

BİM539, Ders 7: Manuel vs Otomatik Test

Manuel Test

- Şu ana kadar yaptıklarımız
- Yazılımda test planlarını bir insan uygular

Otomatik Test

- Çoğunlukla bundan sonra yapacağımız şey
- Bilgisayarlar test planlarını bizim için oluşturur
- Aralarında oldukça büyük fark bulunmaktadır
 - Tek bir test planı binlerce kez çalıştırılabilir

Manuel / Otomatik Test Karşılaştırma

Tip	Artılar	Eksiler
Manuel Test		
Otomatik Test		

Manuel / Otomatik Test Karşılaştırma

Tip	Artılar	Eksiler
Manuel Test	<ul style="list-style-type: none">+ Vasıfsız personel tarafından yapılabilir+ Test otomasyon araçlarına ihtiyaç yoktur+ Esneklik: Kişiler yazılımdaki küçük değişiklikleri test senaryosuna uyarlayabilir+ İnsanlar test senaryosunda açıkça olmasa bile yazılımdaki sorunları fark edebilir	
Otomatik Test		

Manuel / Otomatik Test Karşılaştırma

Tip	Artılar	Eksiler
Manuel Test	<ul style="list-style-type: none">+ Vasıfsız personel tarafından yapılabilir+ Test otomasyon araçlarına ihtiyaç yoktur+ Esneklik: Kişiler yazılımdaki küçük değişiklikleri test senaryosuna uyarlayabilir+ İnsanlar test senaryosunda açıkça olmasa bile yazılımdaki sorunları fark edebilir	<ul style="list-style-type: none">- Yoğun zaman ve kaynak ihtiyacı- Tekrarlanabilirlik problemleri (ör. Test zamanlama konusunda hassas ise insanlar bu konuda çok da iyi değildir)- İnsan hatalarına açık- Testler kullanıcı arayüzü ile sınırlandırılır (Ör: birim test yapamaz)
Otomatik Test		

Manuel / Otomatik Test Karşılaştırma

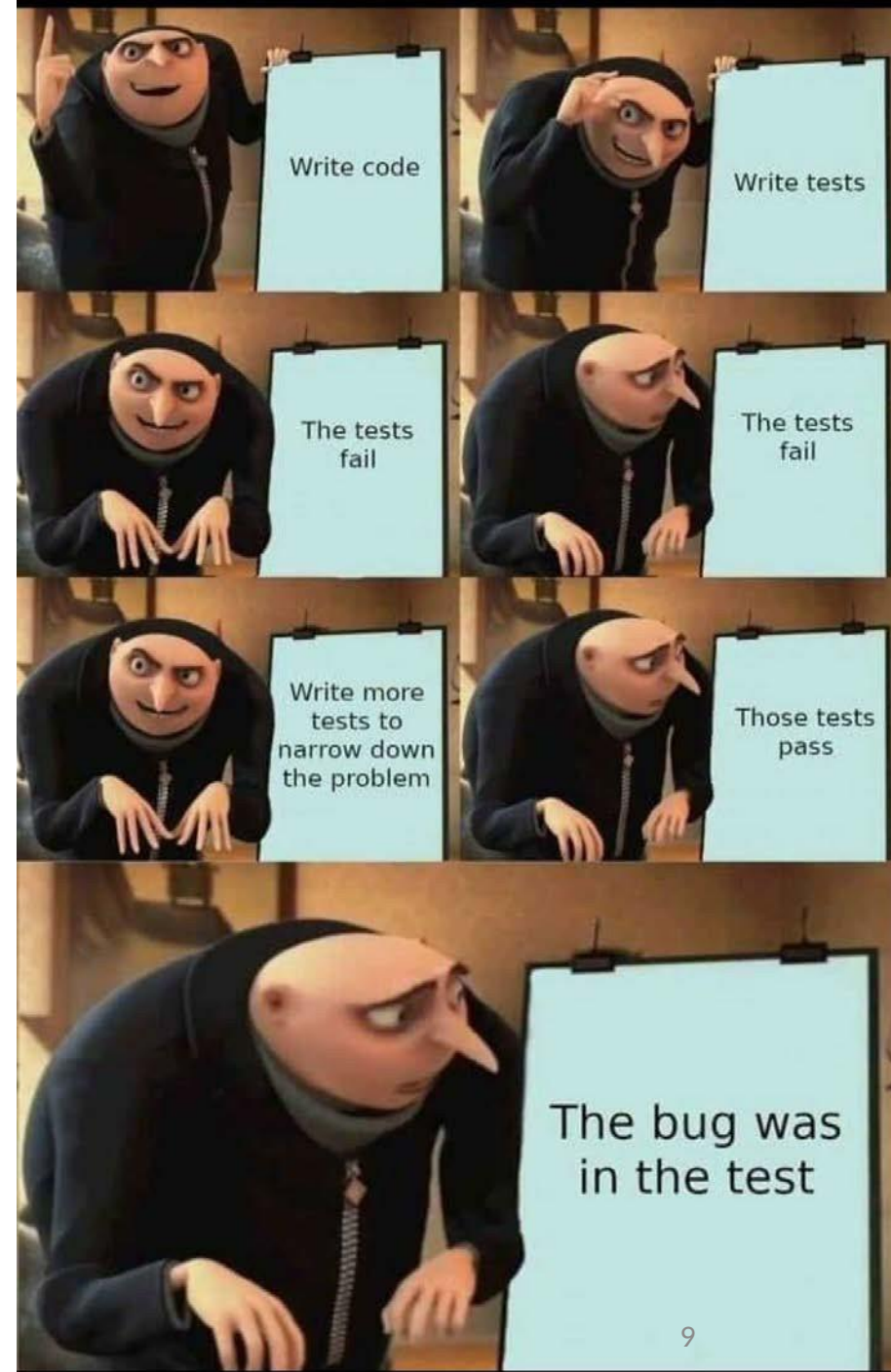
Tip	Artılar	Eksiler
Manuel Test	<ul style="list-style-type: none">+ Vasıfsız personel tarafından yapılabilir+ Test otomasyon araçlarına ihtiyaç yoktur+ Esneklik: Kişiler yazılımdaki küçük değişiklikleri test senaryosuna uyarlayabilir+ İnsanlar test senaryosunda açıkça olmasa bile yazılımdaki sorunları fark edebilir	<ul style="list-style-type: none">- Yoğun zaman ve kaynak ihtiyacı- Tekrarlanabilirlik problemleri (ör. Test zamanlama konusunda hassas ise insanlar bu konuda çok da iyi değildir)- İnsan hatalarına açık- Testler kullanıcı arayüzü ile sınırlandırılır (Ör: birim test yapamaz)
Otomatik Test	<ul style="list-style-type: none">+ Ayarlamalar yapılıncaya çalıştırması hızlı ve daha az maliyetlidir+ Tekrarlanabilirlik açısından daha iyidir+ İnsan hatalarına yer yoktur+ Junit gibi araçları kullanarak birim test gerçekleştirebilir	

Manuel / Otomatik Test Karşılaştırma

Tip	Artılar	Eksiler
Manuel Test	<ul style="list-style-type: none">+ Vasıfsız personel tarafından yapılabilir+ Test otomasyon araçlarına ihtiyaç yoktur+ Esneklik: Kişiler yazılımdaki küçük değişiklikleri test senaryosuna uyarlayabilir+ İnsanlar test senaryosunda açıkça olmasa bile yazılımdaki sorunları fark edebilir	<ul style="list-style-type: none">- Yoğun zaman ve kaynak ihtiyacı- Tekrarlanabilirlik problemleri (ör. Test zamanlama konusunda hassas ise insanlar bu konuda çok da iyi değildir)- İnsan hatalarına açık- Testler kullanıcı arayüzü ile sınırlandırılır (Ör: birim test yapamaz)
Otomatik Test	<ul style="list-style-type: none">+ Ayarlamalar yapılıncaya çalıştırması hızlı ve daha az maliyetlidir+ Tekrarlanabilirlik açısından daha iyidir+ İnsan hatalarına yer yoktur+ Junit gibi araçları kullanarak birim test gerçekleştirebilir	<ul style="list-style-type: none">- Test altyapısını oluşturmak için zamana ihtiyaç vardır- Test altyapısını oluşturmak için vasıflı çalışanlara ihtiyaç duyulur- Test araçlarına ihtiyaç bulunur (ve bunları öğrenmeye)- Hassas: Yazılımdaki ufak değişiklikler testin geçersiz olmasına sebep olabilir bu yüzden bakım maliyetlidir.- Kısıtlı: Testler yalnızca açıkça yazılan şeyleri test eder.

Otomatik Test HASSASTIR

- Makinalar insanlar gibi esnek değildir
- Uygulamadaki küçük değişiklikler bile testte hataya sebebiyet verebilir
 - Ör. Uygulamanın GUI'sinde ufak bir değişiklik
 - Testler de uygulama ile birlikte güncellenmelidir
 - Yani, Test kodları kodların bir parçası haline gelir (Ve onlar da bakıma ihtiyaç duyar)
 - Sonuç olarak, uygulamayı debug ettiğiniz kadar testleri de debug edersiniz!
- “Self-healing” testleri daha esnek hale getirebilir
 - Ancak bu henüz yeteri kadar gelişmiş değildir



Otomatik test KISITLIDIR

- “Bilinen bilinenler var. Bunlar bildiğimizi bildiğimiz şeyler. Bilinen bilinmeyenler var. Yani bilmediğimizi bildiğimiz şeyle. Ama bir de bilinmeyen bilinmeyenler var. Bilmediğimizi bilmediğimiz şeyler.

-Donald Rumsfeld

- Otomatik test yalnızca “bilinen bilinmeyenleri” kontrol eder
 - Test senaryosunun kontrol etmemiz gerektiğini söylediği şeyler
- Test senaryoları “bilinmeyen bilinmeyenleri” kontrol etmez
 - Hiç kimsenin test senaryolarına dahil etmeyi düşünmediği şeyler
- Manuel test “bilinmeyen bilinmeyenleri” kontrol edebilir
 - İnsanlar, belirtilmemiş olsa bile bir test senaryosu yürütürken bazı şeyleri fark eder

Çözüm: Her iki yöntemi karıştırın

- Pek çok takım hem manuel hem de otomatik yöntemleri kullanır
- Otomatik testler genelde manuel testlerden çok daha fazladır
 - Pek çok işi otomatik testler yapar
 - Manuel testler otomatik testlerin bir şeyi kaçırıp kaçırmadığını kontrol eder
 - Ör. GUI layout'ta doğru görünüyor mu?
 - Ör. Uygulama fonksiyonları kullanıcı bakış açısı ile iyi çalışıyor mu?
- Bazen full otomasyon mümkün olabilir
 - Özellikle uygulama GUI içermiyorsa
 - Google ve diğer büyük şirketler genellikle, insan incelemesi olmadan deploy edebilecekleri yeterince kapsamlı test paketlerine sahiptir.

Otomatik Test Senaryosu Yazımı

- Otomatik test senaryoları da manuel test senaryoları ile aynıdır.
 - TANIMLAYICI
 - AÇIKLAMA
 - ÖN KOŞULLAR
 - ÇALIŞTIRMA ADIMLARI
 - SON KOŞULLAR
- Aralarındaki fark, daha spesifiktir
 - Bilgisayarlar tam olarak ne yapacağını belirtir

Otomatik Test: Kara Kutu veya Saydam Kutu?

- Cevap: Her ikisi de!
- Kara Kutu Testi: Uygulama arayüzünü kullanarak sistem seviyesinde test
 - Script farklı komut satırı komutları ile uygulamayı çalıştırır
 - Script uygulama ile etkileşime geçer ve kullanıcı girdileri vererek program çıktısını inceler
 - GUI uygulamaları için bile script vardır: Daha sonra söz edeceğimiz araçlardan olan *Selenium*
- Saydam Kutu Testi: (genel olarak) birim seviyesinde testlerdir
 - Script doğrudan kaynak koddaki metotları parametrelerle çalıştırır
 - Script her metodun dönen değerini ve/veya son koşulunu kontrol eder
 - Genellikle framework'e bağımlıdır: *JUnit* (Java testi için framework)

Textbook Chapter 12'yi okuyunuz