TEST MÜHENDİSİ NASIL OLMALI?İHTYACI VE SORUMLULUKLARI NELERDİR?

.Süreç Hakkında Bilgili Olmalı

.İletişimi Kolay kurmalı

.Yazılım İçerigini Bilmeli

.Yazılımcıyı Degil Yazılım Test Ettiğini UNUTMAMALIDIR

.Kibar Bİr Dil Kullanmalıdır +

Yazılım Testi Hata Olmadığını Değil Hata Olduğuğnu Göstermektir

Yazılım Testi Hatayı Bulur Nerede Olduğu İle İlgilenmez Bu Hata Ayıklama İşidir

İyi Test Standardı : IEEE 1008-1987

Test Grubu Yazılım Grubundan Bagımsız Olmalıdır

Test Bir Onaylama Mevkii Değildir

Test Hatanın Varlığını Kanıtlar

Detaylı Test İmkansızdır

Hata Kümelenmesi Diye Bir Kavram Vardır.Örnegin : Bir Yazılımcımız Piskolojik Olarak O gün İyi Degildir Bir hata Buldugumuzda O günki O kod Civarı İyi İncelenmelidir Çünkü Hata olasılığı Yüksektir

Test Bağlam Bagımlıdır yani Bir E-business ile Bir Askeri Proje Aynı Testlere Tabii Tutulamaz

Her Güncelleme Sonrası REGRESSYON TESTİ Yapılır

TEST Hiçbir Zaman Bitmez

Agile İterativdir Parçalara Bölerek Bitirmeyi Planlar.Her Bölüm Sonundada Test Yapılır.

Yazılımcının Yaptıgı Test Degil KENDİNİ SINAMADIR.Şayet Test Aktivitesi Sınırlı ve Bilinçli Kullanıcıların Kullandığı Program Değilse

Test Mühendisini YALNIZLAŞMA İHTİMALİ VARDIR Buna Ragmen Mantıklı ve Mantıksız İsteklerde Tarafsız Olmak Zorundadır +

Testte Sorular :

İstenilenler Yapılmışmı ->Validation(onaylama)

İşlevleri Yerine Getiriyormu->Verification(Doğrulama)

İstenilen Hızda Yapıyormu ->Performans

İstenilen Kadar İşlevi Yapıyor->Load (Yükleme)

İşlevi En Çok Ne kadar Yapıyor ->Stress

İstenileni Kolay Yapıyormu ->Usability

İstenileni Güvenle Yapıyormu->Realibity

İstenileni Herzmana Yapıyormu->Compatibility->Uygunluk +

Test Seviyeleri : Birim Test->Birim İntegrasyon Test->Sistem Testi ->Sistem İntegrasyon Testi->Kabul Testi

Entegrasyon Stratejileri: 1)Big Bang 2)Top-Down 3)Bottom -Up +

Testler : FONKSİYONEL ->BLACKBOX---->.............Durumunda Ne yapar Sorusuna Cevap Arar

FONKSİYONEL OLMAYAN-> Pek Çok Test-->...NasılYapar Sorusuna Yanıt Arar

YAPISAL TESTLER--->WHİTEBOX---->........Yapı Mimari Testidir (Code Coverage,Statement Covreage,Decision Coverage,Path Covreage)

REGRESYON TESTLERİ +

Code Coverage(Kod Kapsamı): Toplam Kodun Ne Kadarlık Kısımı Test Edidiğini Gösterir

Decision Coverage(Döngü Kapsamı): Çalıştırılan Döngülerin Tüm Dönüglere oranını Gösterir

Statement Coverage: Koddaki Durumları Çalıştırma Oranı

Test Coverage: Çalıştırılan Test Caselerin Toplam Caselere Oranını Verir +

STATİK KOD ANALİZİ : Kod Çalıştırılmadan Yapılır (Rewiew Kısımı Vardır Gözden Geçirme Kısmıdır Kodlar İncelenir Gözden Geçirilir.. )

Rewiew 2 Ayrılır BİREYSEL : DESK CHECK(Yazılımcı Kendi Yapar ) ,PROOF CHECK(Yazılımcı Bir Uzmanın Hazırladığı Dokuman İle Yapar)

GRUP : Rewiew,WALKTHROUGH(Toplantı Düzenlenir Kodu Yazan Kişi Satır Satır Kodun Üzerinden Geçer Anlatır ),İNSPECTİON(Uzman Ekip Kodları İnceler)

DİNAMİK KOD ANALİZİ : KOD Çalıştırılarak Yapılır +

Use Case : Bir Aktor (A) Bir Sistem Vardır (S) vardır. Senaryolar Üretilir .Eger Bu Senaryolar Sorunsuz Geçiliyorsa Buna HAPPY PATH(Mutlu Yol) Denilmektedir

+

CYCLOMATİC COMPLEXİTY: Kodun Ne Kadar Karmaşık Olduğunu Ölçmedir

BİRİM TEST ==UNİT TEST==MODÜL TEST== COMPONENT TESTİ (Aynı Anlamdadır)

STRUCTURAL TEST==WHİTEBOX TEST (Aynı Anlamdadır)

FONKSİYONEL ==BLACK BOX==OPAQUE==CLOSE (Aynı Anlamdadır)

DUMAN TESTİ== MANTIK TESTİ (Küçük Test Biter Büyük Test Başlar) Örnegin :Sistemi Tamamen Teste Başlatmadan Önce Bir Kaç Basit Fonksiyonu Yapması İstenir Saglarsa Teste Başalnır (EXİT CRİTERİA)

Conformance==Uygunluk Testi

Yenilenme Testi==Bir Hata Ne kadar Olunca Ne kadar Sürede Başa Dönebilir +

TDD(Test Driven Developmet) : Önce Testi Yazıp Daha Sonra Program Kodu Yazmaktır

BDD(Behaviour Driven Development): İşi Yaptırana Sunulan Test Diyebiliriz. Bu yüzden Herkesin Anlayacağı Dilde Yazılmalıdır Konuşma Dili Kullanılmalıdır .3 Durumu vardır GİVE,WHEN,THEN

örnegin Bir Dropbox HesabınaResim Yüklemek İşlemini Ele Alırsak ..GIVE: Dorpbox Hesabını Olması WHEN: Resim Yükleme İşlemi

THEN ise :Bu İşlemin Güvensiz Başarı İle Yapılması +

SELENİUM : Web Tabanlı Projelere Test İçin FİREFOX Tarafından Geliştirilmişitir

BUG(Böcek) Teriminin Bulunuşu ? Test Sırasında Hata Vardır Kısa Devreye Bir Böcek Neden Olduğu Farkedilmiş Ve Literatüre Hata Olarak Bug Denmiştir +

Agile İterativdir Parçalara Bölerek Bitirmeyi Planlar.Her Bölüm Sonundada Test Yapılır.

Yazılımcının Yaptıgı Test Degil KENDİNİ SINAMADIR.Şayet Test Aktivitesi Sınırlı ve Bilinçli Kullanıcıların Kullandığı Program Değilse

Yazılımcının Test Yapmasını avantajı Nedir? 1)Kodu İyi Bilir 2)Kaçırıdıgı Hatayı Kolay Görür Kolay Bulur

Yazılımcının Test Yapmasının Dezavantajı Nedir? 1)Kendi İşini Bozmak İstemez 2)Öznel Değerlendirir Hatayı Göremez +

Firma Dışı 3.Parti Firmalara Neden Test Yaptırmanın Avantajları ? 1)Bilgi Ve Deneyime Sahiplerdir 2)Tarafsız Değerlendirir

Firma Dışı 3.Parti Firmalara Neden Test Yaptırmanın DezAvantajları ? 1)Maliyeti Fazla 2)Ürün Hakkında Yapısı İşleyişi Bilgisi Yoktur 3)Deneyimini Paylaşmaz 4)Eğer Gizli veya Güvenliği

Yüksek Olan İşler Yapıyorsak Şirketimizin Bilgilerini Tekniklerini Test Danışmanları Sızdırabilir Başaka Sirketlere Ulaştırabilir

+

TEST SÜRESİ NASIL BELİRLENİR? TEST SÜRESİ=Test Case X Test İşlem Süresi /Testci Sayısı

HATA YAKALAMA ORANI NASIL BULUNUR? HATA YAKALAMA ORANI=Hata(testte bulunan)/Hata(testte bulunan) X Hata(Canlıda Bulunan) +

TEST PLANI NASIL OLMALI? 1)Her Plan Ayrıntıları ile Gayet Açık Bir Şeklide Yazılmalıdır Çünkü Ekibin Hepsi Bunu Kullanacağı ve Onlarla İletişimi Kurmaya Yardımcı Olacağı Düşünülmelidir

2) Plan Effort: Test İçin PlanlananHarcanacak Emek ACtual Effort: Gercekte Harcanan Emek Şayet ;

Plan Effort<Actual Effort ise Ya Testci Sayısı Artmalı Ya Test Case Azaltılmalı Yada Süre Artırılmalı