hatalarının bulunmasını amaçlar.
 - Sınama çalışmaları ile bulunabilecek tüm hatalar gözden geçirme ile bulunamaz, ancak gözden geçirme daha verimlidir.
 - Gözden geçirme resmi veya gayri resmi olabilir.
 - Çalışmalar resmi gözden geçirmelerin daha etkili olduğunu göstermiştir.
 - Gözden geçirme toplantılarının yapılması durumunda.
 - Çevik süreçlerden XP'deki eşli programlama da bir tür gözden geçirmedir.
 - Resmi yazılım gözden geçirme: Formal software review
 - Toplantı şeklinde yapılır.
 - Toplantının bir yöneticisi (review leader) bulunur.
 - Bu konuda çeşitli standartlar önerilmiştir.
 - Ör: IEEE 1028 standardı

15

YAZILIM SINAMA TEKNİKLERİ

RESMİ YAZILIM GÖZDEN GEÇİRME ÇALIŞMALARI

- IEEE 1028 std. göre bir resmi yazılım gözden geçirme adımları:
 - Değerlendirme başlangıcı: Değerlendirme yöneticisi standart bir checklist kullanarak, verimli bir toplantı için gerekli koşulları sağlar.
 - Yönetimin hazırlanması: Sorumlu yönetim gözden geçirme için gerekli kaynakları hazırlar ve toplantının standartlara uygun yürütülmesini sağlar.
 - Gözden geçirme prosedürlerine genel bakış: Değerlendirme yöneticisi tüm değerlendiricilerin gözden geçirmenin amaçlarını ve prosedürlerini anladığından emin olur.
 - Bireysel hazırlık: Değerlendiriciler bireysel olarak inceleme toplantısına hazırlanır. Bu amaçla gözden geçirilecek malzemede hataya yol açabilecek olası bozukluklar (anomaly) aranır.
 - Grup incelemesi: Bireysel hazırlıkların sonuçları önceden belirlenen yer ve zamandaki toplantıda biraraya getirilir ve sonuç raporu üzerinde uzlaşıya varılır.
 - Düzenleme: İncelenen çalışmanın yazar(lar)ı veya atanacak bir başka kişi/ekip, önceki adımda belirlenen noktaları düzeltir.
 - Sonlandırma: Değerlendirme yöneticisi düzeltmelerin yeterliliğini inceler.

16