

Sertifikalı Test Uzmanı
Temel Seviye Ders Programı(Versiyon 2018)
Ders Notları

Önsöz

Hazırlanan ders notları ISTQB® Sertifikalı Test Uzmanı Temel Seviye Ders Programının 2018 Versiyonu çalışmaları sonrasında oluşturulmuştur. Bu notların amacı yoğun bilgi barındıran bir dokümanı daha az sayfada ve dikkat çekici bir şekilde oluşturma gayretidir.

Ders notları hazırlanırken daha önce çıkmış bazı dönem soruları ve örnek sorular incelenerek önemli kısımlar atlanmamaya çalışılmıştır. Yine de ders notlarının geliştirilmesi amacıyla görüş ve düşüncelerinizi mail adresine paylaşarak bu ders notlarının geliştirilmesine yardımcı olabilirsiniz.

Daha ayrıntılı bir şekilde sınava hazırlanmak için ders notları sonrası <https://www.turkishtestingboard.org/> adresinde yer alan yayınlardan Sertifikalı Test Uzmanı Temel Seviye Ders Programı ve ISTQB Yazılım Testi Terimler Sözlüğü tercih edilebilir.

Ders notları temel seviye ders programındaki gibi aşağıdaki bölümlerden oluşmaktadır:

Yazılım Testinin Temelleri - Yazılım Geliştirme Yaşam Döngüsü Boyunca Testler - Statik Testler - Test Teknikleri - Test Yönetimi - Yazılım Testleri için Araç Desteği

Hazırlanan bu ders notlarının kaynağı ISTQB'nin bu ders programının kaynağı ISTQB® Sertifikalı Test Uzmanı Temel Seviye Ders Programının 2018 Versiyonudur.

1.YAZILIM TESTİNİN TEMELLERİ

Yazılım Testi nedir?

1- Yazılım testi, yazılımın kalitesini değerlendirmenin ve kullanım sırasında oluşabilecek

yazılım hatası riskini azaltmanın bir yoludur.

2- Yazılım testi birçok farklı faaliyeti içeren bir süreçtir.

3- Test süreci aynı zamanda test planlama, test analizi, test tasarımı, testin uyarlanması,

test ilerlemesini ve sonuçlarını raporlama ve bir test nesnesinin kalitesini değerlendirme gibi faaliyetleri içerir

Test Etme ve Hata Ayıklama

•Test etme ve hata ayıklama farklı aktivitelerdir.

•Testlerin yürütülmesi, yazılımdaki hataların neden olduğu arızaları gösterebilir

•Hata ayıklama ise arızaların arkasında yatan hataları bulan, analiz eden ve

düzelten yazılım geliştirme faaliyetidir.

Test Prensipleri

1. Testin amacı, yazılımda hataların olduğunu göstermektir; yazılımda hata kalmadığını ispatlamak değildir.
2. Yazılımı %100 test etmek imkansızdır.
3. Erken test, zaman ve para tasarrufu sağlar.
4. Hatalar yazılımın belli alanlarında yoğunlaşır.
5. Antibiyotik direnci
6. Yazılım testi, projenin bağlamına, koşullarına göre değişiklik gösterir.
7. Yeni hata bulamıyoruz, başarılı bir yazılım elde ettik yanılgısı

Hataların kök nedenleri, hataların oluşmasına sebep olan en baştaki eylemler veya koşullardır. Hatalar, kök nedenlerini belirlemek için analiz edilebilir, böylece gelecekte benzer hataların ortaya çıkma olasılığının azaltılması hedeflenir.

İnsanların yaptığı hataların nedenleri:

- Zaman baskısı
- İnsani yanılma payı
- Deneyimsizlik veya yetersiz yetkinlik
- Gereksinimler ve tasarım ile ilgili yanlış iletişim de dâhil olmak üzere proje ekip üyeleri arasındaki yanlış iletişim
- Kodun, tasarımın, mimarinin, çözülecek temel problemin ve/veya kullanılan teknolojilerin karmaşıklığı
- Özellikle sistem içi ve sistemler arası etkileşimlerin sayısının fazla olduğu durumlarda, sistem içi ve sistemler arası arayüzler hakkındaki yanlış anlaşılımlar
- Yeni, henüz tecrübe edilmemiş teknolojiler

Antibiyotik direnci

Aynı testler sürekli tekrar edilirse, en sonunda bu testler artık yeni hatalar bulamamaya başlar. Yeni hataları bulmak için mevcut testler ve test verilerinin değiştirilmesi ve yeni testlerin yazılması gerekebilir (Tıpkı sürekli antibiyotik kullanımında bir süre sonra aynı çeşit antibiyotiğin artık bakterileri öldürmede etkili olmaması, bakterilerin direnç kazanması gibi).

•Kalite Yönetimi

- Bir kurumu kalite açısından yönlendiren ve kontrol eden tüm faaliyetleri içerir.

•Kalite Güvence

- Uygun kalite seviyelerinin elde edilmesini destekleyen test faaliyetleri de dâhil olmak üzere çeşitli faaliyetleri içerir.

•Kalite Kontrol

- Kalite güvence, tüm sürecin doğru şekilde yürütülmesini hedeflediğinden testlerin doğru şekilde yapılmasını desteklemektedir.

üreten bir eylem.
ileyen kusur. (örn.

en olabilir.

Test Süreci Ana Aktivite Grupları
*Test planlaması, test hedeflerini yerine getirme yaklaşımıdır.
*Test gözetimi, planlanan ile gerçekleşen ilerlemenin sürekli karşılaştırılmasını içerir.
*Test kontrolü, test planında belirlenen hedeflere ulaşmak için gerekli önlemlerin alınmasını içerir.
*Test analizi, ölçülebilir kapsama kriterleri açısından “neyin test edileceğini” belirler.
*Test tasarımı, “nasıl test edilecek?” sorusuna cevap verir.
*Test uyarılama, “şu anda testleri koşturmak için gerekli şeylere sahip miyiz?” sorusuna cevap arar.
*Test koşturumu, sırasında, test grupları test koşturma çizelgesine uygun olarak çalıştırılır.
*Test tamamlama aktiviteleri, bir yazılımın piyasaya sürülmesi, bir test projesinin
tamamlanması, bir çevik proje döngüsünün tamamlanması,
bir test seviyesinin tamamlanması veya bir bakım sürümünün tamamlanması gibi proje
kilometre taşlarında gerçekleştirilir.

• Test Gözetim ve Kontrolü

• Test Analizi

• Test Tasarımı

• Test Uyarılama

•Test Koşumu

• Test Tamamlama

Şekil- Test Süreci Ana Aktivite Grupları

Test Esası ve Test Çalışma Ürünleri Arasında İzlenebilirlik

İyi bir izlenebilirlik aşağıdakileri de destekler:

- * Test kapsamının değerlendirilmesine
- * Değişikliklerin etkisinin analiz edilmesi
- * Testlerin denetlenebilir şekilde yapılması
- * BT yönetim kriterlerinin karşılanması
- * Test esastaki unsurların durumunu (ör. testleri geçen gereksinimler, testlerde başarısız olan gereksinimler ve bekleyen testleri olan gereksinimler) içerecek şekilde test ilerleme raporlarının ve test özet raporlarının anlaşılabilirliğinin artırılması
- * Testlerin teknik yönlerinin, anlaşılır şekilde paydaşlara aktarılması
- * İş hedeflerine göre ürün kalitesinin, süreç kapasitesinin ve proje ilerlemesinin değerlendirilmesi için bilgi sağlanması

İnsan Psikolojisi ve Test

Hataların ve arızaların bulunması, yazılım ürününün ve yazılım ürününü geliştirenin eleştirilmesi olarak algılanabilir.

Psikolojide yer alan “doğrulama sapması” tezinin göre, mevcut görüş ve inançlara uymayan bilgilerin kabul edilmesi kolay değildir.

Doğrulama sapmasının yanında diğer bilişsel eğilimler insanların testlerde elde edilen bilgileri anlamalarını veya kabul etmelerini zorlaştırabilir.

Test uzmanlarının iyi iletişim kurmasına aşağıdaki örnekler verilebilir:

- * Savaşarak değil iş birliği yaparak başlayın.
- * Herkese daha iyi kalitede yazılımlar geliştirmek olan ortak amacınızı hatırlatın.
- * Testin faydalarını vurgulayın.
- * Hatalı öğeyi oluşturan kişiyi eleştirmeden, test sonuçlarını ve diğer bulguları tarafsız ve gerçekçi bir şekilde iletin.
- * Objektif ve gerçeklere dayalı hata raporları yazın ve bulguları gözden geçirin.
- * Karşınızdaki kişinin nasıl hissettiğini ve verdiğiniz bilgilere olumsuz tepki vermesinin sebeplerini anlamaya çalışın.
- * Karşınızdaki kişinin söylediğiniz şeyi anladığını ve kendinizin de onun söylediği şeyi anladığınızı doğrulayın.

Bir test uzmanının düşünce tarzı merak, profesyonel karamsarlık, eleştirel bir bakış açısı, ayrıntılara dikkat, iyi ve pozitif iletişim ve ilişkiler için motivasyon içermelidir. Bir test uzmanının düşünce tarzı, test uzmanı deneyim kazandıkça gelişme ve olgunlaşma eğilimindedir.

Bir yazılımcının düşünce tarzı, bir test uzmanının düşünce tarzının bazı unsurlarını içerebilir, ancak yazılımcılar, genellikle çözümler tasarlamak ve oluşturmakla ilgilidir, bu çözümlerde neyin yanlış olabileceğini fazla düşünmez. Ek olarak, psikolojideki doğrulama sapması da kendi çalışmalarındaki hataları bulmalarını zorlaştırır.

2.YAZILIM YAŞAM DÖNGÜSÜ BOYUNCA TEST

Bir yazılım geliştirme yaşam döngüsü modeli, yazılım geliştirme projesinin her aşamasında gerçekleştirilen aktivite çeşitlerini ve aktivitelerin birbirleriyle mantıksal ve kronolojik olarak nasıl ilişkili olduğunu açıklar.

2.1.1 Yazılım Geliştirme ve Yazılım Testi

Bu dokümana göre yaygın yazılım geliştirme yaşam döngüsü modelleri aşağıdaki sınıflara ayrılır:

- Sıralı yazılım geliştirme modelleri(Doğrusal, sıralı bir faaliyetler akışı)
- Döngüsel ve artımlı yazılım geliştirme modelleri

A) Şalele modelinde, yazılım geliştirme faaliyetleri (örneğin: gereksinim analizi, tasarım, kodlama, testler) birbiri ardına tamamlanır. Bu modelde, test aktiviteleri ancak diğer tüm yazılım geliştirme aktiviteleri tamamlandıktan sonra gerçekleştirilir.

B) V-modeli, test sürecini yazılım geliştirme sürecinin tamamına entegre ederek erken test prensibini uygular. Ayrıca, V-modeli, karşılık gelen her yazılım geliştirme aşaması ile ilgili test seviyeleri içerir; bu da erken testi desteklemektedir. Bu modelde, her test seviyesiyle ilişkili testlerin yürütülmesi sıralı olarak ilerler, ancak bazı durumlarda kesişmeler gerçekleşir.

Sıralı yazılım geliştirme modelleri tüm özellikleri tamamlanmış bir yazılım oluşturulduktan sonra yazılımın paydaşlar ve kullanıcılarla paylaşılmasını hedefler, bu yüzden de yazılımın canlıya alınması için aylar veya yıllar gerekebilir.

1.Sıralı bir yazılım geliştirme modeli, yazılım geliştirme sürecini doğrusal, sıralı bir faaliyetler akışı olarak tanımlar. Bu, yazılım geliştirme sürecinde her aşamanın önceki aşama tamamlandığında başlaması gerektiği anlamına gelir. Teorik olarak aşamaların kesişimi yoktur, ancak uygulamada takip eden aşamadan erken geri bildirim almak faydalıdır.

2.Döngüsel yazılım geliştirme modeli, ele alınan yazılım özelliklerinin genellikle sabit bir süreye sahip bir dizi döngüde belirlenmesi, tasarlanması, oluşturulması ve birlikte test edilmesi anlamına gelir. Son yazılım teslim edilinceye veya yazılım geliştirme durdurulana kadar her döngü, genel yazılım özellikleri kümesinin büyüyen bir alt kümesi olan çalışan yazılımı sunar.

3.Artımlı yazılım geliştirme modelleri ise, parçalar halinde gereksinimleri belirlemeyi, sistemi tasarlamayı, oluşturmayı ve test etmeyi içerir; bu, yazılımın özelliklerinin adım adım artması anlamına gelir. Bu artışlarda ele alınacak yazılım özelliklerinin boyutu değişiklik gösterebilir, bazı modellerde büyük parçalar, bazılarında ise daha küçük parçalar bulunur. Ele alınacak bir yazılım özelliği, yeni bir sorgu seçeneği veya kullanıcı arayüzü ekranında yapılan tek bir değişiklik kadar küçük olabilir.

Test seviyeleri, birlikte düzenlenen ve yönetilen test aktivite gruplarıdır.
Test seviyeleri, yazılım geliştirme yaşam döngüsü içindeki diğer aktivitelerle bağlantılıdır. Bu dokümanda kullanılan test seviyeleri aşağıda verilmiştir:

- Kabul Testleri
- Sistem Testleri
- Entegrasyon Testleri
- Birim Testleri

Şekil- Test Seviyeleri

2.Entegrasyon Testi

Entegrasyon testleri, birimler veya sistemler arasındaki etkileşimlere odaklanır.

- Arayüzlerin kalitesine dair güvenin oluşturulması,
- Arayüzlerin kendisinde, birimlerde veya sistemlerde hataların bulunması,
- Hataların daha üst test seviyelerine kaçmasının önlenmesi gibi hedefleri vardır.

Birim testlerinde olduğu gibi bazı durumlarda otomatikleştirilmiş entegrasyon regresyon testleri, değişikliklerin mevcut arayüzleri, birimleri veya sistemleri bozmadığına dair güvence sağlar.

Birim entegrasyon testleri genellikle yazılımcıların sorumluluğundadır. Sistem entegrasyon testleri ise genellikle test uzmanlarının sorumluluğundadır.

3.Sistem Testi

Sistem testleri, bütün bir sistemin veya ürünün davranış ve yeteneklerine odaklanır ve genellikle sistemin gerçekleştirebileceği uçtan uca işleri (fonksiyonallite) ve bu işleri gerçekleştirirken gösterdiği fonksiyonel olmayan davranışları ele alır.

- Sistemin tamamlandığının ve beklendiği gibi çalışacağının sağlanmasının yapılması
- Bir bütün olarak sistemin kalitesine dair güven oluşturulması gibi hedefleri vardır.

Sistem test ortamı ideal olarak canlı ortama karşılık gelmelidir.

Sistem testlerini genellikle bağımsız test uzmanları gerçekleştirir.

İfied Pro
lim özell

döngü g
aç gelişt

ılt uzunl
bir kere

prototipleme): Görsel veya deneysel yazılım özellikleri oluşturulabilir veya bunlardan vazgeçilebilir.

ğilimind
şekilde ik

in, birkaç
gibi, aynı

dıktan sc
grup hali

4.Kabul Testi

Kabul testleri de sistem testleri gibi genellikle bütün bir sistemin veya ürünün davranışına ve yeteneklerine odaklanır.

- Bir bütün olarak sistemin kalitesine dair güven oluşturulması
- Sistemin tamamlandığının ve beklendiği gibi çalışacağının sağlanmasının yapılması gibi hedefleri vardır.

Kabul testleri genellikle sıralı bir yazılım geliştirme yaşam döngüsündeki en son test seviyesi olarak düşünülür, ancak başka zamanlarda da gerçekleştirilebilir.

Kabul testleri genellikle müşterilerin, kullanıcıların, ürün sahiplerinin veya sistem operatörlerinin sorumluluğundadır ve diğer paydaşlar da bu sürece katılabilir.

m

gi

aşamalarda

Bu yöntemlerin kullanıldığı birçok yazılım geliştirme çalışması kendi kendini yöneten ekipler kavramını da içerir; bu da test çalışmasının düzenlenme şeklini ve test uzmanları ve yazılımcılar arasındaki ilişkiyi değiştirebilir.

Bu yöntemler, büyüyen bir sistem oluşturur; bu yöntemlerde canlıya alma yazılım özelliği bazında, döngü bazında veya daha klasik bir ana sürüm bazında olabilir. Yazılım özelliklerinin canlıya alınıp alınmadığına bakılmaksızın, yazılım büyüdükçe regresyon testleri giderek önem kazanır.

Sıralı modellerin aksine, döngüsel ve artımlı modeller haftalar veya hatta günler içinde kullanılabilir yazılımlar sunabilir, ancak gereksinimlerin tamamını içeren tam bir ürün setinin sunulması aylar, hatta yıllar sürebilir.

1.Fonksiyonel Testler

Fonksiyonel testleri, sistemin gerçekleştirilmesi gereken fonksiyonları değerlendiren testleri içerir.

Fonksiyonel gereksinimler; iş gereksinimleri, kullanıcı gereksinimleri, epikler, kullanıcı hikâyeleri, kullanım senaryoları veya fonksiyonel spesifikasyonlar gibi çalışma ürünlerinde tanımlanmış olabilir.

Fonksiyonlar gereksinimler sistemin "ne yapması" gerektiğini tanımlar.

2.Fonksiyonel Olmayan Testler

Bir sistemin fonksiyonel olmayan testleri, sistemlerin ve yazılımların, kullanılabilirlik, performans veya güvenlik gibi özelliklerini değerlendirir.

Fonksiyonel olmayan testler sistemin yapılması gerekenleri **"ne kadar iyi"** yaptığını ölçülemeye çalışır. Tüm test seviyelerinde ve sıkça gerçekleştirilmeli ve mümkün olduğunca erken yapılmalıdır.

3.Beyaz Kutu Testleri

Beyaz kutu testleri, sistemin inşaatına dayanan testleri oluşturur. İçerik; kod, mimari, iş akışları ve/veya sistem içindeki veri akışlarını içerebilir.

Beyaz kutu testlerinin bütünlük derecesi yapısal kapsam ile ölçülebilir. Yapısal kapsam, bir yapısal öğenin testlerle ne ölçüde ele alınmış olduğunu belirtir ve kapsanan öğenin yüzdesi olarak ifade edilir.

Beyaz kutu test tasarımı ve koşumu, kodun yazılma şekli (örneğin kod kapsamı araçlarını kullanmakt) ve/veya kodun nasıl çalıştığıdır.

4. Değişiklikle İlgili Testler

Bir sistemde, bir hatayı çözmek için veya yeni veya değişen fonksiyonallite nedeniyle değişiklikler yapıldığında, değişikliklerin hatayı çözdüğünü veya fonksiyonalliteyi doğru bir şekilde hayata geçirdiğini ve öngörülemeyen olumsuz sonuçlara neden olmadığını doğrulamak için testler yapılmalıdır.

A) Onaylama testleri: Bir hata çözüldükten sonra, hata nedeniyle başarısız olmuş tüm test senaryoları tekrar test.

inin amacı, asıl
p
ktır.

odun bir
leğişikliğinin
mlerinin
şekilde
tenmeyen yan

etkileri regresyon denir. Regresyon testleri, bu gibi istenmeyen yan etkileri bulmak için yapılan testleri içerir.

Onaylama testleri ve regresyon testleri tüm test seviyelerinde yapılabilir.

el
ina

ellikle

miş bir
göre

Test çeşidi, belirlenmiş test hedeflerine dayanarak bir yazılımın belirli özelliklerini veya bir sistemin bir bölümünü test etmeyi amaçlayan bir test aktiviteleri grubudur.

- Bütünlük, doğruluk ve uygunluk gibi **fonksiyonel** kalite özelliklerinin değerlendirilmesi,
- Güvenilirlik, performans, güvenlik, uyumluluk ve kullanılabilirlik gibi **fonksiyonel olmayan** kalite özelliklerinin değerlendirilmesi,
- Birim veya sistemin **yapısının veya mimarisinin** doğru, eksiksiz ve gereksinimlerde belirtildiği gibi olup olmadığının değerlendirilmesi,
- **Değişikliklerin etkilerinin değerlendirilmesi:** hataların giderildiğini onaylama (onaylama testleri) ve düzeltilen hatanın veya yapılan değişikliğin istenmeyen değişiklikleri tetikleyip tetiklemediğini bulma (regresyon testleri) gibi.

kurumlarca tabanlı (örneğin hastane yönetim sistemi kullanıcı grupları) hakkında bilgiler gibi özel beceriler veya bilgiler yer alabilir.

güçlendirme çalışmaları için uygun potansiyel veya mevcut müşteriler ve/veya operatörler veya bağımsız bir test ekibi tarafından gerçekleştirilir.

4b-Beta testleri ise potansiyel veya mevcut müşteriler ve/veya operatörler tarafından kendi ortamlarında yapılır.

Yaygın k
1-Kullan
Genellikl
ortamda
uygunluğ
2-Opera
Operato
simüle e
3-Sözleş
Sözleşm
yazılım ü
gerçekle
4-Alfa ve

- Sistem entegrasyonu testleri,

Birim ve sistem entegrasyon testleri entegrasyonun kendisine odaklanmalıdır.

Statik Testin Faydaları

- Testlerin maliyet ve süresinin düşürülmesi

- Yaşam döngüsünün kısımlarında veya daha az arıza olması kullanım ömrü boyunca olan toplam kalite düşürülmesi

- Gereksinimlerdeki tutarsızlıkları, belirsizlikleri, çelişkileri, çıkarmaları, yanlışlıkları ve fazlalıkları ortaya çıkararak tasarımdaki veya kodlamadaki hataların önlenmesi

- Gözden geçirmeler sırasında ekip üyeleri arasındaki iletişimin iyileştirilmesi

Statik ve Dinamik Testler Arasındaki Farklar

3.Statik Testler

Yazılımın veya diğer çalışma ürünlerinin manuel incelenmesine (gözden geçirmelere) veya kodun veya diğer çalışma ürünlerinin araç kullanılarak değerlendirilmesine (statik analizlere) dayanır. Her iki statik test çeşidi de test edilen kod veya çalışma ürününü gerçekten çalıştırmadan değerlendirir.

Her türlü çalışma ürünü statik testler kullanılarak incelenebilir. Örneğin: İş gereksinimleri, fonksiyonel gereksinimler ve güvenlik gereksinimleri dâhil olmak üzere tüm gereksinim çeşitleri, epikler, kullanıcı hikâyeleri ve kabul kriterleri, mimari ve tasarım gereksinimleri, kod...

genellikle dışarıdan görülebilen davranışlara odaklanmasındır.

Dinamik testlere göre statik testler ile çalışma ürünlerinde yer alan bulunması ve düzeltilmesi daha kolay ve daha ucuz olan yaygın hatalar şunlardır:

Gereksinim hataları, tasarım hataları, kodlama hataları, standartlardan sapmalar, hatalı arayüz gereksinimleri, güvenlik açıkları, test esası izlenebilirliğinde veya kapsamında boşluklar veya yanlışlıklar.

Ayrıca, çoğu sürdürülebilirlik hatası sadece statik testler (örneğin, hatalı modülerleştirme, birimlerin tekrar kullanılabilirliğinin düşük olması, analiz edilmesi ve değiştirilmesi zor olan kodlar) ile bulunabilir.

Gözden Geçirme Süreci

Gözden geçirmeler, resmi veya gayri resmi olabilir.

etmez ve

mi, gözden
nenin

Bakım Testleri

Üretim ortamlarına alındıktan sonra yazılım ve sistemlerin bakımının yapılması gerekir. Herhangi bir değişiklik yapıldığında, hem değişikliklerin ne kadar başarılı olduğunu değerlendirmek hem de sistemin değişmeyen kısımlarındaki olası yan etkileri kontrol etmek için bakım testleri yapılmalıdır. Değişiklik, bir platformdan diğerine taşıma, kullanımdan kaldırma gibi nedenler bakım için tetikleyicidir.

Bakım için Etki Analizi

Etki analizi, bir değişikliğin beklenen ve olası yan etkilerini, istenilen sonuçların elde edilip edilemeyeceğini ve değişiklikten etkilenecek alanları belirlemek amacıyla yapılır.

Gözden Geçirme Çeşitleri

1. Gayri resmi gözden geçirme (örneğin; çalışma arkadaşının kontrol etmesi, eşleştirme, eşli gözden geçirme)

- Resmi (dokümanite edilmiş) bir sürece dayanmaz.
- Sonuçlar dokümanite edilebilir.
- Kontrol listelerinin kullanımı isteğe bağlıdır.

2. Üzerinden geçme

- Gözden geçirme toplantısı genellikle çalışma ürününün yazarı tarafından yönetilir.
- Yazıcının olması zorunludur.

- Kontrol listelerinin kullanımı isteğe bağlıdır.

3. Teknik gözden geçirme

- Gözden geçiriciler yazarın teknik olarak dengi olan diğer veya aynı disiplinlerdeki teknik uzmanlardan seçilmelidir.

4. Test Tasarım Teknikleri

Test tekniğinin amacı, test koşullarını, test senaryolarını ve test verilerini belirlemeye yardımcı olmaktır.

Kara kutu test teknikleri, test nesnesinin iç yapısını dikkate almadan test nesnesinin girdi ve çıktılarına odaklanır.

Beyaz kutu test teknikleri (yapısal veya yapıya dayalı teknikler olarak da bilinir), test nesnesinin mimarisinin, ayrıntılı tasarımının, iç yapısının veya kodunun analizine dayanır

Tecrübeye dayalı test teknikleri, testlerin tasarlanması, uyarlanması ve koşuturulması için yazılımcıların, test uzmanlarının ve kullanıcıların tecrübelerini kullanır.

oluşturulur.

4. Teftiş

- Kurallara ve kontrol listelerine dayanarak, resmi, dokümanite edilmiş bir süreci izler.
- Gözden geçirme toplantısından önce bireysel hazırlık gereklidir.
- Gözden geçiriciler çalışma ürünü yazarının dengi veya çalışma ürünü ile ilgili diğer disiplinlerde uzman kişilerdir.
- Belirlenen giriş ve çıkış kriterleri kullanılır.
- Yazıcı olması zorunludur.
- Gözden geçirme toplantısı eğitimli bir moderatör tarafından yapılır.
- Yazar; gözden geçirme lideri, okuyucu veya yazıcı olarak görev yapamaz.
- Potansiyel hata kayıtları ve gözden geçirme raporu oluşturulur.

Gözden Geçirme Tekniklerinin Uygulanması

Hataları bulmak için bireysel gözden geçirme (bireysel hazırlık) faaliyeti sırasında uygulanabilecek birkaç gözden geçirme tekniği vardır.

1. Kurgusuz

Çok az hazırlık gerektiren ve yaygın kullanılan bir tekniktir. Bu teknik büyük ölçüde gözden geçiricilerin becerilerine bağlıdır ve farklı gözden geçiriciler tarafından bildirilen birçok benzer sorunun bulunmasına yol açabilir.

2. Kontrol listesine dayalı

Gözden geçiricilerin gözden geçirme başlangıcında (örneğin moderatör tarafından) dağıtılan kontrol listelerine dayanarak sorunları belirlediği sistematik bir tekniktir.

Gözden geçirme kontrol listesi deneyimle elde edilen ve olası hatalara dayanan bir dizi sorudan oluşur.

Dayalı tekniğin en önemli avantajı tipik hataları sistematik bir şekilde kapsamasıdır.

3. Provalar

Bu yaklaşım, çalışma ürününü nasıl okuyacaklarına katılımcılarla çalışma ürünü üzerinde "provalar" yaparak gözden geçiricilerin işini kolaylaştırır.

Bu teknik, çalışma ürününden etkilenen her bir paydaşın çalışma ürününü değerlendirdiği bir tekniktir.

Genellikle roller arasında farklı son kullanıcı grupları (deneyimli, deneyimsiz, kıdemli, çocuk vb.) ve kurumdaki belirli roller (kullanıcı yöneticisi, sistem yöneticisi, performans testi uzmanı vb.) yer alır.

5. Bakış açısına dayalı

Bu teknik de farklı paydaş bakış açılarını dikkate alır. Tipik paydaş bakış açıları arasında son kullanıcı, pazarlama, tasarımcı, test uzmanı veya kurum yönetimi yer alır.

Bakış açısına dayalı okuma ayrıca gözden geçiricilerin, çalışma ürününü kullanmaya çalışmasını da gerektirir.

Ayrıca, bakış açısına dayalı okumalarda kontrol listelerinin kullanılması beklenir.

Yapılan çalışmalar, bakış açısına dayalı okumanın gereksinimleri ve teknik çalışma ürünlerini "gözden geçirmek için en etkili teknik" olduğunu göstermiştir.

- Özellikle teftişler gibi daha resmi gözden geçirme çeşitleri için yeterli eğitim sağlanır.

- Çıktıların iletilmesi ve analizi
- Hataların giderilmesi ve raporlama

Beyaz Kutu Test Teknikleri

Beyaz kutu testleri, test nesnesinin iç yapısına dayanır. Beyaz kutu test teknikleri tüm test seviyelerinde kullanılabilir ancak bu bölümde açıklanan koda bağlı iki teknik en yaygın şekilde birim test seviyesinde kullanılır.

1.Komut Testi ve Kapsamı

Komut testi, kod içinde yer alan yürütülebilir komutların üzerinden geçilip bu komutların çalıştırılmasıdır. Kapsam, testler tarafından çalıştırılan komutların sayısının test nesnesindeki çalıştırılabilir komutların toplam sayısına bölünmesi ile ölçülür ve normalde yüzde olarak ifade edilir.

2.Karar Testi ve Kapsamı

Karar testi, koddaki kararların üzerinden geçilip bu kararların çalıştırılması ve karar çıktılarına dayanarak kodun test edilmesidir. Bunun için test senaryoları karar noktasındaki kontrol akışlarını takip eder (örneğin, bir IF komutu için, biri doğru çıktı ve biri yanlış çıktı; bir CASE komutu için tüm olası çıktıların test senaryoları gerekli olacaktır). Kapsam, testler tarafından çalıştırılan karar çıktılarının sayısının test nesnesindeki karar çıktılarının toplam sayısına bölünmesi ile ölçülür ve normalde yüzde olarak ifade edilir.

Komut ve Karar Testlerinin Önemi

%100 komut kapsama yüzdesi sağlandığında koddaki tüm çalıştırılabilir komutların en az bir kez test edilmesi sağlanır, ancak tüm kararların tamamen test edilmesi sağlanmaz. %100 karar kapsamı sağlandığında, tüm karar çıktıları çalıştırılır; buna, doğru çıktının ve yanlış çıktının test edilmesi de dahildir. %100 karar kapsamına ulaşılması %100 komut kapsamına ulaşılmasını garanti eder (ancak bunun tersi geçerli değildir).

Tecrübeye Dayalı Test Teknikleri

Tecrübeye dayalı test teknikleri uygulanırken, test senaryoları, test uzmanının beceri ve sezgilerinden ve benzer uygulama ve teknolojilerdeki tecrübelerinden elde edilir. Bu teknikler, diğer daha sistematik tekniklerle kolayca bulunamayan hataları bulmada yardımcı olabilir. Test uzmanının yaklaşımına ve tecrübesine bağlı olarak, bu teknikler çok çeşitli kapsam ve etkinlik derecelerine ulaşabilir. Kapsamın değerlendirilmesi zor olabilir ve bu tekniklerle ölçülemeyebilir.

1.Hata Tahminleme

3.1

Hata tahminleme, test uzmanının bilgilerine dayalı olarak yanlışların, hataların ve anızaların oluşmasını sağlamak için kullanılan bir tekniktir; bu bilgiler aşağıdaki gibidir: Uygulamanın geçmişte nasıl çalıştığı, yazılımcıların yapmaya eğilimli oldukları hata çeşitleri, uygulamalarda oluşan arızalar. Hata tahminleme tekniğine metodolojik bir yaklaşım, olası yanlışların, hataların ve arızaların bir listesini oluşturmak ve bu arızaları ve bunlara neden olan hataları ortaya çıkaracak testler tasarlamaktır.

2.Keşif Testi

Kar

bel kay tab yaz gırlı ak: tan

Kar kaş kar ser yaz kor içe kur say say noı

3.Kontrol Listesine Dayalı Testler

Kontrol listesine dayalı testlerde, test uzmanları bir kontrol listesinde bulunan test koşullarını kapsayacak şekilde testler tasarlar, uygular ve koşturur. Bu kontrol listeleri tecrübe, kullanıcı için neyin önemli olduğu veya yazılımın niçin ve nasıl başarısız olabileceği bilgisine dayanarak oluşturulabilir. Fonksiyonel ve fonksiyonel olmayan testleri desteklemek için kontrol listeleri oluşturulabilir. Ayrıntılı test senaryoları olmadığında, kontrol listesine dayalı testler rehberlik ve belirli bir yere kadar tutarlılık sağlayabilir. Bunlar üst seviye listeler olduğu için, gerçek testlerde bazı değişikliklerin ortaya çıkması muhtemeldir; bu da potansiyel olarak daha büyük kapsama ama daha düşük tekrarlanabilirliğe yol açar.

Kar hiç önl alır sağ ger bul tes koş bağı uyğ

Testte Bağımsızlık

Test Eforunu E

5.Test Yönetimi

Test eforu tahminlemesi, belirli bir proje, sürüm veya döngüdeki testlerin hedeflerini yerine getirmek için yapılması gereken testlerle ilgili çalışmanın miktarını öngörmektir. Test çalışmasını etkileyen faktörler arasında ürünün özellikleri, yazılım geliştirme sürecinin özellikleri, testi yapacak kişilerin özellikleri ve test sonuçları sayılabilir.

- Kurum içinde, proje yönetimi veya üst düzey yöneticilere bağlı çalışan bağımsız test ekibi veya grubunun olması
- Kurumdan veya kullanıcılardan seçilen test uzmanlarının

Bazı Tanımlar

Risk, gelecekte olumsuz sonuçlara yol açacak bir olayın gerçekleşme olasılığını içerir. Risk seviyesi, olayın olasılığı ve etkisi (zararı) ile belirlenir.

Ürün riski, bir çalışma ürününün (örneğin bir gereksinim, birim, sistem veya test) kullanıcılarının ve/veya paydaşlarının meşru ihtiyaçlarını karşılamaması ihtimalini içerir.

Kalite riski, bir ürünün belirli kalite karakteristikleriyle ilişkilendirildiğinde (örneğin, fonksiyonallık, güvenilirlik, performans, kullanılabilirlik, güvenlik, uyumluluk, sürdürülebilirlik ve taşınabilirlik), ürün risklerine kalite riskleri de denir.

Proje riski, gerçekleşmesi durumunda, projenin hedeflerine ulaşma imkânı üzerinde olumsuz etkiye sahip olabilecek durumları içerir.

kullanılması gibi

- **Süreç uyumluluk(veya standartlara uyumluluk)**
- Yönlendirmeli (veya danışarak)
- **Regresyon hassasiyetli:** Yazılımın mevcut çalışan özelliklerinin bozulmasını (regresyon) engelleme amacıyla hayata geçirilmektedir.
- **Tepkisel:** Bu tür test stratejilerinde, testler, test edilen birim veya sisteme ve test koşumu sırasında meydana gelen olaylara verilen tepkilerle şekillenir. Keşif testleri, tepkisel stratejilerde kullanılan yaygın bir tekniktir.

Test Tahminleme Teknikleri

Yeterince test etmek için gerekli olan eforu belirlemek amacıyla kullanılan bazı tahminleme teknikleri vardır. En sık kullanılan tekniklerden ikisi aşağıda verilmiştir:

- **Metrik bazlı teknik:** önceki benzer projelerin metriklerine veya tipik değerlere dayanarak test eforunu tahmin etme. Örneğin; hata giderme modelleri, (burndown tabloları (yapılacak işler tablosu))
- **Uzman bazlı teknik:** testi gerçekleştirecek kişilerin tecrübesine veya uzmanlara dayanarak test eforunu tahmin etme. Örneğin; . Wideband Delphi tahminleme tekniği, poker planlama(ekip üyelerinin kendi deneyimlerine dayanarak yapılacak iş için gereken çalışmayı tahmin ettikleri bir teknik)
- Bağımsız test uzmanları bir dar boğaz olarak görülebilir veya sürümdeki gecikmelerden sorumlu tutulabilir.
- Bağımsız test uzmanları (örneğin test nesnesi hakkında) bazı önemli bilgilere sahip olmayabilir.

Giriş Kriterleri ve Çıkış Kriterleri (Hazır ve Tamamlandı)

Test faaliyetinde giriş kriterleri (genellikle çevik yazım geliştirmede "Hazır" tanımı olarak adlandırılır), belirli bir test faaliyetinin gerçekleştirilmesi için gereken ön koşulları tanımlar.

Giriş kriterleri karşılanmazsa, faaliyetin daha zor olması, daha uzun sürmesi, daha maliyetli ve daha riskli olması muhtemeldir.

Çıkış kriterleri (genellikle çevik yazılım geliştirmede "Tamamlandı" tanımı olarak adlandırılır), bir test seviyesi veya bir test grubunu tamamlandı olarak tanımlamak için hangi koşulların sağlanması gerektiğini tanımlar.

Çıkış kriterleri karşılanmasa bile, bütçenin tükenmesi, planlanan süresinin tamamlanması ve/veya ürünün piyasaya sürülmesi için oluşan baskı nedeniyle test faaliyetlerinin kısa kesilmesi de yaygın bir durumdur.

Test Yöneticisi ve Test Uzmanının(Tester) Görevleri

Test yöneticisi, genel olarak test sürecinden ve test faaliyetlerine başarılı bir şekilde liderlik edilmesinden sorumludur. Test yöneticisi rolü; profesyonel bir test yöneticisi, proje yöneticisi, yazılım geliştirme yöneticisi veya kalite güvence yöneticisi.

HATA YÖNETİMİ

Testlerin hedeflerinden biri de hataların bulunması olduğu için testlerde bulunan hatalar raporlanmalı ve kaydedilmelidir.

Dinamik testler sırasında oluşturulan bir hata raporu genellikle aşağıdakileri içerir:

- Hatanın seri numarası veya sırası
- Bulunan hatanın başlığı ve kısa bir özeti
- Hata raporunun tarihi, düzenleyen kurum ve yazar kişi
- Test ögesinin (test edilen yapılandırma ögesi) ve ortamının tanımı
- Hata raporunun durumu (ör. açık, ertelenmiş, tekrarlanmış, düzeltilmeyi bekliyor, onaylama testleri bekleniyor, yeniden açıldı, kapatıldı)
- Sonuçlar, öneriler ve onaylar

Bazı test grupları, geliştirme sürecini belirleyen bir test koşum çizelgesinde düzenlenebilir.

Test koşum çizelgesi; önceliklendirme, bağımlılıklar, onaylama testleri, regresyon testleri ve testleri koşturmak için en verimli sıralama gibi faktörleri dikkate almalıdır.

Bazı durumlarda birçok test sıralaması mümkündür; bu sıralamaların verimlilik seviyeleri de farklılık gösterir. Bu gibi durumlarda, test koşum verimliliği ile önceliklendirme arasında bir denge kurulmalıdır.

Risk Bazlı Testler ve Ürün Kalitesi

Risk, test eforunu odaklamak için kullanılır. Testlere nerede ve ne zaman başlanacağını ve daha fazla dikkat gerektiren alanları belirlemek için kullanılır. Testler, olumsuz bir olay gerçekleşme olasılığını azaltmak veya olumsuz bir olayın etkilerini azaltmak için kullanılır.

Testler, belirlenen risklerin riskini azaltma ve onlar hakkında geri bildirim sağlama, kalan (ele alınamamış) riskler hakkında ise geri bildirim sağlamak için kullanılır.

Ürün risk analizi, ürün risklerinin belirlenmesi, belirlenen her bir risk için gerçekleşme olasılığı ve etkilerinin değerlendirilmesi aktivitelerini içerir.

Risk bazlı testler, ürün risk analizini gerçekleştirmek için proje paydaşlarının ortak bilgi ve sezgilerine dayanır. Üründe arıza olasılığının en aza indirilmesi için risk yönetimi faaliyetleri disiplinli bir yaklaşım sağlar.

Test Araçlarındaki Önemli Hususlar

Test araçları, bir veya daha fazla test faaliyetini desteklemek için kullanılabilir. Bu araçlar aşağıda verilmiştir:

- **Doğrudan testlerde kullanılan araçlar:** test koşum araçları ve test verisi hazırlama araçları gibi
- Gereksinimlerin, test senaryolarının, test prosedürlerinin, otomatikleştirilmiş test betiklerinin, test sonuçlarının, test verilerinin ve hataların yönetilmesine yardımcı olan ve test koşumunun raporlanması ve gözetimi için kullanılan araçlar
- Araştırma ve değerlendirme için kullanılan araçlar
- Testlerde yardımcı olan herhangi bir araç (bu anlamda Excel de bir test aracıdır)

Test Aracı Sınıflandırması

Testlerde araçların kullanılmasının potansiyel faydaları ve sunduğu fırsatların yanında bazı riskleri de vardır. Bu durum, özellikle test koşumu araçları (genellikle test otomasyonu olarak da adlandırılan) için geçerlidir.

Test koşumunu desteklemek için araç kullanmanın potansiyel faydaları aşağıda verilmiştir:

- Tekrarlanan manuel testlerde azalma dolayısıyla zamandan tasarruf
- Daha fazla tutarlılık ve tekrarlanabilirlik
- Daha objektif değerlendirme
- Testlerle ilgili bilgilere daha kolay erişim

Testleri desteklemek için araç kullanmanın potansiyel riskleri aşağıda verilmiştir:

- Araç için beklentiler gerçekçi olmayabilir.
- Bir aracın kullanılmaya başlanması için gereken zaman, maliyet ve efor olması gerekenden daha az tahmin edilebilir.
- Araçtan yüksek miktarda ve sürekli fayda elde etmek için gereken zaman ve efor olması gerekenden daha az tahmin edilebilir.
- Aracın ürettiği test varlıklarının bakımı ve güncel tutulması için gereken efor olması gerekenden daha az tahmin edilebilir.
- Araca çok fazla bağlı olunabilir.
- Test varlıklarının versiyon kontrolü ihmal edilebilir.
- Gereksinim yönetimi araçları, yapılandırma yönetimi araçları, hata yönetimi araçları gibi kritik önemdeki araçlar ve birden çok tedarikçinin araçları arasındaki ilişkiler ve birlikte çalışabilirlik sorunları gözden kaçırılabilir.
- Aracın tedarikçisi iflas edebilir, aracı desteklemeyi bırakabilir veya aracı farklı bir kuruma satabilir
- Tedarikçi; bakım, destek, yeni versiyonlar ve hata düzeltmeleri için yeterli desteği vermeyebilir
- Açık kaynak kodlu bir yazılımın geliştirilmesi geçici olarak durdurulabilir
- Yeni bir platform veya teknoloji test aracı tarafından desteklenmeyebilir
- Aracın net bir sahibi olmayabilir.

Test Araç Desteği

Test koşumu araçları

Test koşumu araçları, otomatikleştirilmiş test betikleri kullanarak test nesnelerini koşturur. Bu tür bir araç, genellikle belirgin bir fayda elde etmek için önemli ölçüde çalışma gerektirir.

Veri güdümlü test yaklaşımı, test girdilerini ve beklenen sonuçları, genellikle bir elektronik tablo şeklinde ayırır ve giriş verilerini okuyabilen ve aynı test betiğini farklı verilerle çalıştırabilen daha genel bir test betiği kullanır. Betik dilini bilmeyen test uzmanları daha sonra bu önceden tanımlanmış betikler için yeni test verileri yaratabilir.

Aksiyon kelimesi güdümlü test yaklaşımında, genel bir betik, gerçekleştirilecek işlemleri tanımlayan anahtar kelimeleri (bunlara aksiyon kelimeleri de denir) işler, ardından ilişkili test verilerini işlemek için anahtar kelime betiklerini çağırır. Test uzmanları (betik diline hakim olmasalar bile), anahtar kelimeleri ve ilgili verileri kullanarak testleri tanımlayabilirler, bunlar da test edilen uygulamaya göre uyarlanabilir.

Model Bazlı test (MBT) araçları, bir fonksiyonel gereksinimin, aktivite diyagramı gibi bir model biçiminde kaydedilmesine olanak sağlar. Bu görev genellikle bir sistem tasarımcısı tarafından gerçekleştirilir. MBT aracı, test senaryosu gereksinimlerini oluşturmak için modeli yorumlar; bu, daha sonra bir test yönetim aracına kaydedilebilir ve/veya bir test koşum aracı tarafından yürütülebilir.



