

**BAŐKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YAZILIM TEST SÜRECİ VE TMMİ MODELİNDE
PRİSMA YAKLAŐIMI UYGULAMASI**

DOĐA SERDAROĐLU

YÜKSEK LİSANS TEZİ

2015

**YAZILIM TEST SÜRECİ VE TMMi MODELİNDE
PRISMA YAKLAŞIMI UYGULAMASI**

**SOFTWARE TESTING PROCESS AND AN APPLICATION
OF PRISMA METHODOLOGY IN TMMi MODEL**

DOĞA SERDAROĞLU

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
BİLGİSAYAR Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2015

“Yazılım Test Süreci ve TMMi Modelinde PRISMA Yaklaşımı Uygulaması” başlıklı bu çalışma, jürimiz tarafından, 06/02/2015 tarihinde, **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan : Prof. Dr. İbrahim AKMAN

Üye (Danışman) : Prof. Dr. A. Ziya AKTAŞ

Üye : Doç. Dr. Hasan OĞUL

ONAY

.../02/2015

Prof. Dr. Emin AKATA
Fen Bilimleri Enstitüsü Müdürü

TEŐEKKÜR

Sayın Prof. Dr. A. Ziya AKTAŐ'a (tez danıőmanı), alıőmanın sonuca ulaőtırılmasında ve karőtılaőtılan gülklerin aőtılmasında her zaman yardımcı ve yol gsterici olduėu için...

OTI Holding İzmır ofis alıőanlarına, sre boyunca destek oldukları için...

Sınıf arkadaőtım Duygu Dede'ye, tm sorularıma yanıt verdiėi için...

ok deėerli arkadaőtım Serhat Yıldırım'a, her zaman yardımcı olduėu için...

Ve ok deėerli aileme, her zaman yanımda oldukları için...

ÖZ

YAZILIM TEST SÜRECİ VE TMMi MODELİNDE PRISMA YAKLAŞIMI UYGULAMASI

Doğa Serdarođlu

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışmanın asıl amacı yazılım test süreci için kullanılan TMMi(Test Maturity Model Integration) ve PRISMA(Practical Risk Based Testing) tekniklerinin anlaşılmasıdır.

Tez çalışmasına kısa bir giriş yapıldıktan sonra yazılımda kalite ve yazılımda test olguları gözden geçirilmiştir. Sonrasında PRISMA ile olan ilişkisi sebebiyle ISTQB(International Software Testing Qualifications Board) rehber dokümanı özetlenmiştir.

Uygulama olarak web tabanlı yeni geliştirilen bir yazılım üzerinde süreç uygulanmıştır. Geliştirme boyunca, yazılım test sürecinde detaylı bir araştırma yapılmıştır. Web tabanlı yazılım projesi üzerinde TMMi modeli 2. seviyede uygulanmıştır ve TMMi modelinin Test Planlama aşamasında PRISMA yaklaşımı gösterilmiştir.

ANAHTAR SÖZCÜKLER: Risk odaklı test yaklaşımı, test sürecinin iyileştirilmesi, PRISMA yaklaşımı, TMMi yaklaşımı, yazılımda kalite, yazılımda test.

Danışman: Prof.Dr. A. Ziya AKTAŞ, Başkent Üniversitesi, Bilgisayar Mühendisliği Bölümü.

ABSTRACT

SOFTWARE TESTING PROCESS AND AN APPLICATION OF PRISMA METHODOLOGY IN TMMi MODEL

Doğa Serdaroğlu

Başkent University Institute of Science

Department of Computer Engineering

The main objective of this study was to understand TMMi(Test Maturity Model Integration) and PRISMA(Practical Risk Based Testing) techniques for software testing.

In the thesis , after an introduction, software quality and testing are reviewed. Next, ISTQB(International Software Testing Qualifications Board) is summarized for its relationship in particular with PRISMA.

As an application , a web-based software was developed. During the development a detailed investigation on software testing processes, a web-based software project applying TMMi level 2 and PRISMA approach with in the Test Planning stage of TMMi was demonstrated.

KEYWORDS: Risk based testing approach, improving the testing process, PRISMA approach, TMMi approach, quality in software, testing in software.

Supervisor: Prof.Dr. A. Ziya AKTAŞ, Başkent University, Department of Computer Engineering

İÇİNDEKİLER LİSTESİ

SAYFA

| | |
|-----------------------------------------------------------------------------------------------------|-----------|
| ÖZ..... | i |
| ABSTRACT | ii |
| İÇİNDEKİLER LİSTESİ..... | iii |
| SİMGELER VE KISALTMALAR LİSTESİ | viii |
| ŞEKİLLER LİSTESİ..... | ix |
| TABLolar LİSTESİ | x |
| 1. GİRİŞ..... | 1 |
| 1.1 Literatür İncelemesi | 1 |
| 1.2 Tezin Amacı | 2 |
| 1.3 Tez Çalışmasının Yapısı | 3 |
| 2. YAZILIMIN KALİTESİ VE TESTİNE GENEL BAKIŞ | 4 |
| 2.1 Giriş..... | 4 |
| 2.2 Kaliteye Genel Bakış | 4 |
| 2.3 Yazılımda Kalite | 5 |
| 2.3.1 Genel | 5 |
| 2.3.2 CMMI | 8 |
| 2.3.3 Yazılımda kalite ölçümü ve yazılım metrikleri..... | 10 |
| 2.4 Yazılımda Test | 14 |
| 2.4.1 Genel | 14 |
| 2.4.2 Yazılımda test türleri | 16 |
| 2.5 Yazılımda Kalite ve Test İlişkisi | 21 |
| 3. ULUSLARARASI YAZILIM TEST YETERLİLİK KURULU'NUN YAZILIM TESTİ KILAVUZU ÖZETİ (ISTQB)..... | 22 |
| 3.1 Genel..... | 22 |
| 3.2 Testin Temelleri..... | 22 |
| 3.2.1 Yazılım sistemleri | 22 |
| 3.2.2 Hatalı yazılımların sebepleri | 22 |
| 3.2.3 Yazılım geliştirme ve bakım üzerinde testin rolü | 22 |
| 3.2.4 Ne kadar test yeterlidir? | 23 |
| 3.3 Yazılımda Test Nedir? | 23 |
| 3.3.1 Yazılımda testin yedi temel ilkesi | 24 |
| 3.3.2 Test süreci ve aşamaları | 25 |

| | |
|------------------------------------------------------------------|-----------|
| 3.3.3 Yazılım test seviyeleri | 28 |
| 3.3.4 Test türleri | 31 |
| 3.3.5 Statik teknikler | 33 |
| 3.3.6 Test tasarım teknikleri | 34 |
| 3.3.7 Test yönetimi | 36 |
| 3.3.8 Test planlaması ve değerlendirilmesi | 37 |
| 3.3.9 Test stratejileri ve test yaklaşımları | 38 |
| 3.3.10 Test raporlaması ve kontrolü | 39 |
| 3.3.11 Proje/Ürün riskleri ve test | 39 |
| 3.3.12 Test araçları | 40 |
| 4. TMMi (TEST MATURITY MODEL INTEGRATION) YAKLAŞIMI | 42 |
| 4.1 Giriş | 42 |
| 4.2 TMMi'nin Arka Planı ve Tarihçesi | 43 |
| 4.3 TMMi'nin Kapsamı | 43 |
| 4.3.1 Yazılım ve sistem mühendisliği | 44 |
| 4.3.2 Test seviyeleri | 44 |
| 4.3.3 TMMi ve CMMI | 44 |
| 4.3.4 TMMi için genel değerlendirme | 44 |
| 4.4 TMMi'nin Genel Yapısı | 45 |
| 4.5 TMMi'nin Bileşenleri | 46 |
| 4.5.1 Olgunluk seviyeleri | 46 |
| 4.5.2 Test süreç alanları | 46 |
| 4.6 Genel Amaçlar ve Genel Uygulamalar | 47 |
| 4.6.1 Sürecin planlanması | 48 |
| 4.6.2 Kaynakların sağlanması | 48 |
| 4.6.3 Sorumlulukların belirlenmesi | 48 |
| 4.6.4 Çalışanların eğitilmesi | 48 |
| 4.6.5 Konfigürasyon yönetimi | 49 |
| 4.6.6 Paydaşların belirlenmesi ve konuya dâhil edilmesi | 49 |
| 4.6.7 Sürecin izlenmesi ve kontrolü | 49 |
| 4.6.8 Üst düzey yönetim ile gözden geçirme yapılması | 49 |
| 4.7 CMMI Süreç Alanlarının TMMi için Desteklenmesi | 50 |
| 4.8 CMMI Süreç Alanları Doğrulama ve Onaylama | 50 |
| 4.9 TMMi Olgunluk Seviyeleri | 50 |

| | |
|----------------------------------------------------------------------|-----------|
| 4.9.1 Seviye 1:Başlangıç(Initial)..... | 53 |
| 4.9.2 Seviye 2:Yönetilen(Managed) | 53 |
| 4.9.3 Seviye 3:Tanımlanan(Defined)..... | 60 |
| 4.9.4 Seviye 4:Ölçülen (Management and Measurement) | 67 |
| 4.9.5 Seviye 5:İyileştirilmiş(Optimization)..... | 70 |
| 5. RISK FAKTÖRÜ VE ÜRÜN/PROJE RİSK YÖNETİMİ..... | 76 |
| 5.1 Risk Faktörüne Genel Bakış..... | 76 |
| 5.1.1 Test aktivitelerinin sınırlamaları..... | 79 |
| 5.1.2 Test sürecinin risk odaklı yürütülmesi | 80 |
| 5.2 Ürün Risk Yönetimi..... | 80 |
| 5.2.1 Risk yönetiminin temel aktiviteleri | 81 |
| 6. PRISMA(PRACTICAL RISK BASED TESTING) SÜRECİ..... | 98 |
| 6.1 Sürece Genel Bakış | 98 |
| 6.1.1 Ürün risk matrisi | 98 |
| 6.1.2 Farklı test seviyeleri | 99 |
| 6.1.3 Risklerin tanımlanması : Gereksinimler ve beyin fırtınası..... | 99 |
| 6.1.4 Risk analizi: Etki ve olasılık faktörleri | 100 |
| 6.2 Başlangıç Aşaması..... | 100 |
| 6.2.1 Standart sürecin tanımlanması..... | 100 |
| 6.2.2 Faktörlerin belirlenmesi | 101 |
| 6.2.3 Her faktör için ağırlıklandırma yapılması | 101 |
| 6.2.4 Puanlama aralıklarının belirlenmesi | 101 |
| 6.2.5 Kuralların belirlenmesi..... | 101 |
| 6.3 Planlama Aşaması | 102 |
| 6.3.1 Kapsamın tanımlanması | 102 |
| 6.3.2 Uyum faktörleri ve kuralların belirlenmesi | 102 |
| 6.3.3 Belgelerin bir araya getirilmesi | 103 |
| 6.3.4 Ürün risk maddelerinin tanımlanması..... | 103 |
| 6.3.5 Paydaşların tanımlanması ve seçilmesi | 105 |
| 6.3.6 Faktörlerin paydaşlara ayrılması | 106 |
| 6.4 Başlangıç Toplantısı Aşaması..... | 106 |
| 6.4.1 Sürecin açıklanması..... | 107 |
| 6.4.2 Ürünün açıklanması | 107 |
| 6.4.3 Anlaşmaya varılması..... | 108 |

| | |
|---------------------------------------------------------------------------------------------------|------------|
| 6.5 Ayrıntılı Risk Tanımlaması..... | 108 |
| 6.6 Bireysel Hazırlık Aşaması..... | 109 |
| 6.6.1 Her risk maddesi için faktörlerin puanlanması..... | 110 |
| 6.6.2 Tahminlerin belgelenmesi | 110 |
| 6.6.3 Kişisel gözden geçirme | 111 |
| 6.6.4 Test yönetimi..... | 111 |
| 6.7 Bireysel Puanların İşleme Alınması..... | 112 |
| 6.7.1 Puanlamanın kontrol edilmesi | 112 |
| 6.7.2 Bireysel puanların işleme alınması..... | 112 |
| 6.7.3 Konu listesi..... | 113 |
| 6.8 Paydaş Görüş Toplantısı | 114 |
| 6.8.1 Konu listesini işleme koymak | 115 |
| 6.8.2 Matris alanını analiz etmek | 115 |
| 6.8.3 Risk seviyelerini doğrulanması..... | 116 |
| 6.8.4 Toplantının özetlenmesi | 116 |
| 6.9 Farklılaştırılmış Risk Odaklı Test Yaklaşımı Tanımlamak | 117 |
| 6.9.1 Önceliklendirmenin yapılması | 118 |
| 6.9.2 Test tasarım tekniklerinin belirlenmesi | 119 |
| 6.9.3 Yenilenen geliştirme yaklaşımı..... | 121 |
| 6.10 PRISMA Sürecinin Avantajları, Dezavantajları ve Dikkat Edilmesi Gereken Ursular..... | 122 |
| 6.11 Ürün Risk Matrisi İçin Örnek Hesaplama..... | 124 |
| 7. TMMi VE PRISMA YAKLAŞIMLARININ TEST SÜRECİ ÜZERİNDE UYGULANMASI İÇİN BİR ÖRNEK..... | 129 |
| 7.1 GDS Tanımı(Global Distribution System)..... | 129 |
| 7.2 Problem Tanımı..... | 129 |
| 7.3 TMMi Yaklaşımı Süreç Alanlarının Uygulanması | 130 |
| 7.4 Test Politikası ve Test Stratejisi Süreç Alanının Uygulanması | 131 |
| 7.5 Test Planlaması Süreç Alanının Uygulanması | 132 |
| 7.6 PRISMA Yaklaşımının TMMi Süreci İçinde Uygulanması | 133 |
| 7.6.1 Başlangıç aşamasının uygulanması..... | 134 |
| 7.6.2 Planlama aşamasının uygulanması | 141 |
| 7.6.3 Başlangıç toplantısı aşamasının uygulanması | 145 |
| 7.6.4 Ayrıntılı risk tanımlaması aşamasının uygulanması | 146 |

| | |
|-------------------------------------------------------------------------|------------|
| 7.6.5 Bireysel hazırlık aşamasının uygulanması | 146 |
| 7.6.6 Bireysel puanların işleme alınması aşamasının uygulanması | 149 |
| 7.6.7 Paydaş görüş toplantısının yapılması | 151 |
| 7.6.8 Farklılaştırılmış risk odaklı test yaklaşımının uygulanması | 153 |
| 7.7 Testin İzlenmesi Ve Kontrolü Süreç Alanının Uygulanması | 156 |
| 7.8 Test Ortamı Hazırlık Süreç Alanının Uygulanması | 158 |
| 7.9 Test Tasarımı Süreç Alanının Uygulanması | 159 |
| 8. ÖZET VE SONUÇ | 163 |
| KAYNAKLAR LİSTESİ | 165 |

SİMGELER VE KISALTMALAR LİSTESİ

| | |
|--------|----------------------------------------------------|
| CMMI | Capability Maturity Model Integration |
| TMMi | Test Maturity Model Integration |
| PRISMA | Practical Risk Based Testing Approach |
| TPI | Test Process Improvement |
| ISTQB | International Software Testing Qualification Board |
| SEI | Software Engineering Institute |
| SQA | Software Quality Assurance |

ŞEKİLLER LİSTESİ

| | <u>Sayfa</u> |
|---------------------------------------------------------------------------------|--------------|
| Şekil 4.1 TMMi olgunluk seviyeleri | 52 |
| Şekil 5.1 Ürün risk yönetimi aktiviteleri | 82 |
| Şekil 5.2 Ürün risk matrisi örneği..... | 91 |
| Şekil 6.1 Ürün risk matrisi bölgeleri | 98 |
| Şekil 6.2 Risk maddelerinin yerleştirilmiş olduğu ürün risk matrisi örneği..... | 116 |
| Şekil 6.3 Puanlamalar sonucu oluşturulan ürün risk matrisi örneği | 128 |
| Şekil 7.1 Faktör tanımlama ve ağırlıklandırma ekranı | 140 |
| Şekil 7.2 Risk maddesi tanımlama ekranı | 143 |
| Şekil 7.3 Paydaş tanımlama ekranı | 144 |
| Şekil 7.4 Örnek puanlama tablosu..... | 148 |
| Şekil 7.5 Örnek ortalama puan tablosu | 150 |
| Şekil 7.6 Taslak ürün risk matrisi..... | 152 |
| Şekil 7.7 Karar verilen ürün risk matrisi | 153 |
| Şekil 7.8 Ürün risk matrisi çeyreklerine yönelik test yaklaşımları..... | 155 |
| Şekil 7.9 Test senaryosu örneği | 161 |

TABLolar LİSTESİ

| | <u>Sayfa</u> |
|----------------------------------------------------------------------|---------------------|
| Tablo 5.1 Ürün risk maddelerinin izlenebilirlik matrisi örneđi | 96 |
| Tablo 5.2 Test uygulama durumu tablosu | 97 |
| Tablo 6.1 Proje yöneticisi puanlama tablosu | 125 |
| Tablo 6.2 Yönetimden seçilen paydaşın puanlama tablosu | 125 |
| Tablo 6.3 Yazılım mimarı puanlama tablosu | 126 |
| Tablo 6.4 Ortalama puanlar tablosu | 126 |
| Tablo 6.5 Final puanlama tablosu | 127 |
| Tablo 7.1 Puanlama yapılacak olan test maddeleri | 143 |

1. GİRİŞ

Yazılım geliştirme süreci sonunda ortaya çıkan yazılım ürününün kaliteli olması önemlidir. Bir yazılım ürününün kalitesini değerlendirmek ise oldukça zordur. Bu nedenle yazılım geliştirme sürecinde kalite ve süreç yönetimi model ve standartlarına duyulan ihtiyaç günden güne artmaktadır. Yazılım projelerinde ortaya çıkan bütçe ve zaman aşımaları ,başarısızlıklar ve ürünün hatalı ya da talep edilenden farklı ortaya çıkması organizasyonları etkin bir yazılım kalite yönetim süreci yönetme ve bu sistemin sürekli olmasını sağlama yoluna itmiştir. Bu tez çalışmasında yazılım kalitesinin özellikle test yoluyla iyileştirilmesi konusu ele alınmıştır.

1.1 Literatür İncelemesi

Yazılım geliştirme alanında kullanılan en önemli süreç iyileştirme ve kalite yönetim modellerinden birisi Bütünleşik Yetenek Olgunluk Modeli (CMMI)'dir. CMMI referans alınarak yapılan süreç iyileştirme çalışmalarındaki asıl amaç, etkin bir süreç yönetimi altyapısı oluşturmak ve projelerde bu süreçleri doğru bir şekilde kullanabilmektir [Chrissis,2011].Yazılım geliştirme sürecinde kalitenin ölçütleri en kolay olarak test sürecinde ortaya çıkan sonuçlara göre nicel olarak ölçülebilir.Bu aşamada test sürecinin iyileştirilmesi yazılım ürününün kalitesini doğrudan arttıracaktır[Naik,2008]. TPI(Test Process Improvement) modeli test sürecinin iyileştirilmesi için ortaya çıkarılan bir modeldir ve sürecin daha anlaşılır ve verimli olmasını sağlar[Sogeti, 2008]. Son on yılda test sürecinin olgunlaşması için en çok referans alınan ve kullanılan model olmuştur. TPI modeli için tanımlanmış bir yol haritası vardır ve bu yol haritası üzerinden ilerleme kaydedilir. TPI modeli, bu modeli benimseyen organizasyonun yapısına uygun hale getirilebilir. TMAP(Business Driven Test Management) modeli müşterinin istekleri doğrultusunda şekillenen bir modeldir[Koomen, 2006]. Kapsamlı bir test süreci, test yönetimi ve deneysel ipuçları içerir. Birçok geliştirme modeline adapte edilebilir. TMMi(Test Maturity Model Integration) ise test sürecinin iyileştirilmesi için detaylandırılmış bir modeldir ve CMMI yaklaşımını tamamlayıcı niteliklere sahip olacak şekilde oluşturulmuştur[Veenendaal,2012].TMMi modeli test sürecinin iyileştirilmesine katkıda bulunurken, test sürecinin organizasyonlar için

yönetilmeyen ve dikkate alınmayan bir süreç yerine yönetilen, tanımlanan ve ölçülen bir süreç haline getirilmesini sağlayan aşamalar ve seviyeler içerir.

Yazılım sürecinde doğru risk tanımlamalarının yapılması ve test sürecinin risk maddeleri üzerinden yönetilmesi yazılım ürünün kalitesini arttıran farklı bir etkidir. PRISMA(Practical Risk-Based Testing) modeli test sürecinde olma olasılığı yüksek olan ve yazılıma etkisi yüksek olacak risk maddelerini ortaya çıkararak test sürecinin bu maddeler üzerinden ilerlemesini sağlar[Veenendaal,2012]. TMMi modelinin içerisinde bulunan test planlama aşamasında PRISMA modelinin uygulanması zaman ve bütçe açısından test sürecinin en verimli şekilde sonuçlanmasını sağlayacaktır.

1.2 Tezin Amacı

Son yirmi yıldır hemen hemen herkes günlük hayatında bilgisayar kullanmaktadır ancak çok az insan bilgisayar yazılımının kullanımı ve işlevselliği etkilediğini bilir. Akıllı telefonlar, uzay araçları, bilgisayar oyunları, web siteleri hatta kol saatleri bile birer yazılım sayesinde kullanılabilir hale gelmiştir. Bilgisayarlar için oluşturulmuş tüm yazılım ve uygulamalar, belirli sonuçların ortaya çıkması için tasarlanmış talimatlar kümesinden oluşur. Yazılım, kullanıcıların donanımsal sistemler ile iletişim kurmasının en kolay yoludur. Teknoloji gelişmeye devam ettikçe yazılımın önemi de giderek artacaktır.

Yazılımın kullanışlı olması, sorun çıkarmaması, sürdürülebilir ve yenilenebilir olması kullanıcılar ve yazılımı ortaya çıkaranlar için önemlidir. Yazılımın en az hata ile, zamanında ve kullanışlı bir şekilde ortaya çıkması için yazılımın test sürecinden mutlaka geçmesi gerekir. Bilgisayar yazılımlarına olan beklentinin artmasıyla yazılımda test faktörü de zamanla daha da önemli hale gelmiştir ve yazılımın kalitesini belirleyen bir ölçüt olarak ele alınmaya başlanmıştır. Yazılım testi, yazılım yaşam döngüsünde uygulanması gereken herhangi bir adımdan,tamamen ayrı ele alınması gereken bir süreç haline gelmiştir. Organizasyonlarda zaman ve bütçe, yazılım test sürecini kapsayacak şekilde oluşturulmaya başlanmıştır.Yazılım testi konusunda bir çok çalışma yapılmış, yeni yaklaşımlar ve uygulamalar oluşturulmuştur. Yazılım test süreci giderek gelişmeye devam etmektedir. Gereksinimlere dayalı test yaklaşımı, çevik yazılım geliştirme sürecine adapte olmuş test yaklaşımı, kullanıcı senaryolarına dayalı test yaklaşımı

gibi bir çok test yaklaşımı üzerinde çalışılmıştır; ancak en çok kullanılan test yaklaşımlarından biri TPI (Test İyileştirme Süreci) modelidir ve daha tutarlı, kesin sonuç veren, denetlemeye ve gözden geçirmeye dayalı bir test yaklaşımı olarak ortaya çıkmıştır. TMMi modelinin ise TPI modeli ile birçok benzer yönü vardır. Ancak TPI modeli kalite modellerine kolay entegre olabilecek şekilde geliştirilmemiştir. TMMi modeli ise CMMI modeline entegre şekilde uygulanabilir. PRISMA ise risk odaklı bir test yaklaşımıdır; Test sürecinin hızlanmasına yardımcı olur ve test sürecinin güvenilirliğini artırır. Bu tezin amacı, TMMi ve PRISMA yaklaşımlarının birlikte bir örnek problem üzerinde uygulanabilirliğini göstermektir.

1.3 Tez Çalışmasının Yapısı

Bu tez çalışmasının I. bölümü olan Giriş bölümünde tezin amacı anlatılmış ardından II. Bölümde yazılım kalitesi ve testi genel olarak ele alınarak ayrı ayrı uygulama alanları ve önemlerine değinilmiştir. III. Bölümde uluslararası yazılım test ve yeterlilik kurulu tarafından hazırlanmış olan kılavuz olarak sunulmuştur. Bu kılavuz yazılım test dünyasında kabul edilen en geçerli kılavuz olması sebebiyle tez çalışmasında yer almıştır. IV. Bölümde TMMi modeli ayrıntılı bir şekilde anlatılmış ve organizasyonların alabileceği TMMi seviyeleri ve o seviyelerde uygulamaları gereken süreç alanları ayrıntılı bir şekilde tanımlanmıştır. V. Bölümde risk odaklı test yaklaşımı özet olarak anlatılmış, yararları ve önemine değinilmiş sonrasında VI. Bölümde ise risk odaklı test yaklaşımının daha kullanılabilir ve anlaşılır olmasını sağlayan PRISMA yaklaşımı ayrıntılı olarak anlatılmıştır. VII. bölümde seçilen bir proje üzerinde TMMi modeli içerisinde PRISMA yaklaşımı uygulanarak bir sonuca ulaşılmıştır ve öneriler verilmiştir. Tezin sonunda konu ile ilgili kaynaklar ve referanslar sunulmuştur.

2. YAZILIMIN KALİTESİ VE TESTİNE GENEL BAKIŞ

2.1 Giriş

Bir yazılımın kalitesi ile o yazılımın testi arasında çok yakın bir ilişki bulunmaktadır. Bu bölümde yazılım kalitesi ve testi konusunda bir genel bakışa yer verilecektir. Ardından yazılımda kalite ve test ilişkisi anlatılacaktır.

2.2 Kaliteye Genel Bakış

Kalite elle tutulamayan, tartışılabilir, yargılanabilir, tam olarak tartılamayan ya da ölçülemeyen bir olgudur. Kaliteye farklı bakış açıları vardır. Bazıları kaliteyi kontrol edilemez ve ölçülemez olarak yorumlarken bazıları lüks, masraflı ve ayrıntılı olarak tanımlamıştır [Kan,1995].

Kalite yönetimi insan ve risk yönetimi ile birlikte yazılım süreç yönetiminin genel bir parçasıdır. Kalite yönetiminin bazı yönleri diğer yönetim aktiviteleriyle birlikte ele alınır. Tek istisna proje yönetimidir. Proje yönetimi, kalite yönetimiyle paralel yürütölmek istenir ancak kalite yönetiminin kendi bütçesi ve takvimi vardır. Ayrıca kalite yönetiminin en önemli görevlerinden biri proje yönetimindeki kaliteyi sağlamaktır. Kalite yönetimi faaliyet ve sonuçları proje yönetimindeki bütçe ve takvim değişikliklerini kapsamalıdır [Maciaszek,2007].

Günümüzde en yüksek rekabet gücüne sahip kuruluşlarda kalite yönetiminin temelinin sürekli iyileştirmeye dayalı olduğunu görüyoruz. Sürekli iyileştirme, sürekli arayışı ve dinamizmi ifade eder. Hedef belli bir standardı tutturmak değil, seviyeyi, hedeflenen seviye ne olursa olsun sürekli iyileştirmektir. Bu yaklaşımla mükemmellik arayışına ve sıfır hata sonucuna ulaşılır [Köse,2012].

Kalite kavramı çok yeni bir kavram olarak nitelendirilse de çok eski zamanlardan beri önem verilen bir değerdir. Kalite kavramı çok sık kullanıldığı için insanlar tarafından bilinen bir kavramdır. Ancak bu kavram günümüzde teknik bir yöntem değil, işletmelerin müşterilerini memnun etmeye ve işletmelerde “sosyal sorumluluk” bilincinin gelişmesine katkı sağlayan bir kavram olarak da yönetimle birlikte anılır olmuştur. Bunun sonucu olarak Toplam Kalite Yönetimi kavramı karşımıza çıkmıştır [Köse,2012].

Toplam Kalite Yönetiminin ana felsefesi herhangi bir kurumda çalışan herkesin katılımı ile sürecin sürekli olarak iyileştirilmesi ve geliştirilmesi esasına dayanır. Sürekli iyileştirme kavramı Masaaki Imai tarafından Toplam Kalite Yönetimi kavramına kazandırılmıştır. Bu kavram bir felsefeyi, bir yaşam tarzını ifade etmektedir. Japonlara göre sürekli iyileştirme öyle bir düşüncedir ki, her Japon her geçen günün bir öncekinden daha iyi olması için, evinde, işinde, sosyal yaşamında sürekli bir gayret içinde olmalıdır. Bu gelişmenin boyutu hiç önemli değildir. Önemli olan küçük ama emin adımlarla daha ileriye gitmektir [Köse,2012].

Tasarımın kalitesi tasarımcının ürüne ne kadar özendiğiyle orantılıdır. Kullanılan materyallerin cinsi, tolerans değerleri ve sözleşmeye uyumluluk performansı, tasarımın kalitesi için katkı sağlar. Sonuç olarak kalite önemlidir; ancak kullanıcı önemsemezse hiçbir şey önemli değildir. Örneğin; eğer bir yazılım ürünü kullanıcılarına önemli avantajlar sağlıyorsa, kullanıcılar güvenilirliği ve performans problemlerini göz ardı edebileceklerdir [Pressman,2001].

2.3 Yazılımda Kalite

2.3.1 Genel

Bir yazılımın kalitesi, kullanım amaçlarına göre açıkça tanımlanmış fonksiyonel gereksinimlerine uyum, kullanıcı isterlerine yanıt verebilme, açıkça belgelendirilmiş yazılım geliştirme standartlarına sadık kalma, yüksek güvenilirlik sağlama, üretilen yazılımda çeşitli özelliklere sahip olma ve teslim sonrası yeterli ve tatmin edici destek sağlama olarak tanımlanabilir [Sarıdoğan,2008].

Son onbeş yıldır yazılım kalitesi kavramı önem kazanmıştır. Bunun nedenlerinden birisi şirketlerin nesneye dayalı programlama gibi yeni kodlama teknikleri benimsemiş olmalarıdır. Buna ek olarak yazılım kalitesinin önemine karşı ve yazılım sektörünün gelişmesinde kalite tekniklerinin benimsenmesi konularında büyük bir farkındalık oluşmuştur. Ancak, yazılım kalitesi, üretim kalitesiyle paylaşılacak kadar karmaşık bir kavramdır. Üretimdeki kalite kavramı sadece üretilen ürünün beklentileri karşılaması yönündedir. Bu tanım tüm ürünlere uygulanabilir ancak yazılım sistemleri için bazı problemler içerir. Yapılan sözleşmede ürün karakteristikleri müşterinin isteği doğrultusunda yapılandırılabilir ancak geliştirme aşamasında anlaşmada olmayan bakım gereksinimleri gibi

gereksinimler eklenmesi gerekebilir. Kalite ölçütlerinin nasıl ele alınması gerektiğini belirten belirli bir yöntem de yoktur. Tamamen doğru gereksinimlere sahip bir yazılım dokümanı hazırlamak çok zordur. Bununla birlikte yazılım ürünleri beklenen tüm gereksinimleri karşılamalıdır çünkü kullanıcılar kendi beklentilerini karşılamadığı sürece yüksek kaliteli bir ürün olduğunu düşünmeyeceklerdir.

Gereksinimler ya da kullanıcı isterleri kalitenin ölçülmesine temel oluştururlar. Onlardan uzaklaşmak aslında fonksiyonel kaliteden uzaklaşmak demektir. Belirlenmiş standartlara uyularak geliştirilen bir yazılım belirli bir teknik kalite seviyesindedir; ancak fonksiyonel olarak işini göremeyen yazılım kaliteli olarak adlandırılmaz. Geliştirme standartlarının dışına çıkılması durumunda ise teknik kaliteden ödün verilmiş olur. Yazılımın sahip olması gereken doğruluk, sağlamlık, modülerlik, anlaşılabilirlik, bakım kolaylığı gibi özelliklerin eksikliği, fonksiyonel olarak çok iyi olan bir yazılımın kalitesinin eksikliği anlamına gelir [Saridoğan,2008].

Basit anlamda düşünürsek yazılımda kalite hataların olmayışı olarak özetlenebilir. Aynı zamanda önemli olan gereksinimlere uyumluluktur çünkü eğer yazılım çok fazla fonksiyonel hata içerirse gereksinimler de istenen işlevlerle uyuşmaz. Bu tanım iki kavramı içerir; hata oranı ve güvenilirlik. Hata oranı, kaynak koddaki milyon satır başına düşen hata miktarı olarak düşünülebilir. Güvenilirlik ise belirlenen bir (n) zamandaki başarısızlık sayısıdır denilebilir [Kan,1995].

Kalite birçok farklı formda incelenebilir. Tasarımın kalitesi, gereksinimlere uyumluluk kalitesi ve performans kalitesi, kalite tiplerinden bazılarıdır. Tasarımın, müşteri açısından kusursuz olması tasarımın kaliteli olduğunu gösterir. Gereksinimlere uyumluluk kalitesi, ürünün ortaya çıkışında müşterinin belirlediği gereksinim kriterlerini bire bir sağlamasıyla ölçülür. Performans kalitesi ise tasarlanan fonksiyonların tüketici tarafından kullanılabilirliği ile ölçülür. Kaliteyi etkileyebilecek birkaç anahtar terim vardır. Aşağıda bu terimler kısaca özetlenmiştir [Summers,2000].

- **Müşterinin Beklentisi:** Sadece müşteri, bir yazılımın kendi istekleri ve beklentileri doğrultusunda olduğuna karar verebilir.

- **Güncel Fikir:** Müşteri yazılımın kalitesini sadece teslimat sırasında değil kullanım sırasında da yargılayacaktır.
- **Gereksinimler:** Yazılımın yapısı müşteri tarafından bilinçli olarak anlatılamayabilir ancak müşterinin isteklerini hissederek ya da anlamaya çalışarak kaliteli yazılım çıkarılması beklenir.
- **Teknik Özellikler:** Yazılımın teknik yapısı kelime kelime müşteri tarafından tanımlanmış olmalıdır.
- **Kişisel Yaklaşım:** Yazılım ortaya çıkaran kişinin kişisel görüşlerine göre şekillenebilir.

Yazılım geliştiriciler yüksek kaliteli yazılımın önemli bir amaç olduğu konusunda hemfikir olurlar ancak bu yazılımda kalitenin nasıl tanımlandığına bağlı olarak şekillenir. Üretenler ve kullanıcılar için ölçülebilir değer kriterlerine bakılarak kullanışlı yazılımın ortaya çıkması en etkili yazılım kalite tanımıdır.

Kullanışlı yazılım kullanıcı isteklerine göre içerik, fonksiyon ve özelliklerinin tamamlanmasıyla ortaya çıkar ama önemli olan bunu, güvenilir hatasız bir yolla ortaya çıkarmaktır. Kullanışlı yazılım kullanıcılar tarafından belirlenmiş gereksinimleri karşılayabilmek zorundadır. Ayrıca kaliteli yazılımda olması gereken fonksiyonel olmayan gereksinimler de karşılanmalıdır [Pressman,2001].

Yazılım sözleşmelerinde ve kalite tasarım prosedürlerinde kusursuz bir anlaşma sağlamak önemli bir problemdir. Ancak bakım, güvenlik ve verimlilik gibi yazılım nitelikleri kolayca anlaşmaya dökülemez.

İnsanlar kaliteye yazılım geliştirenlerin düzenli olarak takip ettikleri kalite standartlarını tanımlayarak ulaşılabileceğini düşünürler. Buradaki iddia standartların iyi uygulamalar kapsadığı yönündedir ancak kalite yönetimi için standartlardan çok daha fazlası gerekir.

Formüle edilmiş kalite yönetimi karmaşık ve büyük sistemler geliştiren takımlar için önemlidir. Kalite belgeleri, her alt grubun projede neler yaptığının kayıdır. Bu belgeler diğer çalışanlara önemli görevlerin yapılıp yapılmadığının kontrolünün sağlanmasına yardımcı olur. Büyük sistemler için yazılım kalite yönetimi üç temel aktiviteye dayanmaktadır [Sommerville,2007].

- **Kalite Güvencesi** : Yüksek kaliteli yazılıma yönlendiren prosedür ve standart yapıların oluşturulmasıdır.
- **Kalite Planlama** : Oluşturulan yapıdan uygun prosedür ve standartların seçilmesi ve yazılım projesine dahil edilmesidir.
- **Kalite Kontrol** :Yazılım geliştirme takımının, kalite prosedür ve standartlarının takibini sağlaması için oluşturulan tanımlama sürecidir.

Yazılım geliştirme dünyasında kalite ilkeler topluluğudur ve bu ilkeler iyi ve örnek uygulamalar ile meydana gelmiştir. İyi bir uygulamada hataların erken teşhis edilmesi ve bu teşhislerin yerinde müdahaleler ile erken çözümlenmeleri, yazılımın maliyetini düşürmektedir. Yazılım geliştirme yaşam döngüsünde ürün değil süreçler önemlidir ve bu süreçlerin sürekli iyileştirilmesi hedeflenmelidir. Standartlar belirlenmeli, sürekli ölçümler yapılarak sonuçlar değerlendirilmelidir.

Yazılım geliştirme süreci, amacı bir iş yapış biçiminde standart oluşturmak, değişkenliği azaltmak ve bu şekilde iş yapış biçiminde iyileşme sağlamak olan bir iş yapma yöntemidir. Genellikle bazı alt süreç ve aktivitelerden oluşur. Süreçler mutlaka yazılı hale getirilip belgelenmiş ve tekrarlı işlemlerdir. Her sürecin mutlaka girdileri ve çıktıları vardır.

2.3.2 CMMI

Yazılım geliştirme şirketleri periyodik olarak yeteneklerini geliştirir ve seviyelerini belirler. 1980'lerde kâr amacı gütmeyen Software Engineering Institute (SEI) Amerika Savunma Bakanlığı adına CMM diye bilinen bir sınıflandırma yayınladı.

Savunma bakanlığının amacı yeterliliklerine göre yüklenicilerin yazılım geliştirme yeteneklerini sınavabilmektir. Enstitünün bu sistemi şu anda CMMI olarak bilinen ve kuruluşların yazılım mühendisliği yetkinlik hedeflerinin sağlamaştırılmasında başarı kaydeden bir sistem olarak bilinmektedir. Aslında CMMI in alanı yazılım mühendisliğinden daha geniştir. Fakat yazılım kalitesinde önemli olan CMMI-SW alt kümesidir. Birçok kuruluş geliştirme süreçlerinin kalitesini belirlemek için CMMI-SW kullanmaktadır. Yazılım geliştirme kuruluşları CMMI seviyesi olarak 1(en düşük) ile 5(en yüksek) arasında değerlendirilirler. CMMI modeli organizasyonları sürekli ve aşamalı olmak üzere iki çeşit değerlendirme ile ayırt eder. Sürekli

değerlendirmede organizasyonlara herhangi bir süreç alanını ya da alanlarını seçerek, yalnızca orada iyileştirme yapabilme imkanı sunar. Aşamalı değerlendirme ise belirli adımlar içerir.

CMMI Seviye 1 :Başlangıç (*Initial*)

Yazılım geliştirme süreci çoğu kez geliş güzel yapılmaktadır ve genel olarak kaotik bir durum söz konusudur. Bu seviyede başarı kişisel çabalar ile oluşmaktadır. Herhangi bir kriz durumunda planlar göz ardı edilir. Yazılım konusundaki yetenekler ise önceden öngörülemez.

CMMI Seviye 2 : Yönetilen (*Managed*)

Genellikle proje yönetimi konusundaki kontrol noktaları belirlenmiştir ve projeler önceden belgelenmiş olan proje planlarına göre yürütülmekte ve yönetilmektedir. Yazılım gereksinimleri bir şekilde yönetilir ve bu gereksinimler ile ilişkili ürünler oluşturulur. Bu ürünler ile ilgili denetim de mevcuttur. Proje yönetim sistemi vardır ve bu sistem proje geliştirme sürecini etkin bir şekilde kontrol eder. Projeler planlıdır, başarılıdır, ölçülmüş ve kontrol edilmiştir. Yazılım gereksinimleri, ilgili süreçler, süreçlerin çıktıları ve hizmetler yönetilmektedir. Durum tanımlandığı kadarı ile yönetimin görüşüne açıktır. Anlaşmalar, söz konusu projedeki tüm paydaşların talep ve değişiklik isteklerine uygun bir şekilde verilmiştir.

CMMI Seviye 3: Tanımlanmış (*Defined*)

Yazılım geliştirme süreçleri iyi tanımlanmış ve anlaşılmalıdır. Tüm süreçler standartlar, prosedürler, araçlar ve metodolojiler ile açıklanmış ve anlatılmıştır. Standart süreç tanımları organizasyon içinde tutarlıdır. Projeler, tanımlı standart süreçleri uyarlama rehberlerine uygun olarak kendilerine uyarlayarak kendi süreçlerini oluşturur. Yani, tüm projelerde standart yazılım geliştirme süreçlerinin uyarlanmış hali kullanılır. Bu seviyede, süreçler ikinci seviyeye göre daha özenli ve detaylı olarak tanımlanmıştır.

CMMI Seviye 4 : Nicel Olarak Yönetilen (*Quantitatively Managed*)

Bu seviyede sürecin başarılı olup olmadığını belirlemek için önem arz eden alt süreçler belirlenir ve bu alt süreçler istatistiksel ve diğer nicel ölçüm teknikleri ile

kontrol edilir. Kalite ve süreç performansının nicel hedefleri belirlenir ve bunlar süreçlerin yönetiminde bir ölçüt olarak kullanılır. Nicel hedefler, tedarik makamlarının ihtiyaç ve istekleri, son kullanıcılar, organizasyon ve bu süreçleri geliştirenler temel alınarak belirlenir. Süreçler ölçülür ve ölçülebilen sınırlar içerisinde gerçekleştirilecek aktiviteler gerçek veriler baz alınarak tahmin edilebilir. Üçüncü seviye ile temel bir farklılık vardır. Bu da, süreç performansının öngörülebilir olmasıdır. Bu seviyede, süreç performansları istatistiksel ve diğer nicel teknikler ile kontrol edilebilir ve nicel olarak öngörülebilir iken üçüncü seviyede süreçler sadece nitel olarak öngörülebilir.

CMMI Seviye 5 : Optimizasyon (*Optimizing*)

Bu seviyede, süreçten gelen sayısal geri beslemelerden ve teknolojilerden yararlanılarak sürekli iyileştirilme yapılır. Tüm organizasyon süreç iyileştirmeye odaklanmıştır. Yazılım geliştirme süreci “Sürekli İyileşen” olarak tanımlanabilir. Bu seviye ile dördüncü seviye arasındaki en önemli ayrım, süreçlerin değişme derecesidir. Dördüncü seviyede süreçler, süreç değişkenliğinin özel nedenleri ve sonuçların istatistiksel öngörüsüne odaklanırken, beşinci seviyede süreçler, süreç değişkenliğinin genel nedenleri ve sürecin değişimi ile ilgilidir [Braude and Bernstein,2011].

2.3.3 Yazılımda kalite ölçümü ve yazılım metrikleri

Geçerlilik ve güvenilirlik kalite ölçmesinin en önemli iki kriteridir. Geçerlilik, ne amaçlandığını metrikleri kullanılarak ölçmemizi sağlar. Güvenilirlik ise metriklerin ölçülmesinin ve ölçme metodlarının tutarlılığına karşılık gelir [Kan,1995].

Yazılım kalite ölçme süreci kalite kontrol sürecinin önemli bir parçasıdır. Sistemin her bir birimi analiz edilir ve metrik değerleri, önceki projelerden elde edilen verilerle karşılaştırılır. Ölçme, bileşenlerdeki kalite güvencesine odaklanmak için kullanılabilir [Sommerville,2007].

Kalite güvencesinin sağlanması, bitmiş projelerde kalite süreç ve standartlarının belirlenmesiyle ilgilidir. Kalite, proje yönetiminden bağımsız olması gerektiği için kalite güvencesinin bağımsız biri tarafından yönetilmesi gerekir. Bunun için yazılım kalite güvence (*SQA, Software Quality Assurance*) ekibi oluşturulmalıdır. Bu ekip organizasyon içerisinde görev alan başarılı insanlardan oluşmalıdır ancak bu

elemanlar projede görev almamalıdır. Yazılım geliştiriciler yerine bu takım projenin kalitesinden sorumlu tutulmalıdır. Oluşturulan takım geliştirilen projenin amacı ve önemi ile birlikte uygun seviyelerde fonksiyonel yönetimleri raporlamalıdır. Bu yüzden stratejik, taktik ve işletme yönetimi seviyesi olarak adlandırılabilir [Maciaszek,2007].

Öncelikle ölçütlerin seçilmesi gerekmektedir. Ölçütler sorduğumuz sorulara yanıt verecek nitelikte olmalıdır. Bu sorulara net olarak yanıt veremeyen ölçütler ayrılmalıdır. İkinci aşama ise değerlendirmeye alınacak bileşenler seçilmelidir. Yazılım sistemindeki tüm bileşenler için metrik değerlerini belirlemek mümkün olmayabilir. Bazı durumlarda, ölçme için bileşenlerden temsili bir seçim yapılabilir. Kritik olan bileşenlerde ise metrikler belirlenmelidir. Üçüncü aşamada bileşen karakteristikleri ölçülür. Elde edilen metrik değerleri toplanır. Genelde bunun için veri toplama programları kullanılır. Diğer aşamada ise sıra dışı olarak görülen ölçütler belirlenir. Bileşen ölçütleri veri tabanında karşılaştırıldığında alışılmadık şekilde her metrik için çok yüksek ya da çok alçak değerler olabilir. Bu, o bileşende bileşen değerleri ile ilgili bir problem olduğunu gösterir. Son aşamada ise sıra dışı değerler analiz edilir. Bir metrik için sıra dışı değeri olan bileşende, o bileşenin kalitesi düşer. Yalnızca karmaşıklık için metrik değeri anormalse, ürün düşük kaliteli anlamına gelmez [Summers,2000].

Ürün geliştirme sırasında yazılım ürünlerinin yanı sıra yazılım süreçleri de kalite yönetimi hedefleri arasındadır. Farklı yazılım projeleri için değişen yazılım kalite ölçütleri vardır. Bu kalite ölçütleri sistemin yazılım ürünü tarafından karşılanan fonksiyonel gereksinimlerine en önemli katkıyı sağlar. Bu yazılım kalite ölçülerini sıralarsak; doğruluk, güvenilirlik, sağlamlık, performans, kullanılabilirlik, anlaşılabilirlik, sürdürülebilirlik, ölçülebilirlik, yeniden kullanılabilirlik, taşınabilirlik, ortak çalışabilme özelliği, verimlilik, güncellik ve görünürlüktür. Sayılan kalite ölçülerinden üç tanesi çok önemlidir. Bunlar anlaşılabilirlik, sürdürülebilirlik ve ölçülebilirliktir. Bu üçü uyarlanabilirlik olarak da bilinir [Maciaszek,2007].

Yazılım metrikleri ürün metrikleri, süreç metrikleri ve proje metrikleri olarak üç kategoride incelenebilir. Ürün metrikleri ürünün karakteristikleri olarak belirlenmiştir. Ürünün boyutu, karmaşıklığı, tasarımı, performansı ve kalite seviyesi. Süreç metrikleri yazılım geliştirme ve iyileştirme sürecinde olan

metriklerdir. Geliştirme sırasında hataların çözülmesi, bunun için geçen süre, süreç metrikleri olarak adlandırılır. Proje metrikleri ise projenin özellikleri ve uygulanmasını tanımlayan metriklerdir. Çalışan sayısı, yazılımın döngüsü, takvim, maliyet gibi.

Yazılım kalite metrikleri, yazılım metriklerinin alt kümesi olarak belirlenebilir. Yazılım kalite metrikleri çıkan ürünün kalite metrikleri ve sürecin kalite metrikleri olarak ikiye ayrılabilir. Yani kaliteye hem yazılım süreci içerisinde hem de ürünün kalitesi açısından bakmalıyız [Summers,2000].

Yüksek kaliteli yazılım geliştirme, başarılı yazılım sistemleri için kritik bir faktördür. Yüksek kalite seviyesine gelmek kolay bir iş değildir. Kalite projenin başında hedeflenmeli ve geliştirmenin her aşamasıyla bütünleşmelidir. Yazılım projesinin yaşam döngüsü boyunca veriler ve ölçütler sürecin geçerliliğini anlamak için toplanır ve yazılım geliştirilir. Bu veriler ve ölçütler metrik olarak adlandırılır. Bu ölçütlerin, yazılımın kendi kalite seviyesini arttırmak için belirlenmiş, projenin takvime uyma derecesi ve mühendislerin verimliliği gibi farklı yönleri vardır.

Metrikler verilen özelliklere göre yazılımın ya da sürecin sahip olduğu dereceyi ölçen sayısal ölçülerdir. Her bin satır koddaki hata sayısı ve ortalama yazılım modül büyüklüğü örnek olarak verilebilir. Metrikler yazılımın yaşam döngüsü boyunca toplanır, analiz edilir ve aşağıdakiler için yardımcı olur [Braude and Bernstein,2011]:

- Yazılım test seviyesinin belirlenmesi,
- Proje takviminin değerlendirilmesi,
- Sürecin ilerleyişini takip etmek,
- Yazılım büyüklük ve karmaşıklığını belirlenmesi,
- Projenin maliyetinin belirlenmesi,
- Süreç iyileştirmesi.

Kalite, yazılımın gereksinimleri yerine getirme oranı olarak görüldüğünden beri, kalite uyumluluk derecesi ölçümü zorunluluk haline gelmiştir. Yazılım kalite metriklerini proje için toplanan metriklerin bir alt kümesi olarak tanımlamak kullanışlı olacaktır. Bu metrikler yazılım yaşam döngüsü içerisinde yazılım kalite ve

süreç karakteristikleri üzerine yoğunlaşmaktadır. Bu metrikler aşağıdaki gibi sıralanabilir [Braude and Bernstein,2011]:

- Yazılımın boyutuna oranlı çıkan hata sayısı,
- İki hata arasında geçen süre,
- Müşteri problemleri,
- Müşteri memnuniyeti.

Kalite etmenlerinin ölçülmesi çoğu zaman oldukça zordur, hatta bazen de olanaksızdır. O nedenle bazı metrikler, diğer adıyla ölçütler, tanımlanır, her metrik için 1 ile 10 arasında bir not verilir ve bu notlar yardımıyla sayısal bazı değerler elde edilir. Değerlendirmede dikkate alınacak metrikler aşağıdaki gibi sıralanabilir[Sarıdoğan,2008]:

- **Denetlenebilirlik:** Yazılımın standartlara uyum derecesinin denetim kolaylığıdır.
- **Doğruluk:** Yazılımın arzu edilen işlevleri eksiksiz ve doğru olarak yerine getirebilmesidir.
- **Hassaslık :** Hesaplamaların ve kontrol işlemlerinin sayısal doğruluk derecesidir.
- **Başarı :** Yazılımın arzu edilen işlevleri istenen hızda yerine getirilebilmesidir.
- **Arayüzün Yaygınlığı :** Arayüzlerin, veri aktarım protokolünün, aktarım hızlarının standartlara uyumluluğudur.
- **Verilerin Yaygınlığı :** Yazılımın tamamı içinde standart veri yapılarının ve veri tiplerinin kullanılmasıdır.
- **Bütünlük :** İstenen tüm işlevlerin gerçekleştirilme derecesidir.
- **Büyüklik :** Yazılımı oluşturan kaynak kod satır sayısı, modül sayısı, öz kaynak gereksinimi gibi değerlerdir.
- **Tutarlılık :** Yazılımın geliştirme projesi boyunca aynı tasarım ve belgeleme tekniklerinin kullanılmasıdır.
- **Hata Dayanıklılığı :** Yazılımın bir hatayla karşılaşması durumunda oluşan hasarın büyüklüğüdür.
- **Verimlilik :** Çalışma sırasında başarımlı derecesidir.

- **Geniřleyebilirlik** : Mimarının, veri yapılarının ve yordamsal tasarımın geniřleyebilme derecesidir.
- **Donanım Baęımlılıęı** : Yazılımın üzerinde alıřtıęı bilgisayar donanımına olan baęımlılık derecesidir.
- **İzlenebilirlik** : Program alıřırken kendi kendini izleyebilmesi, hatalarını gosterebilmesidir.
- **Modulerlik** : Program bileřenlerinin iřlevsel olarak birbirinden ayrıık olmasının derecesidir.
- **Kullanım Kolaylıęı** : Programın renilmesinin ve iřletiminin kolaylık derecesidir. Kullanıcı arayznn nitelięi en byk etmendir.
- **Bakım Kolaylıęı** : Yazılıma sonradan uygulanabilecek iyileřtirici ve dzeltici bakımın ne derece kolay ve kısa srede yapılabileceęidir.
- **Belgelendirme** : Yazılımın tamamının tasarımının ve kodlarının anlaşılır řekilde belgelendirilmesinin derecesidir.
- **Mřteri Tatmini** : Tm yazılımların belirli bir kullanıcısı ya da mřterisi vardır. Ama kullanıcıların tatmin olmasıdır.

2.4 Yazılımda Test

Bu blmde yazılımda test olgusu genel hatlarıyla anlatılacaktır.

2.4.1 Genel

Programcılar kodlarını gereksinimleri karřılayacak řekilde yazdıklarını dřnebilirler ancak yazılım sistemleri ok sayıda farklı durum, formller ve karmařık algoritmalar ierir. Bununla birlikte projenin byklę ve alıřanların okluęu karmařıklıęı arttırabilir. Bu yzden hataların varlıęı sadece yazılım iin deęil, kullanıcı ve mřterinin beklentileri iin de problem oluřturur [Kan, 1995].

Genel olarak dřnrsek eęer yazılım tarif edilen gereksinimleri karřılamıyorsa bu yazılım hatalıdır. Hata birok farklı sebebin sonucu olarak ortaya ıkmıř olabilir. Gereksinimler yanlış ya da eksik olabilir. Bazı gereksinimlerin donanımsal ve yazılımsal olarak uygulanması imknsız olabilir. Sistemin genel tasarımı hatalar ieriyor olabilir ya da en sık grldę řekilde yazılan kod hatalı olabilir. Sonu olarak bir ya da birden fazla hata bařarısızlıęın nedeni olabilir [atal,2012].

Test, hataların bulunması, yazılım kalitesi için güven kazanılması, karar mekanizmaları için bilgi elde edilmesi ve eksikliklerin önüne geçilmesi açısından önem taşır. Özetlersek test ürünün beklenen seviyede olduğunu belirlemek, değilse de istenilen ölçüye gelmesini sağlamak için kullanılan bir süreçtir. Test etmek programın kalite ve güvenilirliğinin artmasını sağlar.

Yazılım testi teknik bir iştir ancak bununla birlikte insan psikolojisini ve bazı önemli kriterleri de göz önünde bulundurmak gerekir. Normal şartlarda bir program içindeki tüm olası permütasyonları test etmek isteriz ancak çoğu durumda bu olmaz. Çok basit görünen bir programın bile yüzlerce hatta binlerce olası girdileri ve çıktıları olabilir. Tüm bu olasılıklar için test durumu oluşturmak imkânsızdır. Karmaşık bir uygulamanın tamamen test edilmesi çok uzun zaman alacaktır ayrıca fazla işgücüne ihtiyaç duyulacaktır.

Yazılım test eden kişilerin, yazılım uygulamalarını test etmek için uygun bakış açısına sahip olmaları gerekir. Bazı durumlarda testçinin bakış açısı işlemesi gereken süreçten daha önemli olacaktır.

Yazılım uygulamalarının testlerinin eksik olmasının en önemli nedeni yazılımcıların sürece yanlış tanımlamalarla başlamalarıdır. Bir programı test ettiğimiz zaman ona bir değer katmak isteriz. Yazılıma değer katmak programın güvenilirliğini ve kalitesini artırır. Programın güvenilirliğinin artması ise hataların bulunup düzeltilmiş olmasına bağlıdır. Bir programın çalışıp çalışmadığına bakmak için test etmeyiz bunun yerine programda hatalar olabileceğini varsayarak teste başlarız ve olabildiğince çok hata bulmaya çalışırız. Bu açıdan bakarsak test hataları bulma amacıyla bir programın çalıştırılma sürecidir. Test çoğu insanın zor bulduğu yıkıcı bir süreçtir ancak hepimiz hayatımızda yıkıcı değil yapıcı olmayı tercih ederiz. Bu tanım aslında test verilerinin nasıl tasarlanması gerektiğini içeren ve kimin test yapıp yapamayacağını da belirleyen bir tanımdır. Bu tanımlara ek olarak test etmeyi, başarılı ve başarısız kelimelerinin proje yöneticileri tarafından test durumlarının sonuçlarını kategorize etmek için belirli özelliklere göre kullanması olarak da anlatabiliriz. Proje yöneticileri hata bulunmayan test durumlarını başarılı olarak tanımlarken, hata bulunan test durumlarını başarısız olarak tanımlarlar. Aslında başarısız hoş olmayan bir tanımlama değildir. İyi inşa edilmiş ve koşturulmuş bir test durumu eğer hata bulunup yok edildiyse başarılıdır. Aynı test durumu

defalarca kořturulmuř ve daha fazla hata bulunmamıřsa da bařarılıdır. Gerçek anlamda yazılımı zorlamayan ve test durumlarında hata bulunmayan test bařarısız test olarak tanımlanabilir. Programlamanın doęasını dūřunursek yazılımın hiç hatasız olması inandırıcı deęildir [Kan,1995].

Yazılım testi, bir sistem veya uygulamanın denetlenebilir kořullar altında alıřtırılması ve elde edilen sonuların deęerlendirilmesidir. Denetlenebilir kořulların hem normal hem de anormal kořulları kapsamalı gerekir. Yani testin, bilinli Őekilde hatalı Őeyler yaparak olabilecek Őeyleri nceden belirlemeye ynelik olması gerekir. Olması gereken durumların olmadıęını veya tam tersi olmaması gereken durumların olduęunu denemek ve ortaya ıkartmak testin amacı olmalıdır.

Yazılım testi ařaęıda belirtilen amalar iin yapılır [Tian,2005]:

- Mūřteriye sunulmadan nce rn kalitesinden emin olmak,
- Yeniden alıřma (dzeltme) ve geliřtirme masraflarını azaltmak,
- Geliřtirme iřleminin erken ařamalarında yanlıřları saptayarak ileri ařamalara yayılmasını nlemek, bylece zaman ve maliyetten tasarruf saęlamak,
- Mūřteri memnuniyetini arttırmak ve izleyen sipariřler iin zemin hazırlamak.

2.4.2 Yazılımda test trleri

Test aktiviteleri yazılım sistemlerinin belli sebeplerle ya da sadece test amalı doęrulanmasını hedefler. Test tipleri hedeflenen test srecinin blm blm ilerlemesini saęlar. Bu, yazılımın iinde bulunan bir fonksiyonun testi, iřlevsellięi olmayan ancak kaliteye katkı saęlayan kullanılabilirlik ya da gvenilirlik testleri, yazılım ya da sistemin yapısal durum testleri ve deęiřiklikler sonrası yapılan testler olabilir.

Yazılım modelinde yapısal test iin fonksiyonel test ve fonksiyonel olmayan test yapılır. Fonksiyonel testte mūřteriyle imzalanan anlařma stndeki gereksinimlerin karřılandıęı grlmelidir. Sistemde ya da alt sistemlerde bulunan kullanım senaryoları, anlařmada yazılmıř ya da yazılmamıř olan fonksiyonel maddeler test edilir. Fonksiyonel testte yazılımın dıř ortamla uyumlu alıřıp alıřmadıęı da kontrol edilir yani yazılımın “ne” yaptıęına bakılır. Gvenlik testi,

hatalar bulunduktan sonra fonksiyonların gözden geçirilme testleri, sistemlerin kendilerine entegre olan diğer sistemlerle bütün olarak çalışmasının kontrolünün yapıldığı testler fonksiyonel testlerdir. Fonksiyonel olmayan testler ise performans testi, stres testi, güvenilirlik ve kullanılabilirlik testi, bakım testi, taşınabilirlik testi olarak sayılabilir. Fonksiyonel olmayan testlerde sistemin “nasıl” çalıştığına bakılır. Yapısal testte ise tüm test aşamaları yerine getirilir. Yazılımda kodla birebir çalışılarak yapılan test türüdür ve testin kodun ne kadarını kapsadığı kontrol edilir. Tüm bunların dışında değişiklikler sonucunda yapılması gereken testler vardır. Bir eksiklik ya da hata giderildikten sonra hatanın başarıyla yok edildiğini görmek için tekrar testi yapılmalıdır. Bu teste doğrulama testi adı verilir. Kodu çalıştırmakla karıştırılmaması gerekir. Debug etmek bir koda yapılan bir aktivitedir test aktivitesi değildir. Regresyon testi ise hatalar giderildikten sonra kodun farklı bir yerinde aynı hatadan olma durumuna karşı yapılan ya da kod düzeltildikten sonra kodun başka bir yerinin etkilenmesi durumuna karşı yapılan testtir [Kan,1995].

Geniş bir sistemde, test etmek birçok aşamayı gerektirir. İlk olarak her bileşen diğer bileşenlerden izole bir şekilde, kendi içinde test edilir. Bu tip teste modül testi, birim testi ya da ünite testi diyebiliriz. Tüm birimler test edildikten sonra birimlerin birlikte çalıştığına kontrolünün yapılması için entegrasyon testi yapılır [Kan,1995].

Proje ve süreç için hangi test tekniğinin kullanılabileceği çeşitli faktörlere göre belirlenebilir. Asıl faktör test durumunda ürünün nasıl davrandığıdır. Yazılım kodu, yazılım modelleri ve dokümanları için farklı test tekniklerine ihtiyaç olduğu açıktır. Başka bir faktör ise gereksinimlerin yazılım tarafından kapsanması ve testlerin ayrıntılı yapılmasıdır. Bununla birlikte sistem servisleri ve sistem sınırlamaları için farklı tekniklere ihtiyaç olacaktır. Her yazılımcı sistemi geliştirirken formal olmayan bir şekilde test yapar ancak formal olmayan test eksik ve kusurludur. Yazılımı geliştiren kişiler sistemde hata bulabilme ihtimali en düşük olan insanlardır [Maciaszek,2007].

Test türleri aşağıda kısaca özetlenmiştir [Tian,2005]:

a. Karakutu Testi (Blackbox Testing)

Bu tür testlerde yazılımın yapısı, tasarımı veya kodlama tekniği hakkında herhangi bir bilgi olması gerekli değildir. Yazılımın gereksinimlere yanıt verip veremediği ve fonksiyonelliği sınanmaktadır.

b. Beyazkutu Testi (Whitebox Testing)

Bu tür testler, uygulama kodunun iç mantığı üzerindeki bilgiye bağlıdır. Yazılım kodundaki deyimler, akış denetimleri, koşullar elemanlar sınanır. Kodun her parçasının test edilmesi beklenir.

c. Birim Testi (Unit Testing)

Mikro ölçekte yapılan bu testte, özel fonksiyonlar veya kod modülleri test edilir. Bu test, test uzmanlarınca değil programcılar tarafından yapılır ve program kodu ayrıntılarına ve içsel tasarım biçiminin bilinmesi gereklidir. Uygulama kodu çok iyi tasarlanmış bir mimaride değilse oldukça ayrıntılı bir testtir.

d. Artımsal Entegrasyon Testi (Incremental Integration Testing)

Uygulamanın yeni fonksiyonel elemanları eklendikçe sürekli test edilmesidir. Bu testte uygulamamanın tüm parçaları tamamlanmadan önce yeni eklenen parçanın fonksiyonelliği öncekilerden bağımsız şekilde çalışıp çalışmadığı sınanmaktadır.

e. Entegrasyon Testi (Integration Testing)

Bir uygulamanın farklı bileşenlerinin beraberce uyum içinde çalışıp çalışmadığını sınamak için yapılan bir testtir. Bileşenler, modüller, bağımsız uygulamalar, sunucu uygulamaları biçiminde olabilirler. Bu tür testlere, özellikle sunucu uygulamaları ve dağıtık sistemlerin testlerinde başvurulmaktadır.

f. Fonksiyonel Test (Functional Testing)

Bir uygulamanın fonksiyonel gereksinimleri üzerine odaklandırılan kara-kutu testidir. Genellikle kullanıcı ara yüzleri üzerinde yapılan test türüdür.

g. Yüzeysel Sistem Testi (Smoke Testing)

Uygulamanın tanımlanan gereksinimlerin tümünü karşılayıp karşılamadığını sınamak için yapılan bir kara-kutu testidir.

h. Regresyon Testi (Regression Testing)

Uygulama ve uygulama ortamlarında gerekli değişiklikler ve sabitlemeler yapıldıktan sonra yeniden yapılan testlere regresyon testi denilir. Böylece, önceki testlerde belirlenen sorunların giderildiğinden ve yeni hatalar oluşmadığından emin olunur. Uygulamanın kaç kez yeniden test edilmesi gerektiğini belirlemek güçtür ve bu nedenle, özellikle uygulama geliştirme döneminin sonlarına doğru yapılır.

i. Kabul Testi (Acceptance Testing)

Kullanıcı veya müşteri isteklerine dayanan son test işlemidir. Ayrıca, son kullanıcıların belli bir süre kullanımlarından elde edilen sonuçlar üzerinde de yapılabilmektedir.

j. Yük Testi (Load Testing)

Uygulamanın ağır işlem yoğunluğu altında test edilmesidir. Örneğin, bir Web sitesi için sistem tepkisinin hangi noktada azaldığı veya yanıt veremez olduğunu belirlemek için yapılan testler gibi.

k. Zorlanım Testi (Stress Testing)

Bu test, çoğu kez "yük testi" ve "performans testi" ile aynı anlamda kullanılmaktadır. Aynı zamanda, beklenmedik ağır yükler, belirli eylemler ve taleplerin çok fazla artışı, çok yoğun sayısal işlemler, çok karmaşık sorgulamalar vb. ağır koşullar altında olan bir sistemin işlevsellik testi olarak yapılmaktadır.

l. Performans Testi (Performance Testing)

Bu test türü 'zorlanım' ve 'yük' testi ile eş anlamlı olarak da kullanılabilir. Ancak, yapılması gereken performans testinin gereksinimlerde veya test planlarında açıklanmış olması gerekir.

m. Kullanılabilirlik Testi (Usability Testing)

Kişisel yargılara, hedeflenen son kullanıcı veya müşteri kitlesine bağlı olarak değişir. Kullanıcı yorumları, kullanıcı oturumlarından video kayıtları veya diğer teknikler kullanılabilir. Programcılar ve test uzmanları genellikle bu tür testler için uygun değildir, yani bu testlerin doğrudan son kullanıcılar üzerinde yapılması gerekir. Bu test genellikle geliştirme sürecinin erken aşamalarında yapılır, böylelikle uygulamanın kullanıcı ara yüzlerinde önemli değişiklikler yapılması mümkün olur.

n. Güvenlik Testi (Security Testing)

Yazılımın, gerek iç ve gerekse dış kaynaklı yetkisiz erişimlere, kötü amaçlı kullanımlara karşı korunması ya da güvenliğini test etmek için yapılır.

o. Uyumluluk Testi (Compatibility Testing)

Yazılımın özel bir donanım, yazılım, işletim sistemi, ağ veya ağ protokolü gibi ortamlarda beklenen şekilde çalışıp çalışmadığını sınamak için yapılan testlerdir.

p. Kurma/Kaldırma Testi (Install/uninstall Testing)

Bu test, yazılımın kurulması ve kaldırılması ile ilgili tüm seçenekler ve özelliklerin düzgün şekilde çalışıp çalışmadığını sınamak için yapılır.

q. Ağ Testi (Network Testing)

Çok kullanıcılu uygulamaların ağ ortamında gerçekten ağ üzerinde çalışabilme yeteneklerini ortaya koymak için yapılan bir testtir. İstenirse, farklı ağ işletim ortamları ve iletişim kuralları altında test yapılması tercih edilmelidir.

r. Alfa Testi (Alpha Testing)

Bitirme aşamasına yakınlaşmış olan bir uygulama için yapılan testtir. Bu test sonucunda ürün üzerinde küçük değişiklikler yapılabilir. Programcılar veya test uzmanlarınca değil, son kullanıcılar tarafından yapılır.

s. Beta Testi (Beta Testing)

Uygulamanın tamamlanması ve zorunlu testleri yapıldıktan sonra, son sürümü çıkarmadan önce hatalar veya sorunları saptamak üzere yapılan testlerdir. Programcılar veya test uzmanlarınca değil son kullanıcılar tarafından yapılır.

2.5 Yazılımda Kalite ve Test İlişkisi

Testin yardımıyla hataların bulunması yazılım kalitesinin ölçülmesine katkı sağlar. Test etmek eğer az hata bulduysa yazılım kalitesinin güvenilirliğini artırır ve sistemdeki risklerin görülmesine yardımcı olur. Test esnasında hata bulunduğunda bu hatalar düzeltilebilirse yazılım sisteminin kalitesi artar. Çalışılmış olan önceki projelerden ders çıkarılabilir. Hataların kaynağı ve sebep olduğu durumlar belirlenirse gelecekte yapılacak sistemlerin kalitesinde de artış sağlanacaktır. Kaliteli yazılımı için test aktiviteleri ve kalite aktiviteleri bütünleşik olarak yürütülmelidir [Kan,1995].

Kalite güvencesi yazılım ürününün ya da sürecinin inşası ile ilgiliyken, kalite kontrolü yazılım ürününün test aşamasıyla ilgilidir. Kalite güvencesinin stratejik boyutu varken, kalite kontrolü doğası gereği planlanmış ve işlevseldir.

Ağırlıklı yazılım test aktivitelerinde, kalite kontrol sistem geliştirmenin yaşam döngüsünü uzatır. Test etme kalite yönetim planının en önemli parçasıdır. Test planı, test süreci, test bütçesi, test durumları ve kaynaklarını kapsar. Daha önce geliştirilmiş test edilebilir yazılım ürününde, test etme aşaması kalite güvencesi yerine konuyordu ve yazılım modellerinde ve dokümanlarda kullanılıyordu [Maciaszek,2007].

3. ULUSLARARASI YAZILIM TEST YETERLİLİK KURULU'NUN YAZILIM TESTİ KILAVUZU ÖZETİ (ISTQB)

3.1 Genel

Bu bölümde Uluslararası Yazılım Test Yeterlilik Kurulu (International Software Testing Qualification Board)'nun yayınlamış olduğu belge özetlenecektir [ISTQB,2012].

3.2 Testin Temelleri

Bu bölümde test sürecinin temel bileşenleri özetlenecektir.

3.2.1 Yazılım sistemleri

Yazılım sistemleri, iş uygulamalarından tüketici ürünlerine kadar günlük yaşamın önemli bir kısmında yer alır. Çoğu insan zaman zaman hatalı, doğru çalışmayan ya da beklenen sonuçları vermeyen yazılımları kullanma durumunda kalmıştır. Doğru çalışmayan bir yazılım, para, zaman ve itibar kaybı yanında ölüm ve yaralanma gibi yaşamsal bazı sorunlara da yol açabilir.

3.2.2 Hatalı yazılımların sebepleri

Her insan program kodunda ya da dokümanda hataya sebep olabilecek yanlışlar yapar. Eğer hatalı kod çalıştırılırsa sistem yapması gereken ya da yapmaması gereken bir işlemi başarısız bir şekilde gerçekleştirecektir. Her hata için geçerli olmasa da yazılımda, sistemde ya da dokümanda yapılan hatalar başarısızlığa sebep olacaktır.

Hatalar oluşabilir çünkü insanlar yanılabilirler, çünkü zaman kısıtlıdır, kod ve altyapı karmaşık olabilir, kullanılan teknolojiler zaman içerisinde değişebilir ve sistemler kendi arasında etkileşimde bulunabilir. Radyasyon, elektronik etkileşimler, yazılımın kullanıldığı donanım ortamının değişmesi gibi çevresel etmenlerden ötürü de başarısızlık ortaya çıkabilir.

3.2.3 Yazılım geliştirme ve bakım üzerinde testin rolü

Sistemlerin ayrıntılı test edilmesi ve dokümantasyonu, işlem sırasında problem oluşma riskini azaltır ve yazılımın kaliteli olmasına katkı sağlar. Sistem sürümü ortaya çıkmadan önce hatalar düzeltilirse, kullanıma hazır hale geldiğinde başarısızlık azalacaktır.

3.2.4 Ne kadar test yeterlidir?

Ne kadar test etmenin yeterli olacağına karar vermek risk seviyelerinin dikkate alınmasına bağlıdır, teknik riskler, güvenlik ve işyeri riskleri ve projenin sınırlandığı takvim ve bütçe gibi etkenler göz önünde bulundurulmalıdır.

Test yapmak, proje paydaşlarına müşteriye sunulmadan ya da bir sonraki geliştirme seviyesine geçmeden önce, yazılım sürümü ya da test edilen sistemle ilgili karar bildirimini yapmak için yeterli bilgiyi sağlar.

3.3 Yazılımda Test Nedir?

Test yapmaktan bahsedilirken genellikle test koşmak ya da programı çalıştırmak algılanabilir, ancak bu test etmenin bir kısmıdır; tüm test aktivitesi değildir. Test aktiviteleri test koşturmadan önce ve test koşturduktan sonra olarak ikiye ayrılabilir. Bu aktiviteler planlama, kontrol, test durumlarının oluşturulması, test senaryolarının tasarımı, sonuçların kontrol edilmesi, bitiş kriterlerinin belirlenmesi, test sürecinin ve test esnasında sistem hareketlerinin raporlanması, test aşaması bittikten sonra kapanış aktivitelerinin tamamlanması şeklinde sıralanabilir. Test yapmak aynı zamanda doküman gözden geçirme ve statik analiz yapmayı da gerektirir.

Hem statik test hem de dinamik test aynı amaçlar için kullanılabilir ve test edilmiş sistem, geliştirme ve test süreçlerini iyileştirmek için bilgi sağlamada kullanılabilir.

Testin genel amaçları aşağıdadır:

- Hataları bulmak,
- Yazılım kalitesiyle ilgili güven kazanmak,
- Karar mekanizmaları için bilgi sağlamak,
- Hatalara engel olmak.

Yazılım yaşam döngüsü içinde testlerin erken tasarlanmasını içeren düşünme süreci ve aktiviteleri, kodun içinde hata oluşmadan engellenmesine yardımcı olabilir. Dokümanların gözden geçirilmesi ve sorunların tespiti ve çözümü kod içinde hataların oluşmadan engellenmesine yardımcı olacaktır.

Test ederken farklı görüşlerin varlığı farklı amaçlar için yararlı olacaktır. Örneğin geliştirme esnasındaki testlerde olabildiğince çok hata bulunması iyidir böylece bu hatalar tespit edilip, çözülecektir. Buna rağmen kabul testlerinde asıl amaç sistemin beklenildiği şekilde çalışmasıdır ve güven kazanmak için sistemin gereksinimleri karşılması gerekir. Bazı durumlarda testin amacı yazılımın kalitesini belirlemek ve belirlenen zamana kadar verilecek olan sistem sürümünün risklerini paydaşlara göstermektir. Bakım testinde ise geliştirme esnasında yapılan değişikliklerden sonra yeni hata olmadığına görülmesi gerekmektedir. Kullanım testi esnasında asıl amaç güvenilirlik ve geçerlilik gibi sistem karakteristiklerinin belirlenmesidir.

Sistemi çalıştırmak ve test yapmak farklı kavramlardır. Dinamik test hatalar sonucu ortaya çıkan sistemin başarısızlıklarını gösterir. Sistemi çalıştırmak ise başarısızlığın sebebini analiz eden ve yok eden bir geliştirme aktivitesidir. Testçi tarafından yeniden test yapmak, düzeltilen bir hatanın gerçekten çözüldüğünü garantiye almaktır. Bu aktiviteler için sorumluluk testçilerde ve yazılımcılardadır.

3.3.1 Yazılımda testin yedi temel ilkesi

Yazılım testinde yedi temel ilke aşağıda kısaca tanımlanmıştır:

a. Test hatalarının varlığını gösterir.

Test yapmak hataların varlığını gösterir ancak hiç hata kalmadığının garantisini veremez. Test yapmak yazılımda keşfedilememiş hata bulma olasılığını azaltır ama hiç hata bulunmazsa bu her şeyin doğru olduğunun bir kanıtı değildir.

b. Ayrıntılı test imkânsızdır.

Tüm kombinasyonları ve durumları test etmek mümkün değil veya para ve zaman nedeniyle olurlu değildir. Bunun yerine risk analizi yapmak ve test önceliklerine odaklanmak daha yararlı olacaktır.

c. Erken test önemlidir.

Yazılım geliştirme sürecinde test aktivitelerinin olabildiğince erken başlaması ve hataları erken bulmak tanımlanan amaçlara odaklanılmasını sağlayacaktır.

d. Hata kümelenmesi.

Test süreçleri, birimlerin beklenen hata yoğunluklarına odaklanmasına yardımcı olur. Yazılım birimleri çoğu hatayı müşteriye teslim sürümü öncesinde yapılan test esnasında fark eder.

e. Böcek Paradoksu.

Aynı kullanıcı senaryoları ve test durumlarının sürekli uygulanması sonucu aynı yazılım hatalarının tekrar tekrar bulunması, yeni hataların bulunmasını engeller. Sürekli aynı test senaryoları tekrarlanırsa yeni hata bulunamayacaktır. Böcek paradoksunun üstesinden gelmek için test senaryoları gözden geçirilmeli ve güncellenmelidir. Potansiyel yazılım hatalarını bulmak için olabilecek test senaryoları yazılmalıdır.

f. Test içerik bağımlıdır.

Test aşaması farklı yazılımlar için farklı uygulanır. Örneğin bir e ticaret sitesinin kritik güvenlik testi farklı uygulanır.

g. Hata yok yanılgısı.

Eğer sistem kullanıcı gereksinimlerini ve beklentilerini karşılamıyorsa ve sistem doğru şekilde kullanılamıyorsa, hataların bulunup giderilmesinin bir önemi kalmaz. Yazılımda hata yoktur ama beklentileri karşılamamaktadır. Yazılım kaliteli değildir.

3.3.2 Test süreci ve aşamaları

Test koşturmak test sürecinin en görünen tarafıdır. Etkili ve verimli olması açısından test planları, test durumlarının oluşturulma süresini, testin koşturulması ve geliştirilmesi aşamalarını ve hatta planlama süresini içermelidir.

Test sürecinin aşamaları aşağıda verilmiştir:

- a. Test planlaması ve kontrolü,
- b. Test analiz ve tasarımı,
- c. Test uygulamaları ve test koşturma aşaması,
- d. Bitiş kriterlerinin belirlenmesi ve raporlama,
- e. Test sonlandırma aktiviteleri.

Test sürecindeki aktiviteler ardı ardına olabileceği gibi eş zamanlı olarak da yürütülebilirler. Proje bu aktiviteleri mutlaka uygulamayı gerektirir.

a) Test planlaması ve kontrolü

Test planlaması testin amacının belirlenmesi aktivitesidir ve sözleşme üzerinde test aktivitelerinin amaca uygun yapıldığının göstergesidir. Test kontrolü planlamaya göre gerçek sürecin nasıl işlediğinin kontrolünün yapıldığı, plandan ne kadar sapıldığının rapor edildiği ve test süreci boyunca sürekli devam eden bir aktivitedir. Projenin amaçları ve görevleriyle uyumu için gerekli olan eylemleri içerir. Test kontrolü sırasında test aktiviteleri proje boyunca takip edilmelidir. Test planlaması takip ve kontrol aktivitelerinden geri dönüş olarak fayda sağlar.

b) Test analizi ve tasarımı

Test amaçlarının somut test senaryolarına ve test durumlarına dönüşme aşamasında olan aktivitelere test analizi ve tasarımı aktiviteleri denir.

Test Analizi ve Tasarımı aşağıdaki başlıca görevleri içerir.

- Gereksinimler, mimari yapı, risk analiz raporları gibi test kaynaklarının gözden geçirilmesi,
- Test amaçlarının ve test kaynaklarının test edilebilirliğinin değerlendirilmesi,
- Test maddelerinin analizi, yazılımın yapısı ve davranışı, anlaşma maddeleri sonucu ortaya çıkan test senaryolarının belirlenmesi ve önceliklendirilmesi,
- Yüksek öncelikli test durumlarının tasarlanması ve önceliklendirilmesi,
- Test senaryolarında ve test durumlarında kullanılacak olan test verilerinin belirlenmesi,
- İstenen altyapı ve test araçlarının belirlenmesi ve test ortamının tasarlanması,
- Test kaynakları ile test durumları arasında izlenebilirlik kurulması.

c) Test uygulamaları ve test kořturma ařaması

Test uygulamaları ve test kořturma aktiviteleri test durumlarının birleřtirilmesiyle belirlenmiř test prosedürlerinin uygulandıđı ve test kořturmak için gerekli olan herhangi bařka bir bilginin olduđu aktivitelerdir. Uygun ortam kurulur ve testler kořturulur.

Test uygulamaları ve test kořturma ařaması ařađıdaki bařlıca görevleri ięerir.

- Test durumlarının önceliklendirilmesi, uygulanması ve sonlandırılması,
- Test prosedürlerinin geliřtirilmesi, test verilerinin oluřturulması, test kořturulması, test senaryolarının otomatikleřtirilmesi,
- Testin verimli uygulanması için test prosedürlerinden test gruplarının oluřturulması,
- Test ortamının dođru kurulduđunun dođrulanması,
- Test kaynakları ile test durumları arasında iki taraflı izlenebilirlik kurulduđunun dođrulanması,
- Plana göre test prosedürlerinin manuel ya da test araęları kullanılarak uygulanması,
- Test uygulaması sonucunda test edilen yazılım sürümlerinin, test araęlarının ve test yazılımlarının kayıt altına alınması,
- Beklenen sonuçlarla elde edilen sonuçların karřılařtırılması,
- Sıra dıřı ve özel durumların kaydedilmesi ve bu durumların, kodun içindeki hatanın, test esnasındaki test verisinin, test dokümanının, test esnasındaki çevresel faktörlerin hangisinin sonucunda ortaya ęıktıđının analiz edilmesi,
- Bu sonuçların çeręevesinde sıra dıřı durumların ve özel durumların oluřtuđu test aktivitelerini tekrar etmek.

d) Bitiř kriterlerinin belirlenmesi ve raporlama

Test uygulaması beklenen hedefe ulařtıktan sonra bitiř kriterlerinin belirlenmesi gereklidir. Bu iřlem her test ařaması için yapılmalıdır.

Bitiř kriterlerinin belirlenmesinde ařađıdaki noktalar göz önünde tutulmalıdır:

- Test planında belirlenmiř olan bitiř kriterleri tekrar kontrol edilmelidir,

- Daha fazla test gerekiyorsa ya da plandaki bitiş kriterlerinin değiştirilmesi gerekiyorsa bunlar belirlenmelidir,
- Paydaşlar için test özet raporu yazılmalıdır.

e) Test sonlandırma aktiviteleri

Test sonuç aktiviteleri deneyimleri birleştirmek için etkilerin ve istatistiklerin toplanması için tamamlanmış test aktivitelerinden veri toplar. Bu aktiviteler yazılım sisteminin teslimatı, projenin tamamlanması gibi projenin kilometre taşları içinde ortaya çıkar.

Test Sonuç aktivitelerinin belirlenmesi aşağıdaki aşamaları içerir:

- Teslimatı planlanan sürümlerin teslimatlarının yapıldığının kontrol edilmesi,
- Hata raporlarının kapatılması ya da açık kalan değişim raporlarının üzerine gidilmesi,
- Sistemin onay dokümanlarının oluşturulması,
- Test yazılımının, test ortamının ya da test altyapısının daha sonra kullanılmak üzere arşivlenmesi,
- Test yazılımının bakım için hazır hale getirilmesi,
- Gelecekteki teslimatlar ve projeler için gerekli olan değişikliklerin belirlenmesi yapılan projelerden ders alınması,
- Test olgunluğunu geliştirmek için toplanan bilgilerin kullanılması.

3.3.3 Yazılım test seviyeleri

Yaşam döngüsü modelinde kaliteli test yapmak için birçok karakteristik vardır. Her yazılım geliştirme aktivitesine karşılık gelen bir test aktivitesi vardır ve her test seviyesi için o seviyede uygulanan test hedefleri vardır. Belirlenen test seviyesi için yapılan test analiz ve tasarımının geliştirme ile eş zamanlı yürütülmesi gerekir. Testçilerin geliştirme aşamasında taslaklar hazır olduğu anda sürece dahil edilmesi gerekmektedir. Test seviyeleri projenin akışına ve sistemin mimarisine göre birleştirilebilir ya da yeniden yapılandırılabilir. Her test seviyesi için genel hedefler, test senaryosu oluşturmak için referans verilmiş ürünler, bulunan tipik

hatalar, test kořturma esnasındaki istekler ve araç destekleri gibi özel yaklaşımlar ve sorumluluklar vardır.

a) Bileřen testi

Bileřen testi aynı zamanda birim, modül ya da program testi olarak da adlandırılabilir. Bileřen testi test edilebilir olan program, modül ve sınıfların(class) işlevlerini doğrular ve hataları bulmak için arama yapar. Sistemin ve geliştirme aşamasının durumuna göre sistemin geri kalan kısmından ayrı uygulanması gerekir.

Bileřen testi temel işleyiş sağlamlık ve yapısal testler gibi hem fonksiyonel hem de fonksiyonel olmayan karakteristikleri kapsmalıdır. Test senaryoları, bileřenlerin sözleşmede yazılmış olan yazılım tasarım ve veri modellerine göre belirlenmelidir.

Bileřen testi, test edilmiş olan kodun kullanımıyla ve geliştirme çevresinin desteğiyle birlikte yürür. Uygulamada ise bileřen testine kodu yazan programcı dahil olur. Hatalar buldukları anda düzeltilir. Bileřen testine başka bir yaklaşım ise test senaryolarının kod yazılmadan önce hazırlanmasıdır. Bu yaklaşıma ilk test yaklaşımı ya da test öncelikli geliştirme denir. Bu yaklaşım, test senaryolarının geliştirme sürecinden sonra kodun küçük parçalar halinde inşa edilip birleştirilmesine ve bileřen testlerinin testler geçene kadar uygulanmasına dayanır.

b) Entegrasyon testi

Entegrasyon testi, bileřenler arasındaki arayüzleri, dosya sistemi donanım gibi sistemin farklı parçalarının birleştirilmiş hallerini test etme amacıyla tasarlanır. Entegrasyon testinin birden fazla aşaması vardır. Bileřen entegrasyon testi, bileřen testleri yapıldıktan sonra ve bileřenler entegre edildikten sonra bileřenlerin aralarındaki etkileşimi görmek için yapılan testtir. Sistem entegrasyon testi ise farklı sistemleri ya da yazılımla donanım arasındaki etkileşimleri görmek için yapılan testtir ve sistem testinden sonra yapılması gerekir. Bu aşamada geliştiriciler arayüzün sadece bir tarafını görürler ve bu riskli olabilir. Bu yüzden çoklu platformlar geliştirilmelidir. Hataları bileřenlerden ve sistemlerden izole etmek çok daha zordur, risk artacak ve daha çok zamana ihtiyaç durulacaktır.

Sistematik entegrasyon stratejileri sistem mimarisine, fonksiyonel görevlere, işlem dizilerine ya da sistemin ya da bileşenlerin bazı farklı yönlerine dayanmaktadır. Sırasıyla hata izolasyonunu kolaylaştırmak ve hataları erken bulabilmek için entegrasyon arttırılabilir olmalıdır. Fonksiyonel test kadar fonksiyonel olmayan test de entegrasyon testi içerisinde yapılabilir.

Testçiler entegrasyon mimarisini anlamalı ve entegrasyon planına dahil olabilmelidirler. Eğer entegrasyon testleri bileşenlerden önce ve sistem inşa edilmeden önce planlanırsa, bu bileşenler sırasıyla daha etkili test edilerek inşa edilebilirler.

c) Sistem testi

Sistem testi sistemin ya da ürünün genel davranışıyla ilgilidir. Test ortamı üretim ortamına olabildiği kadar uygun olmalıdır. Böylece test sırasında görülemeyen çevreye bağlı hataların oluşma olasılığı azalacaktır.

Sistem testleri risk odaklı, anlaşmadaki gereksinimleri, iş süreçlerini, sistem davranışlarını açıklayan dokümanları, sistemin işletim sistemleriyle etkileşimlerini kontrol edebileceğimiz test durumları içermelidir.

Sistem testleri, sistemin fonksiyonel olan ve olmayan gereksinimlerini ve veri kalite karakteristiklerini kontrol etmelidir. Testçiler tamamlanmamış ve dokümana yansıtılmamış gereksinimlerle de ilgilenmek durumunda kalacaklardır. Fonksiyonel gereksinimler için olan sistem testleri, sistemin görünen anlaşmada kararlaştırılmış tekniklerinin test edilmesiyle başlar. Altyapı tabanlı teknikler ise daha sonra yapısal bir elemanla alakalı olarak testin titizliğini göstermek için yapılabilir.

d) Kabul testi

Kabul testi müşterilerin ya da kullanıcıların sisteme karşı olan güvenilirliklerini arttırır. Kabul testinde amaç sisteme karşı ya da sistemin bir parçasına ya da sistemin fonksiyonel olmayan karakteristiklerine karşı güven oluşturmaktır. Kabul testinde amaç hata bulmak değildir sistemin piyasaya çıkmaya ve kullanılabilirliğe hazır olduğunun göstergesidir. Kabul testi yazılım döngüsünün farklı zamanlarında ortaya çıkabilir. Bir bileşenin kullanılabilirliğini test etmek için kabul testi birim testi esnasında yapılabilir ya da yeni fonksiyonel bir iyileştirme için kabul testi sistem testinden önce yapılabilir.

Kabul testinin bileşenleri aşağıda listelenmiştir:

- **Kullanıcı testi**

Sistemin uygunluğu ve özel bir iş için yapılandırılan sistemin kullanılabilirliği çalışanlar tarafından kontrol edilir.

- **İşletim testi**

Sistemin kabulü, destek testi, acil durum kurtarma, kullanıcı yönetimi, bakım görevleri, veri yükü ve güvenlik için periyodik kontrollerini içerecek şekilde sistem yöneticisi tarafından yapılır.

- **Sözleşme ve düzenleme testleri**

Sözleşme kabul testleri müşteriye yönelik geliştirilmiş yazılımlarda sözleşmenin güvenilirliğine karşılık yapılan testlerdir. Kabul kriterleri sözleşme esnasında belirlenir. Düzenleme kabul testleri ise devletle anlaşmalı, legal ya da güvenlik düzenlemeleri gibi sözleşmelerde yapılması zorunlu olan testlerdir.

- **Alfa ve Beta testi**

Yazılım satışa sunulmadan önce yazılımcılar potansiyel müşterilerden geri besleme almak isterler. Alfa testi sadece geliştiricilerin kendi yerlerinde uyguladığı bir testtir. Beta testi ise potansiyel müşteriler tarafından kendi ortamlarında kullanılır.

3.3.4 Test türleri

Bir grup test aktivitesi yazılım sisteminin belirli bir kısmının doğrulanması amaçlanarak oluşturulmuş olabilir. Bunlar yazılım için geliştirilmiş bir fonksiyon, güvenilirlik ya da kullanılabilirlik gibi fonksiyonel olmayan bir kalite karakteristiği, yazılımın ya da sistemin yapısı ya da mimarisi olabilir.

a) Fonksiyonel test

Sistemin, alt sistemin ya da bileşenin fonksiyonları, gereksinim dokümanında anlatılmış olabilir ya da dokümanda belirtilmemiş de olabilir. Fonksiyonlar sistemin ne yaptığını anlatır.

Fonksiyonel testler fonksiyonlara, özelliklere ve ayrı sistemlerin birlikte çalışabilirliğine dayalıdır ve bileşen test aşamalarında uygulanmış olabilir. Sözleşmeye dayalı test teknikleri sistemin ya da yazılımın fonksiyonelliğinden

test durumları ve test senaryoları türetilerek oluşturulabilir. Fonksiyonel test yazılımın dış ortama karşı olan durumunu göz önünde bulundurur. Fonksiyonel testin bir tipi olan güvenlik testi, virüs gibi tehditlerin algılanması sonrasında fonksiyonun incelenmesidir. Başka bir tip fonksiyonel test olan birlikte çalışabilirlik testi ise yazılım ürünün bir ya da birden fazla sistemle birlikte çalışabilme kapasitesini ölçer.

b) Fonksiyonel olmayan test

Fonksiyonel olmayan test performans testini, stres testini, kullanılabilirlik testini, bakım testini, güvenilirlik ve taşınabilirlik testleri gibi bunlara benzer birçok testi kapsar. Fonksiyonel olmayan test sistemin nasıl çalıştığını kontrol eder.

Fonksiyonel olmayan test tüm test aşamalarında uygulanabilir. Fonksiyonel olmayan test terimi sistemin karakteristiklerini ölçen testlerin yapıldığını ifade eder. Bu testler bir kalite modeli için referans gösterilebilir. Fonksiyonel olmayan testler de yazılımın dış ortama karşı olan durumunu göz önünde bulundurur.

c) Yapısal / Mimari test

Yapısal test diğer adıyla beyaz kutu testi tüm test seviyelerinde uygulanabilir. Yapısal test, sırasıyla testin titizliğini ölçmeye gereksinimlerde istenen teknik testlerden sonra en iyi uygulanacak testtir.

Kapsam, bir test ortamı vasıtasıyla yazılımı çalıştıran ve testin, kodun ne kadarını kapsadığını yüzde halinde ortaya çıkaran bir ölçüttür. Eğer kapsam yüzde yüz değilse kapsam oranını arttırmak için daha fazla test tasarlanmalıdır.

Tüm test seviyelerinde özellikle bileşen ve bütünleştirme testlerinde test araçları bileşenlerin kod kapsamını ölçmek için kullanılabilir. Yapısal test yaklaşımı genelde sistemin mimarisine dayalı olur.

d) Değişikliklere dayalı test: Tekrar testi ve regresyon testi

Bir hata bulunduktan ve giderildikten sonra hatanın başarıyla yok edildiğinin doğrulanması için yazılım üzerinde tekrar testi yapılmalıdır. Buna doğrulama adı verilir.

Regresyon testi ise kodda yapılan deęişikliklerden sonra ya da deęişiklikler sonucunda çıkabilecek hataların bulunabilmesi için daha önce test edilmiş bir programın tekrar test edilmesidir. Bu hatalar yazılımın test edilmiş parçalarında ya da deęişikliklerle alakasız birimlerde ortaya çıkabilir. Yazılım ortamı deęiştğinde de regresyon testi uygulanması gerekebilir. Regresyon testinin yapılabilirliği açısından testler tekrarlanabilecek şekilde oluşturulmalıdır.

e) Bakım testi

Kurulumu tamamlanmış bir yazılım yıllarca kullanılır. Bu süre boyunca sistemin yapısı, çevresi kontrol altında tutulur çünkü çevre deęişebilir ya da genişleyebilir. Bu nedenle yeni sürüm çıkma aşamasında bakım testi çok kritiktir. Bakım testi, sistemin başka bir ortama taşınması aşamasında yazılımın bu ortama uyum sağlayıp sağlamayacağını test edilmesi açısından da çok önemlidir.

3.3.5 Statik teknikler

Yazılımın uygulamasını gerektiren dinamik testin aksine, statik test teknikleri manuel uygulamalara ve kodun statik analizlerine dayanır. Gözden geçirme yazılım ürünlerinin test edilmesi için farklı bir yoldur ve dinamik test uygulamasından önce yapılması iyi olabilir. Yaşam döngüsünde gözden geçirme yapılması hataların erken bulunmasına yardımcı olur ve testler koşulduktan sonra hata bulunmasından çok daha az maliyete neden olur. Gözden geçirme tamamen manuel aktivite olarak uygulanabilir ancak bu iş için yapılmış araçlar da vardır. Herhangi bir yazılım ürünü gereksinimlerin içeriğine, projenin tasarımına, koda, test planlarına, test durumlarına ve senaryolarına ve ya kullanıcı kılavuzuna göre gözden geçirilebilir.

Gözden geçirme hataların erken bulunması, doğrulanması, verimlilik, geliştirme ve test maliyetini azaltmak açısından yararlıdır. Gözden geçirme, gereksinimlerde ihmal edilmiş olan eksiklikleri ortaya çıkarabilir ve farklı gözden geçirme teknikleri kullanılarak gereksinimler açısından tamamlayıcı olabilir.

Gözden geçirme teknikleri formal ya da formal olmayan şekillerde uygulanabilir. Eğer gözden geçiren çalışanlar için yazılı bir öğretici doküman ya da yazılı bir

gözden geçirme raporu yoksa formal değildir. Eğer gözden geçirme sonrası belirlenen bir amaç varsa, testçiler ve takımdakiler eğitim alıyorsa gözden geçirme süreci formal bir şekilde ilerliyor demektir. Formal gözden geçirme planlama, başlangıç, bireysel hazırlık, sonuçların değerlendirilmesi ve raporlanması, bulunan eksikliklerin giderilmesi ve iş takibi aşamalarından oluşur.

3.3.6 Test tasarım teknikleri

Test geliştirme aşaması formal olan ya da olmayan farklı yollarla uygulanabilir. Formalitenin seviyesi testin durumlarına, olgunlaşma ve geliştirme sürecine, zaman sınırlamasına ve düzenleyen kişilere göre değişir.

Test durumlarından gereksinimlere ve anlaşma maddelerine izlenebilirlik kurmak, gereksinimlerin değişimden sonra daha etkili analiz edilmesini sağlar. Test analizi sırasında ayrıntılı test yaklaşımı, test tasarım tekniklerinin seçilmesinde kolaylık sağlar.

Test tasarımı sırasında test durumları ve test verileri oluşturulur. Bir test durumu girdi değerleri kümesinden, testin koşulları için gerekli olan ön koşullardan beklenen sonuçlardan ve test bitiş durumlarından oluşur. Testin uygulanması esnasında test durumları test prosedür anlaşmasında geçen maddelere göre geliştirilir, uygulanır ve öncelik durumları belirlenir.

Test tekniklerini klasik anlamda kara-kutu ya da beyaz-kutu testi şeklinde ayırabiliriz. Kara kutu ve beyaz kutu testleri yazılımcıların ve testçilerin deneyimlerine dayanarak birlikte de yürütülebilirler.

a) Kara-Kutu teknikleri (Black-box Techniques)

Kara kutu test tasarım teknikleri, temel test dokümanına bakılarak test senaryolarının seçilmesi, test durumlarının yaratılması ve test verilerinin oluşturulması üzerindedir. Bu testler fonksiyonel olan ve olmayan test durumlarını içerir. Kara kutu testi hiçbir bileşenin içinde bulunan yapıyla ilgilenmez.

- **Eşdeğer Aralık Testleri (Equivalence Partitioning):** Girdileri benzer ya da birbirine yakın sonuç verecek yazılım ya da sistemler gruplandırılarak aynı şekilde test işlemine tabi tutulurlar. Çıktılar tiplerine göre, arayüz

parametrelerine göre geçerli ya da geçersiz veriler içermeye durumlarına göre farklı gruptandırılabilirler.

- **Sınır Değer Analizi (Boundary Value Analysis):** Bölümlerin içinde bulunan değerler ve sınır değerleri farklı davranışlar sergilerler. Bu sebepten sınır değerler en çok hatanın çıkabileceği alanlardır. Bir bölümün maksimum ve minimum değerleri o bölümün sınır değerlerini oluşturur. Testler hem geçerli hem de geçersiz sınır değerleri test edecek şekilde tasarlanmalıdır ve testler esnasında her sınır değer için bir test uygulanmalıdır.
- **Karar Ağacı Testi (Decision Table Testing):** Karar ağaçları, mantıksal durumlar içeren sistem gereksinimlerini ele almak açısından iyi bir yoldur. Uygulanması gereken bir sistemin karmaşık kurallarını kaydetmek için kullanılabilir. Karar ağacı oluşturulurken anlaşma analiz edilir ve sistemin durumları, davranışları belirlenir. Girdi durumları ve davranışları doğru ya da yanlış değer alacak şekilde belirtilir. Karar ağaçları bütün girdi değerlerinin doğru ya da yanlış olan tüm kombinasyonları için tetikleme durumlarını içerir.
- **Durum Geçiş Testi (State Transition Testing):** Bir sistem mevcut durumlar için ya da senaryolar için farklı yaklaşımlar sergileyebilir. Bu durumda sistemin görünümü durum geçiş diyagramıyla gösterilebilir. Geçiş diyagramları testçiye yazılımın farklı durumlarda nasıl davrandığıyla ilgili, durumlar arası geçişlerle ilgili yol gösterir. Sistemin test sırasındaki durumları ayrıştırılabilir ve tanımlanabilir. Durum tablosu girdilerle durumlar arasındaki ilişkiyi gösterir ve geçersiz durumların daha önceden belirlenmesine yardımcı olur.
- **Kullanım Senaryosu Testi (Use Case Testing) :** Testler kullanım senaryolarından türetilirler. Kullanım senaryosu aktörler arasındaki etkileşimi anlatır ve kullanım senaryoları teorik olarak ya da sistem seviyesinde açıklanabilir. Her kullanım senaryosu başarılı olması için gereken ön koşullara sahip olduğu gibi senaryo bittiğinde sistemdeki sonuç durumlarını da anlatır.

b) Beyaz-Kutu teknikleri(White-box Techniques)

Beyaz kutu test tasarım tekniklerinde ise bir bileşenin ya da sistemin yapısı analiz edilir. Beyaz kutu testleri direk olarak kodun içindeki fonksiyonlarla ve döngülerle ilgilidir. Bileşen seviyesi, bütünleştirme seviyesi ve sistem seviyesi olarak üç ana başlık altında toplanabilirler.

- **Durum Testi ve Kapsam(Statement Testing and Coverage):** Bileşen testinde yapılan durum kapsamı testi, test aracı üzerinde denenmiş olan uygulanabilir durumların yüzdelerini değerlendirir. Durum testi tekniği özel durumlarda koşturulmak üzere test senaryoları üretilen bir tekniktir.
- **Karar Yapısı Testi ve Kapsam (Decision Structure Testing and Coverage):** Karar yapısı kapsamı, doğru, yanlış gibi ya da (If_then) durumları gibi durumlar içeren sınıfların test edilmesiyle ilgilidir ve bir test aracında uygulanan karar yapısı testi sonuçlarının yüzdeleri değerlendirilir. Karar yapısı test teknikleri özel karar yapıları üzerine yazılan test senaryoları ile uygulanır. Sınıflar kod içinde bulunan karar noktaları ile türetilirler ve kodun farklı yerlerindeki geçişlerin kontrollerini yaparlar.

c) Deneyime dayalı teknikler

Testçinin benzer yazılım kodları üzerinde test deneyimi kazandığı ve hataların çıkabileceği parçaları ve durumları önceden tahmin ederek test senaryoları yarattığı tekniktir. Daha önceki sistem ya da projelerde benzer durumlar için uygulanmış olan test senaryoları ve dokümanlarından yararlanılarak yeni sistem için senaryolar yaratılabilir.

3.3.7 Test yönetimi

Test yönetimi yazılımcılarla kurulan ilişkiler açısından ve testçilerin bağımsız seçilmesi açısından önemlidir. Testlerin proje üzerinde etkili olabilmesi için bağımsız bir test ekibi ve buna bağlı olarak test yönetimi önemlidir.

Hataları bulmanın ve gözden geçirme aşamasının etkili olabilmesi için yazılımı yapan kişilerden bağımsız bir test ekibi olmalıdır. Büyük ve karmaşık projelerde çoklu test seviyelerinin olması ve bu seviyelerin bağımsız testçiler tarafından test edilmesi en iyisidir. Yazılımcılar da bağımsız testçiler tarafından yapılan alt seviyelerdeki test aşamalarına katkıda bulunabilirler. Testçiler test sürecinin

tanımlanmasında ve kuralların belirlenmesinde yetkili olmalıdırlar ancak proje yönetimi tarafından yönlendirilmelidirler. Test yöneticisi ise testçi arasındaki görev dağılımı iyi yapılmalıdır. Test yöneticisinin görevi planlamak, testleri uygulama esnasında takip etmek ve test aktivitelerini kontrol etmektir. Test senaryolarının hazırlanması, gözden geçirme aktiviteleri, test verilerinin uygun hale getirilmesi, testlerin otomatize edilmesi ve performansın ölçülmesi gibi aktiviteler ise testçinin görevleri arasındadır.

3.3.8 Test planlaması ve değerlendirilmesi

Planlama, sistem testi ve kabul testi gibi her test seviyesi için test planının içinde ya da test planından ayrı yapılabilir. Planlama aşamasında organizasyonun yapısına, testin kapsamına ve amacına, risklere, kaynakların test edilebilir olmasına ve uygunluğuna bakılır. Planlama sonrasında çok daha fazla detay ve oluşabilecek riskler ortaya çıkar. Test planlaması sürdürülmesi gereken ve tüm yazılım yaşam döngüsü boyunca kontrol altında tutulması gereken bir aktivitedir. Test planlama aktiviteleri aşağıdaki aşamaları içermelidir.

- Testin amacının kapsamının ve risklerinin belirlenmesi,
- Test seviyeleri tanımlarını, başlangıç ve bitiş kriterlerini içerecek şekilde tüm test yaklaşımının tanımlanması,
- Yazılım yaşam döngüsü aktiviteleri içinde test aktivitelerinin koordine edilmesi ve bütünleştirilmesi,
- Neyin nasıl test edileceği, test aktiviteleri içerisinde hangi rollerin uygulanacağı ve test sonuçlarının nasıl değerlendirileceği ile ilgili kararların verilmesi,
- Test analiz ve tasarım aktivitelerinin programlanması,
- Test uygulaması, koşturulması ve değerlendirilmesi aktivitelerinin programlanması,
- Farklı aktivite atamaları için kaynak atamasının yapılması,
- Test dokümantasyonu için detay seviyelerinin, test miktarlarının ve şablonların belirlenmesi,
- Test hazırlıklarını ve uygulamalarını kontrol etmek, hata çözümlerini, risk durumlarını belirlemek için metriklerin seçilmesi.

a) Test başlangıç ve bitiş kriterleri

Test başlangıç kriteri, testin ne zaman ve ne şekilde başlayacağını tanımlar. Test uygulamasına her test seviyesinin başlangıcında ya da bir grup test koşturulmaya hazır olduğunda başlanabilir. Başlangıç kriterleri aşağıdaki durumları içermelidir:

- Test ortamı hazır ve uygun olmalıdır,
- Test araçları test ortamına hazır olmalıdır,
- Test edilecek kod hazır olmalıdır,
- Test verileri hazırlanmış olmalıdır.

Test bitiş kriterleri ise testin ne zaman sona ereceğini tanımlar. Bitiş kriterleri aşağıdaki durumları kapsar:

- Kodun kapsamı, riskleri ve fonksiyonelliği gibi noktaların tamamlanması beklenir,
- Hata yoğunluğu ve güvenilirlik ölçüsü tahmin edilir,
- Ayrılan bütçe bitmiş olabilir,
- Takvimin sonuna gelinmiş olabilir ve testlerin sonlandırılması istenebilir.

b) Test değerlendirilmesi

Test deneyiminden sonra değerlendirme için iki farklı yaklaşım göze alınır. Bunlardan biri metriklere dayalı yaklaşımdır. Bu yaklaşımda geçmişte sunulan benzer projelere ya da belli değerlere bakılarak test değerlendirmesi yapılır. Diğer yaklaşım ise uzman yaklaşımıdır. Bu yaklaşımda ortaya çıkan iş, işi yapan kişi tarafından ya da uzmanlar tarafından değerlendirilir. Test deneyimi projenin karakteristiği, gereksinimlerin kalitesi, geliştirme sürecinin yapısı, testin sonuçları, maliyet, takvim gibi faktörlere dayalıdır.

3.3.9 Test stratejileri ve test yaklaşımları

Test yaklaşımı belirli projeler için test stratejisinin uygulanmasıdır. Test yaklaşımı test planında ve test tasarımında tanımlanır. Genel olarak projenin amaç ve risk değerlendirmesinin yapılmasına yönelik kararları içerir. Bu yaklaşım, test tasarım tekniklerinin seçilmesi ve test başlangıç, bitiş kriterlerinin belirlenmesi gibi test

süreçlerindeki test planlamasının başlangıç noktasıdır. Birkaç yaklaşımla özetlemek gerekirse; büyük risk içeren alanlardaki testlere yönelmek analitik yaklaşım olarak adlandırılır. Hata oranlarıyla ilgili ya da kullanım alanlarıyla ilgili istatistiksel bilgiler alarak yapılan test, model tabanlı yaklaşım sonucunda oluşur. Hata bulmaya dayalı, deneyime dayalı ve kalite karakteristiklerine dayalı yapılan testler sistemli yaklaşım sonucu ortaya çıkar. Bunlara benzer farklı yaklaşımlar üretilebilir.

3.3.10 Test raporlaması ve kontrolü

Test raporlaması, zamanlama ve bitiş kriterlerinin sağlanması gibi test periyodlarında ne olduğuyula ilgili özet çıkarmakla ilgilidir. Gelecekte yapılacak aktivitelerde bu bilgiler analiz edilir ve tavsiye olacak şekilde desteklenir. Metrikler test seviyesinin sonunda ya da test esnasında toplanmalıdır. Bulunulan seviye için test beklentilerinin yeterli olmasına, belirlenen test yaklaşımının yeterliliğine ve testin ne kadar etkili olduğuna dair metrikler toplanır.

Test kontrolü ise metriklerden elde edilen ve raporlanan bilgi sonucunda yönlendirilen aktivitelerin yapıldığı aşamadır. Takvim değişimi, testin kontrolü sırasında verilen kararlar, önceliklendirilen testler gibi kontrol aktiviteleri bu seviyede gerçekleştirilir.

3.3.11 Proje / Ürün riskleri ve test

Risk bir olayın istenmeyen durumlarda ya da potansiyel problemler sonucunda ortaya çıkan tehlike ya da tehdit olasılığı olarak tanımlanabilir. Risk seviyesi, olmaması gereken bir durum olma ihtimali göz önünde bulundurularak tanımlanır.

a) Proje riskleri

Proje riskleri projenin amaçladığı hedefi yerine getirene kadar geçirdiği süreçte içerdiği tüm risklerdir. Bu riskler organizasyon ile ilgili faktörler, teknik konular ya da üretici ile ilgili konular olabilir. Beceri ve personel eksikliği, kişisel konular, politik konular, testçi ve yazılımcılar arasındaki iletişim problemleri, test sonuçlarından beklenen orana ulaşılamaması sonucunda oluşan uygunsuz davranışlar organizasyon riskleri olarak sayılabilir. Gereksinimlerin doğru tanımlanması, test çevresinin zamanından hazır olmaması, verilerin geç dönüştürülmesi, geliştirme, test dönüşümlerinde geç kalınması ve düşük

kaliteli tasarım ve test verileri kullanılması teknik proje riskleri olarak sayılabilir. Bu riskleri analiz ederken, yönetirken ve azaltırken test yönetiminin proje yönetim prensiplerini iyi değerlendiriyor olması beklenir.

b) Ürün riskleri

Yazılımda ya da sistemde bulunan potansiyel hata alanları ürün riskleri olarak bilinir ve ürünün kalitesi için risk oluşturur. Yazılım ve donanım tek başına ya da firma genel olarak potansiyel zarar içerebilir. Güvenilirlik, kullanılabilirlik gibi konularda yazılım karakteristikleri zayıf olabilir. Veri kalitesi düşük olabilir ve yazılım kendisinden beklenen gereksinimleri karşılamıyor olabilir.

3.3.12 Test araçları

Test araçları teste destek olmak için kullanılabilir. Test koşturma araçları, veri üretim araçları ve sonuç karşılaştırma araçları test sürecinde direk olarak kullanılırlar. Hata yönetim araçları ve gereksinim izleme araçları gibi araçlar ise test sürecinin yönetilmesine yardımcı olurlar. Test araçları test aktivitesini otomatize ederek testin daha etkili olmasına ve güvenilir olmasına yardımcı olur. Genişletilebilir ve tekrar kullanılabilir test kütüphaneleri test araçlarının kurulması kullanılabilir.

a) Test araçlarının sınıflandırılması

Farklı test uygulamaları için farklı test araçları vardır. Araçlar teknolojik, amaca yönelik, ticari gibi birçok kriterlere göre sınıflandırılabilir. Bazı araçlar tek bir test aktivitesine destek olurken bazıları birden çok aktivite için kullanılabilir. Test yönetim araçları, gereksinim yönetim araçları, hata yönetim araçları ve konfigürasyon yönetimi araçları test sürecinin yönetiminde etkili olan araçlardır. Gözden geçirme araçları, statik analiz araçları ve modelleme araçları ise statik test için kullanılan test araçlarıdır. Test tasarım araçları ve test verisi hazırlama araçları ise anlaşmaya destek veren test araçlarıdır. Test koşturma ve bileşen testi için yapılan araçlar, test karşılaştırma araçları, kapsam ölçme araçları ve güvenlik testi araçları ise test uygulamasına ve koşturulmasına destek veren araçlardır. Dinamik analiz araçları, performans testi, yükleme testi, stres testi ve test izleme araçları gibi araçlar ise performansa destek veren araçlardır.

b) Test uygulamasında test araçlarının kullanılmasının yararları ve riskleri

Test uygulamasında test araçlarının kullanılmasının potansiyel yararları ve araç kullanılmasının yaratabileceği fırsatlar vardır. Ancak test araçlarının kullanılması bazı riskler içerebilir.

Test araçlarının kullanılması tekrarları azaltır, daha tutarlı ve objektif değerlendirme sunar ve testlerle ilgili bilgiye ulaşmayı kolaylaştırır. Aynı zamanda test araçlarının kullanılması birçok risk de içerir. Test aracı kullanıldığı için beklentiler artabilir. Test aracının yapılandırılması için zaman planı yapılmassa proje takvimi gecikecektir. Test aracı kullanılarak oluşturulan test senaryoları için daha çok çaba harcamak gerekebilir ayrıca testlerin sürüm kontrolleri ihmal edildiği için takip etmek zorlaşacaktır. Gereksinim yönetim araçları, hata yönetim araçları ve sürüm kontrolü araçları gibi araçların aynı anda kullanılması birbirlerini etkileyebilir ve olumsuz sonuçlar doğurabilir.

4. TMMi (TEST MATURITY MODEL INTEGRATION) YAKLAŞIMI

4.1 Giriş

Organizasyonlar hemen her gün ürünlerin piyasaya çıkış süresini düşürmek, müşteri gereksinimlerini tam olarak karşılayabilmek veya yüksek kalite seviyesine erişmek ve uygun maliyet gibi zorlu iş şartlarıyla karşılaşır. Yazılımın baskın olduğu sistemleri üretmek daha da zahmetlidir. Bu sistemler zamanla günlük yaşamda ekonomik ve sosyal olarak daha önemli bir rol oynamaya başlamışlardır. Yeni yaklaşımlar, teknikler ve araçlar yazılım geliştirme ve bakım görevlerini desteklemek için uygun hale getirilmektedirler. Çünkü kalite konusuna odaklanmasa için yazılım mühendisliği disiplini üzerinde bir baskı oluşmaktadır. Düşük kaliteli yazılım toplum için kabul edilebilir değildir. Yazılım başarısızlıkları büyük kayıplarla sonuçlanabilmektedir. Bu durumda yazılım geliştirmede izlenen test disiplini en önemli kalite ölçütlerinden birisi olarak kabul edilmektedir. Böylece test etmek sadece ürün kalitesi için değil, tüm geliştirme ve üretim süreci performansını etkileyen anahtar bir aktivite haline gelmiştir.

Son on yılda yazılım endüstrisi ürün kalitesini artırmak için önemli yatırımlar yapmış ve çok çaba sarf etmiştir. Yazılımın boyutu ve karmaşıklığı artarken aynı zamanda paydaşlar ve kullanıcılar da çok titiz davrandıklarından bu zor bir iş haline gelmiştir. Bununla birlikte yazılım geliştirme, organizasyonun kurum dışındaki organizasyonlara yaptırdığı bir aktivite haline gelmiştir. Çeşitli kalite iyileştirme yaklaşımlarından oldukça olumlu sonuçlar alınmasına karşın yazılım endüstrisi hala sıfır hatadan tamamen uzaktır. Ürün kalitesini iyileştirmek için yazılım endüstrisi genelde yazılım geliştirme sürecini iyileştirmeye odaklanır. Geliştirme sürecini iyileştirmek için de CMMI süreci kullanılır. CMMI yazılım iyileştirme süreci için kabul edilen bir endüstri standardıdır. Test etme genelde toplam proje maliyetinin %30-%40'ını tutmasına rağmen CMMI gibi geliştirme modellerinde dahi yazılım sürecinde teste yeterli önem verilmez. Bu yüzden test dünyası kendi iyileştirme modelini oluşturmuştur. TMMi test sürecinin iyileştirilmesi için detaylandırılmış bir modeldir ve CMMI yaklaşımını tamamlayıcı niteliklere sahip olacak şekilde oluşturulmuştur.

TMMi yaklaşımı test sürecinin geliştirilmesi için dünya standartlarını hızla yakalayan bağımsız bir test olgunluk modelidir. Test süreci geliştirme iş yönetimindeki önemini anlatır ve organizasyonun ya da projenin test süreci geliştirme aşamasına rehberlik eder. ISTQB, yazılım test uzmanlığı sertifikasyonu sağlayan ve organizasyon ya da projenin yaklaşımından çok test uzmanlığı alanında çalışmak isteyen kişilere yönelik farklı seviyelerde kılavuzlar ve sertifikasyonlar sağlayan bir kuruluştur. TMMi modeli ise projeler ve organizasyonlar üzerinde uygulanabilir.

4.2 TMMi'in Gerekçesi ve Tarihçesi

TMMi, TMMi adlı bir vakıf tarafından test iyileştirme amacıyla geliştirilen bir kılavuzdur. Ayrıca CMMI modeli için tamamlayıcı bir modeldir. CMMI, test yöneticileri, test mühendisleri ve yazılım kalite mühendisleri için önemli olan geliştirme süreci konularını işaret etmektedir. TMMi' da ise test, tüm yazılım ürünü kalitesini kapsayacak şekilde geniş açıdan uygulanmaktadır.

CMMI aşamalarındaki gösterimlere benzer şekilde TMMi da sürecin geliştirilmesi ve iyileştirilmesi için olgunluk seviyesi kavramını kullanmaktadır. Ayrıca test süreç alanları, hedefler ve uygulamalar tanımlanır. TMMi olgunluk kriterlerini uygulamak test sürecini iyileştirmekte ve ürün kalitesi açısından, test mühendisliği aktiviteleri için ve zaman açısından katkı yaratmaktadır. Kaynaklar, test araçları ve iyi eğitilmiş test mühendisleri eksikse, hata önleme süreci kontrol mekanizması olgunlaşmamışsa TMMi için test, düzensiz gelişen ve sıkıntılı bir süreçtir.

Yapılan uygulamalardan çıkarılan sonuç TMMi'in test süreci belirlenirken daha etkili bir süreç yaratılmasına destek olduğudur. Test, yazılım geliştirme süreciyle tamamen birlikte yürütülen profesyonel bir iş halini almıştır. Testin odak noktası artık hata bulmak değil hata önlemek üzerinedir.

4.3 TMMi'in Kapsamı

Bu bölümde TMMi'in kapsadığı alanlar, test seviyeleri anlatılarak genel değerlendirme yapılacaktır.

4.3.1 Yazılım ve sistem mühendisliği

TMMi, hem sistem mühendisleri hem de yazılım mühendisleri için test aktivitelerine ve test iyileştirme süreçlerine destek olmak için ortaya çıkmış bir modeldir. Sistem mühendisliği yazılım içeren ya da içermeyen tüm sistemin geliştirilme aşamalarıyla ilgilenir. Yazılım mühendisliği ise yazılım sistemlerinin geliştirilmesiyle ilgilenir

4.3.2 Test seviyeleri

Test iyileştirme süreci için olan bazı modeller sadece yüksek test seviyelerine odaklanır. TMMi ise statik testi içerecek şekilde olan tüm test seviyelerini ve yapısal testin yönlerini içerir. Dinamik test yönü ile birlikte hem bileşen testi gibi düşük test seviyeleri hem de kabul testi gibi yüksek test seviyeleri TMMi'ın kapsamı içerisinde yer almaktadır.

4.3.3 TMMi ve CMMI

TMMi'ın CMMI modelinin tamamlayıcısı olarak kabul etmek önemlidir. TMMi 'ın benzer CMMI seviyelerinden ya da alt CMMI seviyelerinden süreç içerisinde ayrı bir destek alması gerekmektedir. İstisnai durumlarda yüksek CMMI seviyeleri ile de ilişki kurulabilir. Süreç esnasında ve ayrıntılı uygulamada CMMI, TMMi'ın içinde tekrarlanmaz sadece referans olarak gösterilir.

4.3.4 TMMi için genel değerlendirme

Birçok organizasyonda belirlenen amaçlar için, müşteri ve paydaşlar test ilerleme sürecindeki gelişmeyi kıyaslama yaparak değerlendirirler. Test süreci değerlendirmesi, iyileştirme fırsatlarının belirlenmesi ve organizasyonun bağlı olduğu pozisyon için seçilen model ya da standardın anlaşılabilmesine odaklanılır. TMMi değerlendirme esnasında kullanılacak üstün bir referans model sağlar. Değerlendirme ekibine bulguların tanımlanması ve önceliklendirilmesi için TMMi rehberlik eder. Bu bulgularla birlikte TMMi uygulamalarından elde edilen görüşler organizasyonun geliştirme planı için kullanılabilir. Kendi yapısal değerlendirmesi TMMi'ın bir parçası değildir. TMMi değerlendirmesi için olan gereksinimler TMMi tarafından tanımlanır. Bu gereksinimler, süreç değerlendirme olarak da bilinen SPICE(Software Process Improvement and Capability Determination) ISO/IEC 15504 standardını temel alır. Özel olgunlaştırma seviyelerinin başarısı farklı

organizasyon deęerlendirmeleri için de aynı anlama gelebilir. Tutarlılıktan emin olmak için koyulan kurallar TMMi deęerlendirme gereksinimlerinde bulunmaktadır.

TMMi, CMMI ile bütünleşmiş halde kullanılabilir. TMMi ve CMMI deęerlendirmeleri genellikle hem geliştirme sürecinde hem de test sürecinde deęerlendirilerek birleştirilir. Model yapıları benzer olduęu zaman ve model daęarcıęı ve amaçlar örtüştüęü zaman paralel çalışma ve paralel deęerlendirme, deęerlendirme takımı tarafından sonuçlandırılır. TMMi sürekli modellerde birleştirilmiş test noktalarına referans olmak için kullanılabilir. Testi kapsayan örtüşme alanları deęerlendirilebilir ve TMMi kullanılarak geliştirilebilir.

Test geliştirme sürecinde TMMi referans model olarak kullanılabilen bir temel sağlar. Test geliştirme sürecindeki en önemli adım sürece başlamadan organizasyondan destek almaktır. Verilen bilinçli yönetim desteęi test geliştirme sürecinde etkili bir yaklaşım kullanıldığına paydaşları ikna eder.

4.4 TMMi'in Genel Yapısı

TMMi'in yapısı genel olarak CMMI'in yapısına benzerdir. Bu önemli bir kazançtır çünkü birçok organizasyon CMMI yapısına aşinadır. CMMI yapısı, gerektirdięi hedefler ve tavsiye ettięi uygulamalar yönünden açıkça üstündür. Bu yönleri TMMi'a benzer.

TMMi bileşenleri gerekli olan, beklenen ve eğitici olmak üzere üç ayrı kategoride gruplanmıştır. Gerekli olan bileşenler bir organizasyonun bir test süreç alanında başarılı olabilmesi için neyi karşılaması gerektiğini tanımlayan parçalardır. Gerekli olan bileşenlerin TMMi içinde özel ve genel olmak üzere hedefleri vardır. Eğer bir test süreç alanı başarıya ulaştıysa hedeflere olan memnuniyet karar aşamasında deęerlendirme amacıyla kullanılır.

Beklenen bileşenler, gerekli olan bir bileşenin başarıyla sonuçlanması için ne uygulanması gerektiğini tanımlar. Beklenen bileşenler, geliştirmeyi uygulayan kişiye ya da deęerlendirmeyi yerine getiren kişiye rehberlik eder ve özel ve genel uygulamalar içerirler. Hedef deęerlendirilmeden önce her iki uygulamanın da planda tanımlanmış ve organizasyon tarafından süreç içinde uygulanmış, kabul edilebilir alternatifleri olmalıdır.

Eđitici bileşenler, gerekli olan ve beklenen bileşenlere uygulanacak olan yaklaşımla ilgili organizasyona yardım edecek detayları verir. Alt uygulamalar, örnekler, notlar, referanslar ve öğretici model parçaları gibi.

4.5 TMMi'ın Bileşenleri

TMMi modelinin bileşenleri bu bölümde ayrıntılı olarak anlatılacaktır.

4.5.1 Olgunluk seviyeleri

TMMi olgunluk seviyesi organizasyonun test kalite sürecinde hangi derecede olduğunu önemser. Her seviye, test sürecinin önemli bir parçasını aşamalı olarak yerine getirir. TMMi içinde beş olgunluk seviyesi vardır ve her olgunluk seviyesi bize o seviyede başarılı olmamız için neler yapmamız gerektiğini anlatır. Organizasyonun başarılı olduğu seviye ne kadar artarsa, test süreci de o kadar başarılı geçmiştir diyebiliriz. Olgunluk seviyesine ulaşmak için organizasyon, belirlenen seviye ve ondan önceki seviyeler için test süreç alanı içinde kendinden beklenen hedefleri karşılamalıdır. Her organizasyon en az TMMi 1 seviyesindedir. Bu seviyede hedeflenen ve beklenen amaç yoktur.

4.5.2 Test süreç alanları

Birinci seviye haricinde tüm olgunluk seviyeleri organizasyonun test sürecinde başarılı olmasına odaklı çeşitli süreç alanları içerir. Süreç alanları bir olgunluk seviyesinde başarıya ulaşılabilmesi için gerekli olan önemli noktaları tanımlar. Her test süreç alanı teste yönelik aktiviteleri kümeleyerek belirler. Aktivitelerdeki tüm uygulamalar yerine getirildikten sonra önemli olan geliştirme aktiviteleri bu alanın kapsayacağı şekilde yapılacaktır. TMMi'da sadece bu alanlar belirleyici faktörler olarak tanımlanan test süreç kapasitesinde göz önünde bulundurulur. Olgunluk seviyesinin tüm süreç alanları ve alt olgunluk seviyeleri hedef konulan olgunluk seviyesinin başarılı olması için onaylanmış olmalıdır. Örneğin bir organizasyon TMMi 3 seviyesindeyse, TMMi 2 ve TMMi 3 seviyelerindeki tüm süreç alanlarında başarılı olması gerekmektedir.

Amaç raporu, test süreç alanının amacını açıklar ve bilgi verici parçalardan biridir. Test süreç alanının "Açıklama Notları" bölümü ise test süreç alanında kapsanmış olan önemli kavramları açıklar. Kapsam bölümü test süreç alanı tarafından

tanımlanmış olan test aktivitelerini bölümünü belirler ve eğer mümkünse o test süreç alanı dışında kapsanması gereken test aktivitelerini ortaya çıkarır.

Genel amaçlar test süreç alanının sonuna doğru ortaya çıkar ve genel(generic) olarak adlandırılır çünkü aynı amaçlar farklı süreç alanlarında ortaya çıkabilir. Genel amaç, bir test süreç alanında uygulanmış olan alanların kurumlaştırması için gösterilmesi gereken karakteristikleri tanımlar.

4.6 Genel Amaçlar ve Genel Uygulamalar

Genel amaçlar ve genel uygulamalar çoğunlukla CMMI'dan türetilmiştir. Genel amaçlar sayısal sıra şeklinde organize edilmiştir. Genel uygulamalar da genel amaçlara destek olmak için genel amaçların altında sayısal sıra şeklinde organize edilmiştir. TMMi ve CMMI'ı aynı anda kullanmak isteyen organizasyonlarda olabilecek karışıklığı önlemek için CMMI'ın numaralandırılmış şeması olduğu gibi kabul edilmiştir.

Hedeflenen olgunluk seviyesi hangi amaç ve uygulamaların uygun olduğuna karar vermemize yardımcı olur. İkinci seviyeye ulaşmaya çalışırken ikinci seviyedeki süreç alanları ve genel amaçlara uygun olan genel uygulamalar da eşlik eder. Üçüncü seviyeye ulaşmak için sadece TMMi 3 için geçerli olan genel amaçlar uygundur. Eğer olgunluk, ikinci seviyeye ulaştıysa üçüncü olgunluk seviyesine ulaşılması için, ikinci olgunluk seviyesinin süreç alanlarına dönüş yapılmalı ve üçüncü seviye için belirlenen genel amaçlar uygulanmalıdır. Bu süreç alanlarına üçüncü seviye için geçerli olan uygulamalar eşlik edilmelidir.

Kurumsallaşma süreç geliştirmede önemli bir kavramdır. Genel amaç ve uygulama tanımlarından bahsedildiğinde, kurumsallaşma,işin yapıldığı alanda deneyim ve kökleşme anlamına gelir ve sürecin tamamlanması için gereken sorumluluk alınır. Süreç değişimi için gerekli olan gereksinimler ve amaçlar sürecin aktif ilerlediğinden emin olmak için sürecin uygulanmasında da değişikliğe ihtiyaç vardır. Genel uygulamalar organizasyonun baktığı yönü belirleyecek aktiviteleri tanımlar.

Yönetilen süreç, ürünlerin başarıyla ortaya çıkması için zorunlu olan işleri içeren süreçtir. Bu planlanmıştır ve yönetimle uyumlu bir şekilde yürütülür, deneyimli

personel kullanılır ve kontrollü çıktılar üretilmesi için elverişli kaynaklar vardır. Yönetilen süreçlerde kontrolü yapan, gözden geçirmelere tabi tutulan süreç tanımlarına bağlı kalınmasını değerlendiren ve konu hakkında bilgi sahibi olan paydaşlar vardır. Süreç bir grup ya da organizasyon tarafından temsil edilir. Stres altında oluşturulmuş sürecin devam edebildiğinden emin olmak için kontrol yönetilen süreç tarafından sağlanır.

4.6.1 Sürecin planlanması

Bu uygulamanın amacı sürecin yerine getirilmesi için neye ihtiyaç olduğunun belirlenmesi, sürecin uygulanması için gereken planın hazırlanması, süreç tanımının yapılması, konu hakkında bilgi sahibi olan paydaşlardan gözden geçirme sonrası onay alınması aşamalarının kararlaştırılmasıdır.

4.6.2 Kaynakların sağlanması

Bu uygulamanın amacı, gerektiğinde sürecin tanımlandığı şekilde yerine getirilebilmesi için gerekli olan kaynakların varlığının garantiye alınmasıdır. Kaynaklar, yeterli sermaye, fiziksel beceri, deneyimli çalışan ve uygun araçları içerir.

4.6.3 Sorumlulukların belirlenmesi

Bu uygulamanın amacı, sürecin tamamlanması için gerekli olan sorumlulukların verildiğinden emin olunması ve sürecin yaşam döngüsü boyunca belirlenmiş sonuçların başarıya ulaşmasıdır. Belirlenen insanlar aldıkları sorumluluklar doğrultusunda uzman oldukları alanlara göre görevlendirilir. Sorumluluklar detaylı iş tanımına ya da güncel dokümanlara göre atanabilir.

4.6.4 Çalışanların eğitilmesi

Bu uygulamanın amacı, sürecin başarıya ulaşması için çalışanların yeterli beceriye ve deneyime sahip olduğundan emin olunmasını sağlar. Eğitimler çalışanlara yerine getirecekleri işe göre sağlanır. Genel eğitimler ise bu işlerle etkileşimde olan insanları yönlendirmek için uygulanır. Eğitimler, sürecin anlaşılmasının sağlayarak performansın artmasını sağlar.

4.6.5 Konfigürasyon yönetimi

Bu uygulamanın amacı, yaşam döngüsü boyunca seçilen ürün süreçlerinin güvenilirliğinin sağlanması için bakımının ve kontrolünün yapılmasıdır. Seçilen ürünler konfigürasyon yönetiminin seviyesiyle birlikte, sürecin yerine getirilmesi için tanımlanır. Konfigürasyon yönetimi uygulamaları sürüm kontrolü, kontrol değişimleri, durum tanımlama ve depolama için kullanılan konfigürasyon yönetim araçlarını içerir.

4.6.6 Paydaşların belirlenmesi ve konuya dâhil edilmesi

Bu uygulamanın amacı, sürecin uygulanması sırasında paydaşlardan beklenen katılımın sağlanmasıdır. Konu hakkında bilgi sahibi olan paydaşlar planlama, karar, sorumluluk, gözden geçirme ve problemlerin çözümü gibi aktivitelerde rol sahibi olurlar. Yöneticinin rolü, test olgunluğunun geliştirilebilmesi için görevlerin dağıtılması ve bununla ilgili sorumluluğu üstlenmesidir. Kullanıcının ya da müşterinin rolü ise işbirliği sağlamak, destek vermek ve bazı test aktivitelerine katılmaktır. Müşteriler ya da kullanıcılar kalite aktivitelerine dahil olabilir ve kullanıcının ihtiyaçlarına yönelik test aktiviteleri ile ilgili görevler alabilirler. Müşterinin talebine odaklı destek alınır, ürün risk analizi, kabul testi ve kullanılabilirlik testi gibi aktivitelerde katılım ve fikir ortaklığı önemlidir.

4.6.7 Sürecin izlenmesi ve kontrolü

Bu uygulamanın amacı, sürecin gün gün izlenmesi ve kontrolü üzerinedir. Test sürecindeki uygun görüşlere bakılarak gerektiğinde uygun önlemler alınır. Sürecin izlenmesi ve kontrolü test sürecindeki uygun özelliklerin ölçülmesini ve test sürecinde üretilen işleri içerir.

4.6.8 Üst düzey yönetim ile gözden geçirme yapılması

Bu uygulamanın amacı, üst düzey yönetimden süreçle ilgili uygun görüşün alınmasıdır. Üst düzey yönetim, süreç içinde organizasyon içindeki yönetim seviyelerinin hepsinden fazla yönetim bir sorumluluğa sahiptir. Bu değerlendirme politikası ve gözden geçirme süreç içindeki yönetim içindir ve onlara tümüyle rehberlik eder.

4.7 CMMI Süreç Alanlarının TMMi için Desteklenmesi

TMMi ayrı kullanılabileceği gibi CMMI'ı tamamlayıcı olarak da kullanılabilir. Sonuç olarak birçok durumda TMMi seviyelerinin ona uygun CMMI seviyelerinden ya da daha yüksek CMMI seviyelerinden destek alması gerekebilir. CMMI içinden ayrıntılandırılan süreç alanları ve uygulamalar TMMi da tekrarlanmaz sadece referans gösterilir.

4.8 CMMI Süreç Alanları Doğrulama ve Onaylama

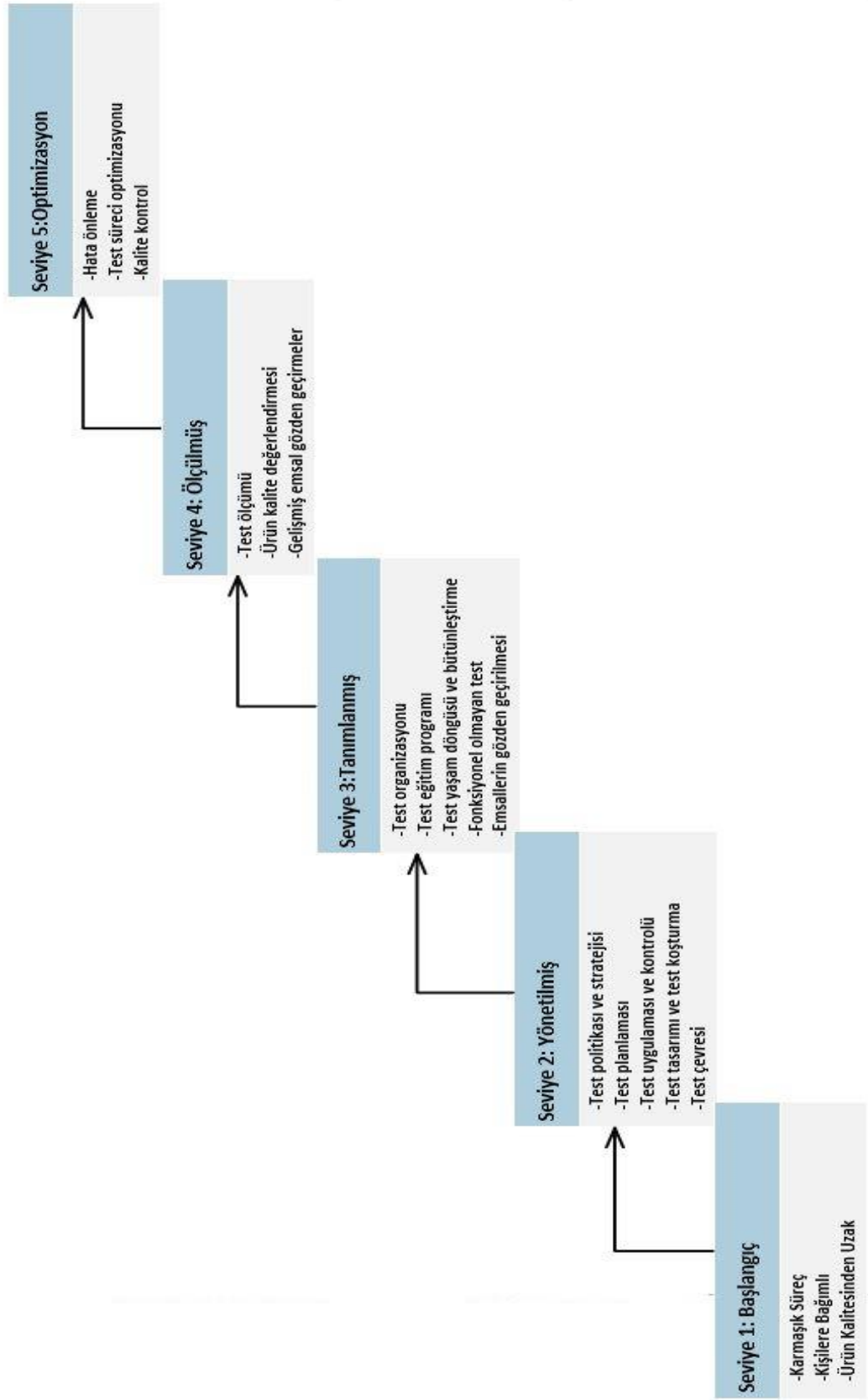
CMMI doğrulama ve onaylamanın test süreç alanları TMMi'in dinamik test süreçlerine destek olarak ya da paralel süreç alanları olarak listelenmemiştir. Bu CMMI süreç alanları için ilişki tamamlayıcıdır. TMMi süreç alanları destek verir ve doğrulama ve onaylama sürecinde ne istendiğini detaylandırır. Uygulama deneyimleri göstermiştir ki TMMi 2 seviyesine uygun bir organizasyon aynı zamanda CMMI doğrulama ve onaylama süreç alanlarına da büyük ölçüde uygunluk sağlar.

4.9 TMMi Olgunluk Seviyeleri

TMMi süreç geliştirmek için ortaya çıkmış bir mimaridir. Test sürecinin organizasyonlar için yönetilmeyen ve dikkate alınmayan bir süreçten yönetilen, tanımlanan ve ölçülen bir süreç haline getirilmesini sağlayan aşamalar ve seviyeler içerir. Her aşamaya yeterli olacak şekilde ulaşılması bir sonraki aşama için temel oluşturur. TMMi'in yapısı kaliteli test sürecinde başarılı olan ve adım adım sonuca ulaşılan, sistematik olarak öğrenilebilecek ve uygulanabilecek zengin test uygulamaları içerir.

TMMi'in test geliştirme sürecinde hiyerarşik olgunluk için tavsiye ettiği beş seviyesi vardır. Organizasyonun her seviye için ulaşmak zorunda olduğu bir olgunluk seviyesi vardır ve bunlar kümelenmiş alanlar şeklinde oluşturulmuş süreçlerdir. Deneyimler göstermiştir ki organizasyonlar belirlenen zaman içinde süreç alanlarının içeriklerine uygun test geliştirme sürecine odaklanarak çaba sarf ederlerse başarıya ulaşmışlardır. Bu alanlar organizasyonun gelişmesine katkı sağlamıştır. Her seviye bir sonraki seviyenin temelini oluşturduğu için bir sonraki seviyeye atlamaya çalışmak verimli olmayacaktır. Bununla birlikte test geliştirme

süreci, organizasyonun iş çevresindeki isteklerine ve projenin olağan isteklerine işaret edecek bir yüksek olgunluk seviyesi içeren bir test süreç alanına odaklanmalıdır.



Şekil 4.1 TMMi olgunluk seviyeleri.

4.9.1 Seviye 1:Başlangıç(Initial)

TMMi 1 seviyesinde test karmaşık, tanımlanmamış ve geliştirme aşamasında yazılımcı tarafından yapılan bir süreçten ibarettir. Organizasyonlar genelde süreci destekleyecek sabit bir çevre sağlamazlar. Böyle organizasyonlarda başarı yeteneğe ve insanların bireysel başarısına bağlıdır. Testler kodlama bittikten sonra plansız bir şekilde oluşturulur. Bu aşamada testin amacı yazılımın büyük problemler oluşmadan çalışmasıdır. Ürünler yeterli kalite ölçülerinden yoksun şekilde ortaya çıkarlar. Yazılım gereksinimleri tam olarak karşılamayabilir, tutarlı değildir ve çok yavaş çalışır. Test için yeterli kaynak, araç ve eğitilmiş çalışan yoktur. TMMi birinci seviye için tanımlanmış test süreç alanı yoktur. Birinci olgunluk seviyesine sahip organizasyonlarda başarılı olan işin tekrarlanmama olasılığı yüksektir buna ek olarak ürün zamanında çıkarılamaz, bütçe aşımı olur ve kalite beklenenin altındadır.

4.9.2 Seviye 2:Yönetilen(Managed)

TMMi 2 seviyesinde test kontrollü bir süreç içerisinde ve kod kısmından bağımsız olarak sürdürülür. Süreç disiplini sağlanır ve stres altında uygulamaların çalıştığından emin olunur. Bununla birlikte testin, kodlamadan sonra yapılması gereken bir aşama olduğu paydaşlar tarafından anlaşılmıştır.

Test sürecinin geliştirilmesi aşamasında tüm program stratejisi kabul edilir. Test planları oluşturulur. Test planının içinde test yaklaşımı tanımlanır ve bu yaklaşım proje risk değerlendirmesine uygun olarak hazırlanır. Risk yönetim teknikleri proje riskini belirlemek için gereksinimlere bakarak tanımlanır. Test planı testte ne istendiğini, ne zaman ve nasıl istendiğini tanımlar. Teslimat paydaşlarla birlikte ve istenenler gözden geçirilerek yapılır. Test izlenir, kontrol edilir ve plana uygun hareket edildiğinden emin olunur ve sapma olma olasılığına karşın hareket edilir. Ürünün ve teslim edilen test hizmeti yönetim tarafından görülebilecek haldedir. Test tasarım teknikleri anlaşmadan bakılarak oluşturulan test senaryolarına bakılarak uygulanır. Test, geliştirme sürecinden sonra tasarım esnasında ya da kodlama esnasında yapılabilir.

TMMi 2 seviyesinde test çoklu seviyelidir. Bu aşamada bileşen, bütünleştirme, sistem ve kabul testi seviyeleri olabilir. Tanımlanan her test seviyesi için belirlenen

stratejiye göre özel test hedefleri konulmuştur. Test ve kod uygulaması süreci ayrı işlemektedir.

TMMi 2 seviyesinde test etmenin asıl amacı belirlenen gereksinimlerin doğrulanmasıdır. Bu seviyede birçok kalite problemi oluşur çünkü test aşaması yazılım yaşam döngüsünün dışında ve geç oluşturulur. Hatalar gereksinimlerden kaynaklanır ve kodun içine yayılır. Formal gözden geçirme yapılmaz ve uygulamaya dayalı test paydaşlar tarafından temel test aktivitesi olarak görülür.

TMMi 2 seviyesinde süreç alanları aşağıdaki gibidir.

- a. Test politikası ve stratejisi
- b. Test planlaması
- c. Testin izlenmesi ve kontrolü
- d. Test tasarımı ve uygulaması
- e. Test ortamı

a) Test politikası ve stratejisi süreç alanı

Test Politikası ve Stratejisi süreç alanının amacı, bir test politikası geliştirmek ve içinde test seviyelerinin açık bir biçimde tanımlandığı bir program stratejisi oluşturmaktır. Test performansını ölçmek için test performans göstergeleri uygulamaya konulur.

Bir organizasyon test politikasını geliştirmek istiyorsa, öncelikle test politikasını açık bir şekilde tanımlamalıdır. Test politikası, organizasyonun tüm hedeflerini, amaçlarını ve teste ilişkin stratejik görüşlerini tanımlar. Test politikasının, organizasyonun tüm iş politikaları ile uyum içinde olması önemlidir. Bir test politikası oluşturulurken, organizasyonun tüm paydaşları ortak bir kanıda buluşmalıdır. Bu ortak kanının tüm organizasyon içinde test (süreç gelişimi) aktiviteleri ile uyum içinde olması gerekir. Bahsi geçen test politikası, test aktivitelerine, hem yeni geliştirme hem de bakım projeleri açısından hitap etmelidir. Bu politikanın içinde test süreci gelişiminin amaçları belirtilmelidir. Ardından, bu amaçlar bir takım anahtar performans göstergelerine dönüştürülecektir. Test politikası ve beraberindeki performans göstergeleri, beklenen ile ulaşılan test seviyesi performanslarıyla bağlantı kurulmasını

sağlar. Performans göstergeleri, paydaşlara, testin ve test süreç gelişiminin değerini göstermelidir. Test performansı, organizasyonun gelişimini ve belirlenen test amaçlarına ulaşip ulaşmadığını sayısal bulgularla gösterecektir.

Test politikası baz alınarak bir test stratejisi belirlenecektir. Test stratejisi, bir organizasyon ya da program (bir ya da daha fazla proje) için genel test gereksinimlerini kapsar. Bu strateji genel ürün risklerine hitap eder ve test politikasına uygun olarak bu risklerin azaltılması için bir süreç sunar. Test stratejisi hazırlığı, bir program ya da organizasyon için geliştirilen ürünleri analiz eden bir genel ürün risk değerlendirmesinin uygulamasıyla başlar.

Test stratejisi, projeler içindeki test aktiviteleri için bir başlangıç noktası ,işlevi görür. Bu projeler, organizasyon ya da program çaplı test stratejileri doğrultusunda düzenlenir. Tipik bir test stratejisi, uygulanacak olan test seviyelerinin açıklamalarını içerecektir, örneğin; bileşen, sistem ve kabul testleri gibi. Her bir test için, asgari düzeyde, amaçlar, sorumluluklar, başlıca görevler, ve başlangıç/bitiş kriterleri tanımlanır. Bir test stratejisi tanımlandığı ve takip edildiği zaman test seviyelerinin üst üste binme olasılığı daha azdır ve bu daha etkili bir test süreci sağlar. Ayrıca, test amaçları ve çeşitli seviyelere yaklaşım sıraya konduğundan süreçte bir boşluk kalma olasılığı daha azdır ki bu da bu süreci daha etkili bir hale getirir.

Test politikası ve test stratejisi değişiminin, genelde bir organizasyonun test süreci geliştikçe ve TMMi'nin seviyelerini tırmandıkça gerekeceği dikkate alınmalıdır.

b) Test planlaması süreç alanı

Test Planlamasının amacı belirlenen riskler ve test stratejisi baz alınarak bir test yaklaşımı tanımlamak, test aktivitelerini uygulamak ve yönetmek için sağlam planlar oluşturmak ve bunları sürdürmektir.

Test görevinin onaylanmasından sonra, test edilecek ürüne, proje organizasyonuna, gereksinimlere ve süreç gelişimine ilişkin etraflıca bir çalışma yürütülür. Test Planlamasının bir parçası olarak, bir ürün risk değerlendirmesinin sonucu ve belirlenen test stratejisi baz alınarak, test

yaklaşımı tanımlanır. Risklerin önceliklerine ve kategorilerine bağlı olarak, ürünün hangi gereksiniminin, ne dereceye kadar, nasıl ve ne zaman test edileceği kararlaştırılır. Amaç, sistemin en yüksek riske sahip olan kısımlarının kapsam alanlarının genişletilmesidir.

Test yaklaşımına bağlı olarak, yapılacak olan iş değerlendirilir ve sonuç olarak, önerilen test yaklaşımına net bir maliyet bilgisi sağlanır. Ürün riskleri, test yaklaşımı ve tahminler, sadece test takımından ziyade daha çok paydaşlara yakın işbirliği içinde olunarak tanımlanır. Test planı test stratejisi ile uyum içinde olacaktır, ya da uyumsuzlukları açıklayacaktır.

Test Planlaması içinde, ortaya çıkacak olan test ürünleri – çıktıları tanımlanır, gereken kaynaklar kararlaştırılır ve altyapıya ilişkin genel bir bakış açısı oluşur. Buna ek olarak da teste ilişkin proje riskleri tanımlanır. Sonuç olarak, test planı, testin neyi, ne zaman, nasıl gerektirdiğini ve bunun kim tarafından yapılacağını tanımlayacaktır.

Son olarak, test planı dokümanı paydaşlar tarafından geliştirilir ve kabul edilir. Test planı, test aktivitelerinin uygulanması ve kontrolü için temel sağlar. Projede, gereksinimlerde, sorumluluklarda, hatalı tahminlerde, düzeltici eylemlerde ve test sürecinde değişikliğe gidilirse, test planının değişim kontrolü süreci kullanılarak gözden geçirilmesi gerekecektir.

c) Testin izlenmesi ve kontrolü süreç alanı

Testin izlenmesinin ve kontrolünün amacı, test ilerleyişi plandan, ürün kalitesi ise beklentilerden önemli derecede saptığı zaman uygun denetleyici eylemlerin yapılması için bir test gelişimi ve ürün kalitesi anlayışını sağlamaktır.

Test gelişimi ve ürünlerin kalitesi hem izlenmeli hem de kontrol edilmelidir. Testin gelişimi, ortaya çıkan ürünlerin, işgücünün, maliyetin, ve programın, test planı içinde belirtilenlerle karşılaştırılması yoluyla denetlenir. Ürünün kalitesi, azaltılmış ürün riskleri, bulunan hata sayısı, görünür hata sayısı gibi göstergeler aracılığıyla denetlenir.

Testin izlenmesi, gerekli bilgileri toplamayı gerektirir. Test raporlarının geçerli olup olmadığına bakmak için işlenmemiş verileri gözden geçirmek ve tanımlanan gelişme ile ürün kalite ölçümlerini hesaplamak buna örnek olarak verilebilir. Test özet raporları, ortak bir test ilerleyişi ve ürün kalitesi anlayışı sağlamak için periyodik olarak yazılmalıdır. Hetzel'in de dediği gibi, *“Test, ürün kalitesinin ölçümüdür.”* Bu yüzden, ürün kalitesi bildirim hakkındaki uygulamalar, bu süreç alanındaki başarı için anahtar niteliğindedir.

Test ilerleyişi plandan ya da ürün kalitesi beklentilerden uzaklaştığı zaman, uygun düzeltici eylemler uygulanmalıdır. Bu eylemler, mevcut plan baz alınarak orijinal planı gözden geçirmeyi ya da yeniden planlamayı gerektirebilir. Orijinal uygulanan planı etkileyen düzeltici eylemler paydaşlar tarafından kabul edilmelidir.

Testin izlenmesinin ve kontrolünün önemli bir parçası test projesi risk yönetimidir. Test projesi risk yönetimi, test planını baltalayan başlıca problemleri olabildiğince erken tanımlamak ve çözmek için uygulanır. Proje risk yönetimini uygularken, testin sorumluluk sınırlarını aşan problemleri tanımlamak da önemlidir. Örneğin, kurumsal bütçe kesintileri, geliştirilmiş iş ürünlerinin gecikmesi ya da değiştirilmiş/eklenmiş fonksiyonellik test sürecini büyük ölçüde etkileyebilir. Daha önceden test planında belirlenen test proje risklerine dayandırılarak, bu riskler denetlenir ve kontrol edilir. Ayrıca, düzeltici eylemler gerektiğinde başlatılır.

Testin izlenmesi ve kontrolü süreç alanı, tahminlere, sorumluluklara, planlara ve beklentilere karşı olarak test ilerleyişini ve ürün kalitesini denetlemeyi, bu ilerleyişi ve ürün kalitesi hakkında paydaşlara rapor sağlamayı ve sonuna kadar düzeltici eylemleri yönetmeyi içerir.

d) Test tasarımı ve uygulanması süreç alanı

Test tasarımı ve uygulamasının amacı, test tasarım tekniklerini kullanarak, yapılandırılmış bir test uygulama süreci işleterek ve kapanışa kasar test hatalarını yöneterek, test tasarım ve uygulaması süresince test süreç kabiliyetini geliştirmektir.

Yapılandırılmış test, mümkünse araçlarla desteklenerek, test tasarım tekniklerinin uygulamaya konulması manasına gelir. Test tasarım teknikleri, test koşullarının ve test tasarım dosyalarının, gereksinimlerden ve tasarım özelliklerinden türetilmesi ve seçilmesi için kullanılır. Test durumları ve senaryoları, test dokümanları içinde ayrıntılı olarak yazılır. Bir test prosedürü, girdi değerlerinin tanımından, uygulama önkoşullarından, beklenen sonuçlardan ve uygulama sonrası koşullardan oluşur. Daha sonraki aşamada, uygulamayla ilgili daha fazla bilgi ortaya çıktıkça, test senaryoları, test prosedürlerine dönüştürülür. Aynı zamanda manuel test senaryoları dosyası olarak da bilinen bir test prosedüründe, test eylemleri ve kontrolleri yürütülebilir bir sıraya göre dizilir. Test prosedürünü yürütmek için gerekli olan test bilgileri oluşturulur. Testler, bu prosedürler kullanılarak uygulanır.

Test tasarım ve uygulama aktiviteleri, test planında tanımlanan test yaklaşımını takip eder. Uygulanan özel test tasarım teknikleri (kara kutu, beyaz kutu ya da deneyime dayalı testler gibi), test planlama sürecinde saptanan ürün risklerinin türüne ve seviyesine bağlıdır.

Test uygulama sürecinde, hatalar bulunur ve hata raporları yazılır. Bunlar, bir hata yönetim sistemi kullanılarak kaydedilir ve her bir belirlenmiş protokol için paydaşlar ile iletişime geçilir. Bir ana hata sınıflandırma şeması oluşturulur ve kapanışa kadar her bir hatayı yönetmeyi de içeren hata yaşam döngüsü sürecini idare etmek için bir prosedür devreye sokulur.

e) Test ortamı süreç alanı

Test ortamı süreç alanının amacı, testleri yönetilebilir ve tekrar edilebilir bir biçimde uygulamanın mümkün olduğu test bilgilerini içeren yeterli bir çevre oluşturmak ve bunu sürdürmektir.

Yönetilen ve kontrol edilen bir test çevresi, her test için vazgeçilmezdir. Gerçek hayat koşullarına mümkün olduğunca yakın olan koşullar altında test sonuçları elde etmek de gereklidir. Bu, özellikle sistem ve kabul testi seviyesi gibi daha yüksek seviye testleri için geçerlidir. Buna ek olarak, her test seviyesinde, test sonuçlarının tekrar gerçekleştirilebilirliği, test çevresindeki istenmeyen ya da bilinmeyen değişiklikler tarafından tehlikeye atılmamalıdır.

Test çevresi gereksinimlerinin oluşturulması projenin erken evrelerinde uygulanır. Gereksinim dokümanı, doğruluğundan, uygunluğundan, elverişliliğinden ve gerçek ortam çevresinin kesinliğinden emin olmak için gözden geçirilir. Önceden hazırlanan gereksinim dokümanı, simülatörler ya da sürücüler gibi bileşenleri ve gerekli test çevresini elde etmek ve/veya geliştirmek için daha fazla zaman sağlanması için avantaj sağlar. Gerekli olan çevrenin türü, test edilecek olan ürüne, ve kullanılacak olan test tiplerine, metotlarına, ve tekniklerine bağlı olacaktır.

Test çevresinin kullanılabilirliği, hitap edilmesi gereken bir çok konuyu kapsar. Her bir test takımı ya da her test seviyesi için ayrı bir test çevresinin olması çok pahalı olabilir. Belki de testi uygulayanların ve bunu geliştirenlerin aynı çevreyi paylaşmaları mümkündür. Eğer öyleyse, test ve geliştirme aktiviteleri aynı çevrede yapıldığından ve süreci negatif bir biçimde etkileyebileceğinden dolayı katı yönetim ve kontrol gereklidir. Eğer kötü bir biçimde yönetilirse, bu durum, çelişki yaratan şüpheler ile insanların aktivitelerine başladıkları zaman çevreyi bilinmeyen ya da istenmeyen bir durumda bulmaları arasında değişiklik gösteren birçok probleme yol açabilir.

Son olarak, test çevresi yönetimi, aynı zamanda, sisteme giriş detaylarını sağlayarak test çevresine erişimi ve test verilerini yönetmeyi, düzenleme yönetimini sağlamayı ve uygulamayı, ve test işleyişi sırasında ilerlemeyi etkileyen konular üzerinde teknik desteği de sağlar.

Test Çevresi süreç alanının bir parçası olarak, genel test verilerine ilişkin gereksinimlerden ve bu verilerin oluşturulması ve yönetiminden de bahsedilir. Spesifik veriler, test tasarımı ve analiz aktiviteleri sırasında tanımlandığı için daha genel test verileri genelde ayrı bir aktivite olarak tanımlanır ve oluşturulur. Genel test verileri, bu testi uygulayan birçok kişi tarafından yeniden kullanılır ve sistemin işlemesi için gereken tüm verileri sağlar. Bunlar ileri düzey verilerden oluşur. Bazen zamanlama gereksinimleri bu aktiviteyi etkiler.

4.9.3 Seviye 3: Tanımlanan (Defined)

TMMi 3 seviyesinde test kodlamadan sonra yapılacak bir aktivite olarak görülmez. Tamamen yazılım geliştirme yaşam döngüsüyle bütünleşmiş şekilde yürütülür. Test planlaması projenin ilk aşamalarında gereksinimlerin yazıldığı safhada yapılır ve doküman olarak hazırlanır. Test planının geliştirilmesi, amaçları, sorumluluklar TMMi 2 seviyesinde belirlenmiştir. Organizasyonun standart test süreci üçüncü olgunluk seviyesi için temel oluşturur. Bir test organizasyonu ve çalışma programı zaten vardır ve test işine profesyonel olarak bakılır. Test geliştirme süreci tamamen endüstriyelmiş ve kabul edilen uygulamalara sahiptir.

Üçüncü seviyedeki organizasyonlar kalite kontrolündeki gözden geçirmenin öneminin farkındadır. Formal gözden geçirme tam olarak dinamik test sürecinde yerine getirilmemektedir. Gözden geçirme yaşam döngüsünün dışında yapılır. Profesyonel test işinde gereksinimler de gözden geçirilmelidir. TMMi 2 seviyesinde test sadece fonksiyonellik testine odaklanır, kullanılabilirlik, güvenilirlik gibi kalite odaklı fonksiyonel olmayan testler için yapılan test tasarımı ve test teknikleri üçüncü seviye içinde kapsanır.

TMMi 2 ve 3 olgunluk seviyeleri arasındaki kritik fark standartların kapsamı, süreç tanımları ve prosedürlerdir. İkinci seviyede belirli projelerde her özel durum için çok fazla fark olabilir. Üçüncü seviyede ise bunlar organizasyon tarafından belirlenen projeye uyarlanır ve bu nedenle uyarlanan ana hatlara izin verilen değişiklikler dışında tutarlılık vardır. Başka bir kritik fark ise üçüncü seviyede süreçler ikinci seviyeye göre çok daha ayrıntılı tanımlanmıştır. Sonuç olarak üçüncü seviye için organizasyon ikinci seviye alanlarının üstünden geçmek zorundadır.

TMMi 3 seviyesinde süreç alanları aşağıdaki gibidir.

- a. Test organizasyonu
- b. Test eğitim programı
- c. Test yaşam döngüsü ve bütünleştirme
- d. Fonksiyonel olmayan test
- e. Emsallerin gözden geçirilmesi

a) Test organizasyonu süreç alanı

Test Organizasyonu süreç alanının amacı, testten sorumlu olan bir grup insanı saptamak ve onları organize etmektir. Teste ek olarak, test grubu aynı zamanda, organizasyonun güncel test sürecinin ve test süreç değerlendirmesinin güçlü ve zayıf yanlarının ayrıntılı bir anlayışını baz alarak, bu konular üzerinde geliştirmeler gerçekleştirir.

Bir test organizasyonu oluşturmak, daha iyi bir test ve yüksek kaliteli yazılım sorumluluğunu kapsar. Süreci başlatmak amacıyla, üst yönetim kurulunun, bir test grubu kurmak ve kaynakları gruba teslim etmek için alınan kararı desteklemesi gerekmektedir. Bu, aynı zamanda, test ve kalite konularına ilişkin alanlarda liderlik sergilemeyi gerektirir. Böyle bir grubun personelleri, test uzmanları olarak adlandırılırlar. Bir test organizasyonu (grubu), yapılandırılmış testte yüksek standartlara ulaşmak için, uzmanlar, test imkânları ve proje ile alakalı test aktiviteleri arasındaki etkili ilişkilerin bir temsilidir. Test grubundan iş, gelişme, ve kalite güvencesine kadar olan iyi tanımlanmış iletişim bağlantıları oluşturulur.

Bağımsız bir test grubuna sahip olmak bir organizasyon için önemlidir. Grubun, organizasyon hiyerarşisi içerisinde biçimlendirilmiş bir pozisyona sahip olması gerekir. “Bağımsız” terimi genel manada kullanılır, ancak; her organizasyon bu konuda kendi yorumlamasını ve bağımsızlığını kendilerine uygun olan seviyede geliştirmek zorundadır. Örneğin; bir test organizasyonu, bir test kaynak havuzuna sahip olan bir test yetkinlik merkezi organize edebilir. Bu tip bir organizasyonda, grup üyeleri test işini yaptıkları organizasyonlar boyunca ya da piyasaya sürülmeden önceki onayı yerine getiren bağımsız bir grup olarak projeler için görevlendirilirler. TMMi açısından, test organizasyonundaki bağımsızlık, testi uygulayan kişilerin mühendislik uzmanları olarak tanınması anlamına gelir. Bu kişiler geliştirici olarak düşünülmez ve en önemlisi geliştirme yönetiminden bağımsız olarak, yönetime rapor verirler. Test uzmanlarının, gelişim organizasyonun baskısıyla engellenmeden, objektif ve tarafsız olmalarına izin verilir.

Test bir uzmanlık alanı olarak algılanır ve test grubu bir gereklilik olarak görülür. Test mühendisliğine, yönetimine ve uygulama alanına ilişkin detaylı ve belli bir alanda uzmanlaşmış bilgi ve yetenekler, test grubunda görevlendirilmiş

motive edilmiş bireylerin karakteristik özellikleridir. İşlevler ve test kariyer yolları, bir test eğitim programı tarafından tanımlanır ve desteklenir. Bu grup, iyi bir test uzmanı olmak için gerekli kabiliyetlere ve motivasyona sahip olan kişilerden oluşturulur. Bu insanlar spesifik bir test işlevi için görevlendirilirler ve ürün kalite hedefleri hakkında farkındalık oluşturmaya ve bu hedefleri başarmaya kendilerini adanlar. Kalite özelliklerini değerlendirip ölçerler ve sistemin müşteri gereksinimlerini karşıladığının garantisini sağlama sorumluluğuna sahiptirler. Aynı zamanda, diğer grup üyelerinin test aktiviteleri, rolleri ve sorumlulukları belirlenir.

Her ne kadar TMMi ikinci seviye gelişimi bazen doğaçlama bir proje olsa da, test organizasyonu içinde iyi bir şekilde organize edilmiş ve yapılandırılmıştır. Diğer disiplinlerin koordine edilmesini de içeren test süreç gelişim aktivitelerini kolaylaştırma ve yönetme sorumluluğu, genellikle yönetim kurulu tarafından desteklenen bir test yöneticisine verilmiştir. Genellikle Test Süreç Grubu diye adlandırılan bir gelişim grubu, çoktan oluşturulmuş ve personeli sağlanmıştır. Süreç gelişimi için adaylar, ölçümleri, öğrenilen dersleri, ve değerlendirme sonuçlarını da içeren çeşitli kaynaklardan elde edilir. Dikkatli planlama, organizasyon boyunca test süreç gelişimi çabalarının yeterince iyi yönetildiğinin ve uygulandığının garantisinin verilmesini gerektirir. Test süreç gelişiminin planlanması, ortaya bununla ilgili bir plan çıkarır. Bu plan, değerlendirme planlamasını, süreç faaliyet planlamasını, pilot uygulama planlamasını kapsar. İlk test gelişimi uygulandığı zaman, uygulama planı kullanılır. Bu plan, organizasyon boyunca gelişimin nasıl ve ne zaman uygulanacağını tanımlar.

b) Test Eğitim Programı süreç alanı

Test eğitim programının amacı, test görevlerinin ve rollerinin etkili ve faydalı bir şekilde uygulanabilmesi için insanların bilgilerinin ve yeteneklerinin gelişimini kolaylaştıran bir eğitim programı geliştirmektir.

Test Eğitim Programı süreç alanı, organizasyonun stratejik iş hedeflerini desteklemek için ve projelerde yaygın olan eğitim ihtiyacını karşılamak için bir eğitim programı içerir. Ayrı projeler tarafından tanımlanan özel eğitim ihtiyaçları, proje seviyesinde idare edilir. Test Eğitim Programı süreç alanı, Test Organizasyonu süreç alanı ile yakından ilgili ve bağlantılıdır. Test Eğitim

Programı süreç alanı ana hedeflerinden biri, test uzmanlarını ve diğer dahil olan paydaşları eğiterek test organizasyonunu desteklemektir. Bir kalite eğitim programı, teste dahil olanların test kabiliyetlerini geliştirmesinin ve alan bilgileri ile teste ilişkin diğer bilgilerini güncellemelerinin garantisini vermektedir. Eğitim programı, kendini işine adanmış bir eğitim grubu tarafından organize edilebilir ve yönetilebilir.

Bir test eğitim programı oluşturmak, yüksek kalitedeki test ekibini desteklemek ve devam etmekte olan test süreç gelişimine katkıda bulunmak için yönetim tarafından eklenen bir sorumluluktur. Testte, çeşitli kabiliyetler gereklidir. Ana kategoriler, test prensipleri, test teknikleri, test yönetimi, test araçları, alan bilgisi, sistem mühendisliği, yazılım geliştirme ve çevreyle uyum becerisidir. Birçok eğitim modülünden oluşan test eğitim programı, bu kategorilere hitap etmek için geliştirilmiştir. Yüksek TMMi seviyelerinde, daha da gelişmiş diğer eğitim kategorileri önemli hale gelecektir. Bazı yetenekler resmi eğitim gerektirirken, bazıları ise etkili ve verimli bir şekilde gayri resmi araçlar vasıtasıyla elde edilebilir.

“Eğitim” terimi, tüm bu öğrenim seçeneklerini kapsamak için, bu süreç alanı boyunca kullanılır. Test eğitim programı, test işleyişi ile rollerine bağlıdır ve test kariyer yollarını kolaylaştıracaktır. Eğitim programının uygulamak, teste dahil olan tüm insanlar için uygun bilgi ve kabiliyet seviyesini garanti etmektedir. Test Eğitim Programının uygulanması, öncelikle organizasyon içinde test eğitim ihtiyaçlarının tanımlanmasını, özel eğitim modüllerinin geliştirilmesi ve elde edilmesini, tanımlanan ihtiyaçlara hitap etmek için eğitimin yürütülmesini ve son olarak, eğitim programının ne kadar etkili olduğunun değerlendirilmesini içerir.

c) Test yaşam döngüsü ve entegrasyon süreç alanı

Test yaşam döngüsü ve entegrasyon süreç alanının amacı, organizasyon içinde kullanılan bir takım test süreci varlıkları ile iş çevresi standartlarını kurmak ve sürdürmektir. Aynı zamanda, geliştirme yaşam döngüsü ile test yaşam döngüsü bütünleştirmek ve senkronize etmek de bu bölümün amaçlarına dahildir. Bütünleştirilmiş yaşam döngüsü, bir proje içine testin erkenden dahil olmasını garantiler. Test yaşam döngüsü ve bütünleştirmenin bir diğer amacı da, tanımlanan riskler ve belirlenen test stratejisi temel

alınarak, birden çok test seviyesi üzerine tutarlı bir test yaklaşımı belirlemek ve tanımlanan test yaşam döngüsüne dayanarak her şeyi kapsayan bir test planı sağlamaktır.

Test organizasyonunun önemli sorumluluklarından biri de, organizasyonun test politikası ve hedefleri doğrultusunda, standart bir test süreci belirlemek, belgelemek ve sürdürmektir. Organizasyon içinde test süreç varlıkları, organizasyon boyunca tutarlı bir test performans sürecini mümkün kılar ve organizasyona uzun vadeli faydalar için bir temel sağlar. Organizasyonun test süreç varlıkları kütüphanesi, insanlar ve organizasyonun projeleri tarafından kullanılmak üzere muhafaza edilmiş parçaların koleksiyonudur. Bu koleksiyon, teslim edilebilir destekleyici taslaklar ve rehberler içeren test süreçlerinin ve test yaşam döngüsü modellerinin tanımları, destekleyici test araçları, süreci uyarlayan rehberler, ve bir test süreci veri tabanı içerir. Organizasyonun test süreç varlıkları kütüphanesi, organizasyon süresince öğrenilen uygulamaları ve dersleri paylaşarak organizasyon içinde öğrenmeyi ve süreç gelişimini destekler. Standart test yaşam döngüsü modelleri, birçok değişik test seviyesi için ana aşamaları, aktiviteleri tanımlamaktadır. Ardından, test aktiviteleri, bu modellere göre projeler içinde uygulanacaktır. Standartlar ve rehberler teste ilişkin ürünler için geliştirilir. Yaşam döngüsü bütünleştirilmesi öyle bir şekilde yapılır ki, testin projelere erkenden dahil olması garantilenir. Örneğin; test planlaması gereksinimlerin belirtilmesi aşamasında başlar, bütünleştirme ve bileşen test planlaması detaylı tasarım evresinde başlatılır. Testi yapan kişiler, sınanabilirliği belirlemek için testin temel dokümanlarını gözden geçireceklerdir ve test yaklaşımı, gelişim planlamasını etkileyebilir. Organizasyonun bir takım standart test süreçleri, belirlenmiş özel süreçler yaratmak için projeler tarafından uygun biçime getirilebilir. İş çevresi standartları, proje iş çevrelerinin oluşturulmasına rehberlik etmek için kullanılır.

TMMi 3 seviyesinde test yönetimi, test görevlerinin koordinasyonunu, sorumluluklarını ve çoklu test seviyelerinde test yaklaşımını işaret eden ana test planı ile ilgilidir. Bu gereksiz fazlalıkları ya da farklı test seviyeleri arasındaki ihmal edilen testlerin olmasını önler ve tüm test seviyelerindeki güvenilirliğin ve kalitenin artmasını sağlar. Proje test planı sonucundan alınan bilgi, bireysel test seviyeleri için detaylandırılmış test planlarının yazılmasını sağlar. Ana test planı, projenin bir kısmı için yapılan test stratejisi

uygulamalarını anlatır. Bu plan belirli seviyelerin uygulamasını ve bu seviyeler arasındaki ilişkileri içerir. Ana test planı, test politikası ve stratejisi ile uyumlu olmalıdır ve özel alanlarda istisnaları ve sapmaları açıklamalıdır. Ana test planı proje planını tamamlayıcıdır ve çoklu test seviyelerindeki tüm test planı ve test yönetim dokümanlarını karşılar. Küçük projelerde ana test planı ve seviyeler içindeki test planı genellikle sadece bir dokümanda birleştirilir.

d) Fonksiyonel olmayan test süreç alanı

Fonksiyonel olmayan test süreç alanının amacı test planlaması, tasarımı ve koşturulması esnasında fonksiyonel olmayan testi içeren test süreci geliştirmektir. Bu, fonksiyonel olmayan ürün risklerinin belirlendiği, fonksiyonel olmayan testin anlaşma içinde saptandığı ve yapısal test koşturma sürecinde fonksiyonel olmayan teste odaklanılan bir test yaklaşımı ile olur.

Ürünlerin kalitesi, paydaşların isteklerinin karşılanması ile ilgilidir. Bu istekler iyi tanımlanmış fonksiyonel olan ya da olmayan gereksinimler şeklinde tanımlanmak zorundadır. Fonksiyonel olmayan gereksinimler genellikle müşteri memnuniyeti açısından daha önemlidir. Bu süreç alanı fonksiyonel olmayan testin kapasitesini arttırmaya yöneliktir. Yazılım ürünlerinin ya da sistemlerinin kalitesini tanımlamaya yönelik temel alınabilecek fonksiyonel olmayan test için nitelikler vardır. Bu kalite nitelikleri, fonksiyonel olmayan test teknikleri kullanılarak belirlenir. Değişik test tekniklerinin uygulanması testçinin uygulama şekline göre, alan bilgisine göre ve tanımlanan niteliklere göre çeşitlenir.

Test yaklaşımının, fonksiyonel olmayan test ürün risklerinin değerlendirme sonuçlarına göre tanımlanması gerekmektedir. Seviyeye ve fonksiyonel olmayan risk tiplerine göre ürünün hangi gereksinimlerinin nasıl ve hangi seviyede test edileceğine karar verilir. Fonksiyonel olmayan ürün riskleri ve test yaklaşımı test uzmanları ve paydaşların işbirliği ile tanımlanır. Testçiler bu kararı tek başına vermemelidir.

Fonksiyonel olmayan test teknikleri mümkünse test araçlarından destek alınarak uygulanır. Test teknikleri türetilmeli, fonksiyonel olmayan test koşulları seçilmeli ve fonksiyonel olmayan gereksinimlerden ve tasarım anlaşmalarından test durumları oluşturulmalıdır. Test durumları manuel test prosedürlerine çevrilmeli ya da test senaryoları ile otomatize edilmelidir. Test

koşturma aşamasında fonksiyonel olmayan test koşturulur ve olay raporu hazırlanır.

e) Emsallerin gözden geçirilmesi süreç alanı

Emsallerin gözden geçirilmesi süreç alanının amacı ürünlerin gereksinimlerle uyduğunu ve seçilen ürünler için hataların erken ve etkili biçimde yok edilmesinin doğrulanmasıdır. En önemli sonuç daha anlaşılabilir bir ürün geliştirmek ve hataların önlenabilir olmasıdır.

Gözden geçirmeler ürünlerin hatalarının bulunması ve değişime ihtiyaç olan alanların tanımlanması için emsalleri kullanılarak sınanmasını sağlar. Gözden geçirmeler 2-7 kişi arasında olacak şekilde bir grup mühendis ile idare edilir. Gözden geçirilecek olan ürünün gereksinim sözleşmesi, tasarım dokümanı, kaynak kodu, test tasarımı ve kullanıcı dokümanı bulunmalıdır. Uygulamada, gözden geçirenlerin seçilmesi için birçok farklı yöntem vardır. Gözden geçiren kişiler aşağıdaki özelliklere göre belirlenmelidir.

- Gözden geçirme konusunda uzmanlaşmış,
- Aynı projedeki kişilerden seçilmiş,
- Özel bilgi potansiyellerinden dolayı yönetici tarafından davet edilmiş,
- Ürünle ilgili özel ilgisi olan iş temsilcilerinden seçilmiş.

Farklı tiplerde gözden geçirmeler tanımlanır ve her birinin kendine özgü amacı vardır. Formal olmayan gözden geçirmelere ek olarak örnekler, teknik gözden geçirmeler ve denetlemeler gibi daha formal gözden geçirme tipleri de kullanılır. Örnekler de, yazılımcı bir grup insana düşündüğü süreç için ve doküman için rehberlik eder. Böylece herkes dokümanı aynı şekilde anlamış olur ve içerik üzerinde ve yapılacak değişikliklerle ilgili fikir birliğine ulaşılır. Teknik gözden geçirmede ise bireysel hazırlıktan sonra içerik ve teknik yaklaşımın kullanılabilirliği açısından grup kendi arasında tartışır. Denetleme ise en formal gözden geçirme tipidir, kaynaklar ve standartlar kullanılarak hatalar bireysel olarak ve grup halinde dokümandan kontrol edilir.

4.9.4 Seviye 4 : Ölçülen (Management and Measurement)

TMMi 2 ve 3 seviyesinde hedefe ulaşmak, testin eksiksiz yapılabilmesi için gerekli olan teknik altyapının, yönetsel ve personel altyapısının bulunduğunu ve test geliştirme sürecinin başarıyla tamamlanabileceğini gösterir. Bu altyapı, testin ilerleyen zamanda daha gelişmiş ve başarılı hale gelmesine yardımcı olur. TMMi 4 seviyesindeki organizasyonlarda test eksiksiz tanımlanmış, temeli sağlam olan ve ölçülebilir bir süreçtir. Test, ürün kontrolleri ve ürünlerle ilgili olan tüm yaşam döngüsü aktivitelerini kapsar.

Büyük organizasyonlarda test sürecinin kalitesini ve verimliliği değerlendirmek, gelişmeleri denetlemek için test ölçüm programları kullanılır. Test ölçüm programı aynı zamanda test performans ve maliyetiyle ilgili öngörülerde bulunur.

Ürün kalitesine uymakla birlikte, ölçüm programının varlığı organizasyonun kalite gereksinimlerini, kalite metrik ve niteliklerini kullanarak ürün kalitesi değerlendirme sürecini yerine getirmesine yardımcı olur. Ürünler, güvenilirlik, kullanılabilirlik ve sürdürülebilirlik gibi kalite nitelikleri için nicel kriterlerin değerlendirilmesi ile geliştirilir. Ürün kalitesi nicel terimlerle anlaşılır ve yaşam döngüsü içerisinde baştan sona kadar tanımlanan amaçlara yönelik yapılandırılır.

Gözden geçirme ve denetimler test sürecinin bir parçası olarak hesaba katılır ve yaşam döngüsünün erken evresinde ürün kalitesi ölçülür. Hata belirleme tekniği gibi emsal gözden geçirmeler, ürün kalite ölçüm tekniğine ve sırasıyla ürün kalite değerlendirmesi test süreç alanına çevrilir.

TMMi 4 seviyesi aynı zamanda testin daha etkili ve verimli yapılmasını hedefleyerek emsal gözden geçirmeler ve dinamik test arasındaki yaklaşımların düzenlenmesini, gözden geçirmelerin kullanılmasını ve test yaklaşımındaki verilerin en uygun haline getirilme aşamalarını kapsar. Gözden geçirmeler tam olarak dinamik test sürecine, test planına ve test yaklaşımına entegre edilmiştir.

TMMi 4 seviyesindeki süreç alanları aşağıdaki gibidir.

- a. Test ölçümü
- b. Ürün kalite değerlendirmesi
- c. Gelişmiş emsal gözden geçirmeler

a) Test ölçümü süreç alanı

Test ölçümünün amacı organizasyona destek verecek olan test sürecinin etkinliğini arttıracak ölçütler toplamak, tanımlamak analiz etmek ve uygulamaktır. Test personelinin verimliliğini arttırmak ürün kalitesini sonuçlandırmak da amaçları arasındadır. Test organizasyonu yönetimin isteklerine destek verecek kapasite ölçütlerinin geliştirilmesi ve sürdürülmesini sağlar.

TMMi 2 ve 3 seviyesinin amaçlarına ulaşmak, test süresince teknik olarak, yönetim açısından ve çalışan altyapısı açısından önemli bir yarar sağlar ve test süreci geliştirilmesine destek verir. Bir organizasyondaki bu altyapı ile formal test ölçüt programları büyümeyi ve başarıyı teşvik etmek için kurulabilir.

Test ölçümü, ürünlerde ve test sürecinde sırasıyla anlaşılması, sürecin etkinliğine destek sağlaması için tanımlamanın, veri toplamanın ve verilerin analiz edilmesinin kapsandığı sürekli olarak devam eden bir süreçtir. Veri toplanması, depolaması, yeniden kullanımı ve iletişimi için olan ölçütler ve analiz metotları bir test ölçüm programında uygulamalara destek vermesi için özelleştirilirler. Bir test ölçüm programının iki ana alanı vardır. Ürün kalitesinin geliştirilmesine ve test sürecine destek verir ve süreç geliştirmesinde yardımcı olur.

Sırasıyla başarılı olabilmesi için test ölçüm programının iş yeri amaçlarıyla, test politikasıyla ve test stratejisiyle bağlantılı olması gerekir. İş yeri amaçları, test ölçüm hedeflerinin ve metriklerinin belirlenmesi için başlangıç noktasıdır. İş yerinin amaçlarından organizasyonların standart test süreçleri için hedefler türetilir. Eğer başarılı bir şekilde uygulanırsa test ölçüm programı test kültürüyle bütünleşen bir parça haline gelecektir ve ölçütler tüm test grupları ve takımları için kabul edilir hale gelecektir. Ölçütler gelecekteki projeler için organizasyonun plan geliştirmesine yardımcı olacaktır. Test ölçütlerine örnek olarak test maliyeti, test durumlarının sayısı, hata verileri ve ürün ölçütleri gösterilebilir. Test ölçümleri süreç alanı aşağıdakileri içerir.

- Tanımlanan bilgi gereksinimi ve iş gücü amaçlarıyla birlikte test ölçütlerinin amaçlarının da belirtilerek birbirleriyle hizalanması,

- Ölçütlerin, analizlerin ve onaylama tekniklerinin belirtilmesinin yanı sıra veri toplama, depolama, yeniden kullanma ve geri besleme mekanizmalarının da belirtilmesi,
- Verinin toplanmasının, depolanmasının, analizinin ve kaydının yapılması,
- Kararların verilebilmesi ve doğru şekilde hareket edilebilmesi için kullanılacak olan sonuçların sağlanması.

TMMi'nin düşük seviyelerinde organizasyonun test süreciyle ilgili veri toplaması gerektiği düşünülür. Ancak düşük TMMi seviyesindeki bir organizasyonda basit bir hata deposu kaynağından hatayla ilgili ölçümleri birleştirmeye başlaması önerilir. TMMi 4 seviyesine gelmek istediği zaman organizasyon daha yüksek seviyedeki test süreci olgunluğunu yerine getirebilmek için ek ölçümlere ihtiyacı olduğunu görecektir. Organizasyonun TMMi 4 seviyesine ulaşması için formal test ölçüm programlarını kullanması gerekmektedir. Birçok organizasyon için genel ölçüm programına destek verecek bir test ölçüm programı pratik olacaktır.

b) Ürün kalite değerlendirmesi süreç alanı

Ürün kalite değerlendirmesinin amacı ürün kalitesiyle ilgili nicel bir anlaşılabilirlik sağlanması ve böylece projelerin ürün kalitesine ulaşma hedeflerine destek sağlamaktır.

Ürün Kalite Değerlendirme, projenin nicel ürün kalite hedef tanımlarını ve bu başarı hedeflerindeki eylem planlarını içerir. Aynı zamanda ürün kalite değerlendirmesi için olan kalite ölçümlerinin tanımını da içerir. Planlanan ürünler, aktiviteler ve ürün kalitesi durumları gözlenir ve mümkün olduğu kadar düzenlenir. Genel hedef, son kullanıcılar için kaliteli ürünler ortaya çıkarmak ve müşterilerin gereksinimlerini karşılayabilmektir.

Bu işlem alanında nicel hedefler, organizasyonun, müşterinin ve son kullanıcının gereksinimlerini karşılayan ürünlerin ortaya çıkması için vardır. Bu hedeflerin başarıyla sonuçlanması için , organizasyonun süreci kendi tanımlanmış test süreçlerine uydurması gerekir ve organizasyon bu süreçte kalite hedeflerine erişebilmek için stratejiler ve planlar geliştirir.

c) Gelişmiş gözden geçirmeler süreç alanı

Gelişmiş gözden geçirmeler süreç alanının amacı; TMMi üçüncü seviyedeki süreç alanındaki “Emsal Gözden Geçirmeler” süreç alanının üzerine inşa edilerek ve dinamik test ile emsal gözden geçirmeleri referans alarak test stratejileri ve test yaklaşımları geliştirmek ve ürün kalitesinin ölçümünü yazılım yaşam döngüsünün erken evrelerinde gerçekleştirebilmektir.

Test sürecinin düzgün ve açık bir şekilde tamamlanması, bu sürecin tüm yazılım yaşam döngüsü aktivitelerinin statik ve dinamik olarak kapsadığı yazılım ürünün ve ilgili iş ürünlerinin planlanması, hazırlanması ve değerlendirilmesinin tam anlamıyla yapıldığı anlamına gelir. Teste olan bu bakış açısı, testin statik ve dinamik analizleri kapsayan, hem de doğrulayan ve gerçekleyen evrimsel test modelinden ortaya çıkmıştır. Bu bakış açısında, gözden geçirmeler testin önemli bir parçasıdır ve doğrulama, geçirme ya da statik analiz tekniği olarak ortaya çıkabilir. TMMi dördüncü seviyede bu bakış açısı statik ve dinamik test ile desteklenir. TMMi üçüncü seviyedeki emsal gözden geçirmeler süreç alanında emsal gözden geçirmelerin yapıldığı ancak dinamik test ile bütünleştirilmediği söylenebilir.

Emsal gözden geçirmeler, ürün ortaya çıkmadan önce hataların ve ürün risklerinin tanımlanması için etkili bir yoldur. Emsal gözden geçirmeler ve dinamik test koordine edildiği zaman erken gözden geçirme sonuçları ve veriler test yaklaşımına yön vermek için kullanılabilir. Test süreci hata kümeleme olarak bilinen test prensibi üzerine inşa edilirse gözden geçirmeler sırasında hataların tipi ve miktarının bulunması daha etkili test yaklaşımlarının geliştirilmesine yardım edecektir. Projenin aşamalarında test yaklaşımı tekrar değerlendirilir ve güncellenir. Gözden geçirme verileri bu güncellemeler sırasında ele alınması gereken verilerden biridir.

4.9.5 Seviye 5: İyileştirilmiş(Optimization)

Daha önce TMMi 1 seviyesinden 4'e kadar belirlenen test geliştirme süreci hedeflerine ulaşılması, tamamen tanımlanmış ve ölçülmüş bir test süreci için organizasyonun altyapısının uygun olduğunu gösterir. TMMi 5 olgunluk seviyesinde olan bir organizasyon, istatistiksel olarak kontrol edilen süreçlerin nitel olarak anlaşılması ilkesine dayalı ve süreçleri sürekli geliştirmeye yatkındır. Test

süreç performansının geliştirilmesi için teknolojik gelişmeleri uygulamak ve artarak yenilenen bir test süreç alanı yaratmak gerekir. Test metot ve teknikleri en uygun hale getirilir ve düzene devamlı odaklanılarak süreç geliştirilir. TMMi için tanımlanan iyileştirme yapılmış test süreci aşağıdaki gibidir:

- Başarılı, tanımlanmış, ölçülmüş, etkili ve verimli,
- İstatistiksel olarak kontrol edilen ve önceden bildirilebilir,
- Hata önleme üzerine odaklanmış,
- Kaynakları olabildiğince verimli kullanabilen bir otomasyonla desteklenmiş,
- Sanayiden organizasyona yapılan teknolojik transferleri destekleyen,
- Testin tekrar kullanılabilirliğini destekleyen,
- Sürekli gelişmeyi sağlayabilmek için süreç değişikliğine odaklanabilen.

Test sürecinin altyapısının gelişmesini sürekli desteklemek, belirlemek, planlamak ve test gelişiminin tamamlanması için test süreciyle ilgilenen bir grup oluşturulmalı, personel becerilerinin gelişmesi için özel eğitim programları almalı ve grup istediği bilgiye ulaşabilmelidir. Bu gruba “Test Süreç Grubu” adı verilir. Test süreç grubu formal olarak TMMi 3 seviyesinde test organizasyonu oluşturulduğunda başlar. TMMi 4 ve 5 seviyesinde sorumluluk arttığı için yüksek seviyeli uygulamalar tanıtılır, tekrar kullanılabilir test maddeleri belirlenir, geliştirme ve bakım için test kütüphanesi oluşturulur.

Hata önleme test süreç alanı, yaşam döngüsü içinde hataları, nedenleriyle önceden analiz edip belirlenmesi için tespit eder ve gelecekte oluşabilecek benzer hataların önlenmesi için gerekli olan aktiviteleri tanımlar. Sıra dışı hatalar test süreç performansında kalite kontrolünün bir parçası olarak tanımlanır ve hata önleme sürecinin bir parçası olarak analiz edilir.

Test süreci, “Kalite Kontrol” süreç alanında istatistiksel olarak yönetilir. İstatistiksel örnekler, güvenilirliğin ölçülmesi, güvenilirlik aralığı test sürecinde değerlendirilir.

TMMi 5 seviyesinde “Test Süreci İyileştirme” süreç alanı, uyum için mekanizmayı ortaya çıkarır. Yeni test tekniklerinin geliştirilmesi ve seçilmesi için belirli bir prosedür vardır. Test tasarımı, regresyon testi, test senaryolarının yönetimi, hata toplama, analiz ve test koşturma süreçlerinde test araçları destek olurlar.

TMMi 5 seviyesindeki Hata Önleme, Kalite Kontrol ve Test Süreci İyileştirme süreç alanları sürecin ilerlemesi için destek olurlar. Bu üç test süreç alanı birbirleriyle yüksek derecede alakalıdır. Hata Önleme süreci, performans sürecinde sıra dışı verileri analiz ederek, hata kapsam analizi yaparak ve hataların tekrar oluşmasına engel olarak Kalite Kontrol sürecine destek olur. Kalite Kontrol süreci, Test Süreci İyileştirme sürecine katkıda bulunur ve Test Süreci İyileştirme süreci Hata Önleme ve Kalite Kontrolü süreçlerine destek olur. Tüm bu süreç alanları dönüşüm içindedir, düşük seviyeli süreç alanları uygulandığında bu uygulamalardan destek alınır. TMMi 5 seviyesinde test etmenin amacı hataları önleme üzerinedir.

TMMi 5 için süreç alanları aşağıdaki gibidir.

- a. Hata Önleme
- b. Kalite Kontrol
- c. Test Süreci Optimizasyonu

a) Hataların önlenmesi süreç alanı

Hataları önlemenin amacı yaşam döngüsünün gelişiminin karşısındaki hataların ortak nedenini analiz etmek ve tanımlamaktır ve ilerde karşılaşılabilecek benzer hataların tanımlanmasıdır.

Evrimsel test modelinin doğrultusunda, TMMi Seviye 5 hata bulmaya yönelik süreç olmaktan hata önlemeye yönelik bir süreç haline gelmiştir. Bu test görüşünün doğrultusunda test etme hatalardan korunmaya odaklanır. Hataların önlenmesi geçmişte karşılaşılan hata önlemlerini tanımlama sebeplerini ve ilerdeki hata çeşitlerinin oluşmasını engelleyecek belirli aksiyonları almayı içerir. Analiz edilecek hataların seçimi, risk faktörünün de dahil olduğu çok çeşitli faktörlere dayanır. Eklenen değer en çok olduğu hatadan korunma alanlarına ve/veya hataların en kritik olduğu alanlara odaklanmak gerekir. Hatadan korunma aktiviteleri organizasyonda alınan derslerin de bir örneğidir.

Bir ürünün hatalarının başlangıç safhasında önlenmesi kaliteyi ve üretkenliği geliştirir. Hataların bulunmadan önce önlenmesi genellikle projenin maliyetini olumlu yönde etkiler. TMMi 5 seviyesinde organizasyon, hangisinin daha

maliyet etkili olduğunu,hatanın belirli bir çeşidinin korunması veya tespiti olup olmadığını bilecektir. Süreç gelişim modelleri, gündelik analiz kullanımını,işlem olgunluğunu bir araç olarak vurgular.Gündelik analizler için metot örnekleri, belirli gündelik analiz toplantıları,hata analizi gibi araçların kullanımı ve neden/sonuç şekilleri,resmi incelemeler sırasındaki gündelik analiz ve standart hata sınıflandırılmasının kullanımınıdır.

Hataların önlenmesi süreç alanı, hataların ortak nedenlerini tanımlama ve analiz etmeyi ve gelecekteki hata çeşitlerinin ortak nedenlerini ortadan kaldırmaya yönelik belirli hareketleri tanımlamayı hem projede hem de organizasyonun herhangi bir yerinde adres gösterir. Gerek geliştirme sırasında gerekse test aşamasında bulunan tüm hatalar süreç alanının kapsamındadır.

b) Kalite kontrol süreç alanı

Kalite kontrolün amacı istatistiksel olarak test sürecini yönetmek ve kontrol etmektir. Bu seviyede test işlem performansı tamamen öngörülebilir ve kabul edilir sınırlarla stabilize edilir.

Ürün kalitesini ön görebilmek ve testi çok daha etkin yapabilmek için temsili örneklere dayanan istatistiksel metotları kullanarak proje seviyesinde test etme uygulanır.

Kalite kontrol bir iş ürününün ve taşınabilirliğin standartlara ve ihtiyaçlara uydurulmasını sağlamak için kullanılan prosedür ve alıştırmaları içerir. Süreç kalite kontrolü istatistiksel teknik ve metotlarla desteklenir. Kalite kontrol sürecinin temeli girdi ve çıktı kümeleriyle her birinin kendi içinde bir süreç olduğu test sürecinin genel açıdan görünümüdür. İdeal olarak her adımın çıktısına, nasıl yürütüleceğini tarif eden kurallar,prosedürler ve/veya standartlar aracılığıyla karar verilir.

Farklılıklar varyasyonlardan meydana gelir.Varyasyonlar insan hatası,süreç dışında etkilenmeler,tahmin edilemeyen olaylar donanım/yazılım bozuklukları ve dahası nedenleriyle olabilir.Eğer süreç adımını etkileyen birçok öngörülemeyen varyasyonlar varsa, o zaman süreç değişken,tahmin edilemez ve kontrol dışı olur.Bir süreçle ilgili tahmin yürütülemezse o zaman o süreç kalite sonuçları verecek kadar güvenilemez.

Süreçlerini ölçülü şekilde kontrol eden bir organizasyon aşağıdakileri yapabilir:

- Sürecin sabitliğine karar verir,
- Süreç performansın tanımlanmış doğal sınırlarla tanımlar,
- Tahmin edilemeyen süreçleri tanımlar,
- Mevcut süreçlerdeki gelişim fırsatlarını tanımlar,
- En iyi uygulanma süreçlerini tanımlar.

Kalite Kontrol Süreci test yaşam döngüsünde tanımlanan performans için amaçlar oluşturmayı içerir. Bu amaçlar tanımlanan test politikasına dayanmalıdır. Test yaşam döngüsü ve entegrasyon süreç alanında daha önce belirtildiği gibi çoklu standart test süreçleri farklı başvuru alanlarının ihtiyaçları, test seviyeleri, yaşam döngüsü modelleri, metotlar ve organizasyonda kullanılan araçlara başvurmak için var olabilir.

c) Test süreç iyileştirmesi süreç alanı

Test Süreç Optimizasyonunun amacı organizasyonda kullanılan, var olan test süreçlerini sürekli olarak geliştirmek ve uygun olabilecek test teknolojilerini tanımlamak ve düzenli bir biçimde organizasyona geçiş sağlamaktır. Bu gelişmeler, organizasyonun iş amaçlarından derlenmiş olan organizasyonun ürün kalitesi ve test süreç performans amaçlarını destekler.

TMMi modelinin en üst seviyesinde ,test süreci projelerde ve tüm organizasyonda sürekli gelişmelidir. Test süreci kapasite büyümesinin devam eden bir süreç olması için ölçülebilir. Organizasyonlar altyapının devamlı büyümesini desteklemek zorundadır. Politikalar, standartlar, eğitim faaliyetleri, araçlar ve organizasyonel yapılar içeren bu alt yapı, TMMi hiyerarşisini atayan süreçler arasında yer alır.

Test süreç optimizasyonu özünde test etmeyi sürekli geliştirmek için sistem geliştirme hakkındadır. Test optimizasyon sürec alanı aşağıdakileri içerir:

- Test süreç değerlendirme ve iyileştirme prosedürlerinin sorumlularının belirlenmesi,
- Zayıf ve güçlü olan test uygulamalarının tekrar kullanım için tanımlanması,
- Organizasyonun test süreç ve teknolojilerini geliştiren uygulamaları devreye alması
- Organizasyon için en iyi uygulamaların seçilmesi,
- Adaptasyon için yeni test araçlarını ve teknolojileri takip etmek,

- Teknoloji ve bilgi transferinin desteklenmesi,
- Yüksek kalite test özelliklerinin tekrar kullanılması.

Test sürecini sürekli olarak geliştirmek, organizasyonun standart test süreci ve projelerin tanımlanan süreçlerini devam eden temel üzerinde tanımlanmasını ve yeni süreçlere adapte olmasını sağlar. Test süreci gelişim aktiviteleri içinde değişen çevrenin bir sonucu olarak da yeni bir test yaşam döngüsüne ihtiyaç duyulur. Bu, daha yüksek seviye yönetim desteğiyle yapılır. Eğitim ve teşvik edici programlar test süreç gelişim aktivitelerine katılacak organizasyondaki herkese cesaret vermek için oluşturulur.

Test Süreç İyileştirmesi planlama, geliştirme, izleme, değerlendirme ve test gelişim hareketlerini ölçmeyi içeren yeni test teknolojileri seçmeyi ve onları organizasyonun standart test sürecine entegre eden pratikleri içerir. Bu süreç alanı test yaşam döngüsü ve entegrasyonu süreç alanları tarafından TMMi beşinci seviyede süreçleri tamamlar ve genişletir.

5. RISK FAKTÖRÜ VE ÜRÜN/PROJE RISK YÖNETİMİ

Bu bölümde risk faktörü ve ürün risk yönetimi konuları özetlenecektir.

5.1 Risk Faktörüne Genel Bakış

Risk, bir olayın olumsuz ya da istenmeyen bir şekilde sonuçlanma olasılığıdır. ISTQB'de ise risk faktöründen gelecekte olumsuz sonuçlanabilecek bir etki ya da olasılık olarak bahsedilir. Riskler bir problem oluştuğu zaman müşterinin, kullanıcının, ortakların ya da paydaşların ürün kalitesiyle ya da proje başarısıyla ilgili beklentilerini düşürmelerine sebep olur. Risk genelde duruma ya da bakış açısına göre çeşitlenebilen kişisel bir değer olarak algılanabilir. Risk, olması gereken durumun etkilerinin aksine istenmeyen bir durum olma olasılığını dengelemek için hesaplanır. Bunun karşılığında işin ne kadarının şansa bırakılabileceği görülmüş olur.

Özünde risk zamandan kaynaklanan belirsizlikler ile ilgili bir sorundur. Bir aktivitenin başlamasıyla bitişi arasında geçen zamanın etkisi, aktivitenin geliştirme sürecindeki çıktılar ile ilgili belirsizlik gösterir. Tüm riskler her zaman endişeye sebep olmamalıdır. Risk seviyesi iki özellik ile belirlenir.

Problemin oluşma olasılığı ve problem oluştuğundan sonra beklenen ile planlanan sonuç arasındaki negatif değer sapması olarak tanımlanabilir.

Zaman ve teste ayrılan bütçe ne kadar fazlaysa o kadar test yazılabilir. Daha fazla test ile riskler direkt olarak ya da dolaylı olarak azaltılmış olur ve test aşamasında bulunan ürünün durumu hakkında paydaşlara daha kesin ve güvenilir bilgiler sağlanabilir. Bir ürünün ya da altsistemin risk seviyesini ölçebilmek için sırasıyla ürünün kendisinin ve çevresinin analizi ve durumu gereklidir. Ne kadar risk içerdiği belirlendikten sonra risklerin bir kısmının ya da hepsinin azaltılması ya da yok edilmesi için ölçütler alınır. Bu risklerin statik olmadığı akılda tutulmalıdır, riskler ürünün kendisi kadar dinamiktir. Riskler ortaya çıkarılır, silinir ya da değiştirilir.

Genel olarak riskler iki farklı kategoriye ayrılır. Ürün riskleri ve Proje riskleri:

a) Ürün riskleri

Ürün kalitesindeki potansiyel problemlerin belli olduğu yerlerde potansiyel problemler “ürün riskleri” ya da “kalite riskleri” olarak adlandırılır. Örnek vermek gerekirse güvenlik eksikliği canlı ortamda ürünün başarısızlığına sebep olabilir. Test disiplini bu tip risk gruplarına odaklanır. Test, ürüne fonksiyonel ya da fonksiyonel olmayan gereksinimler, ürünün karmaşıklığı ve güvenilirliği, dokümantasyon kalitesi ve yazılım kodunun kalitesi şeklinde bakar. Test etmek tüm bu ürün risklerinin uygun testlerle kapsanmasını amaçlar.

b) Proje riskleri

Yüksek kalitede olan gereksinim anlaşmaları, katı proje yönetim planları ürünün başarısını garantilemez. Bu iyi bir başlangıçtır ancak projeyi destekleyen yeterli kaynak yoksa başarı şansı azalır. Projenin başarılı olması için bir kaç önemli konu vardır.

- Personelin gerekli bilgi, yetenek ve beceriye sahip olması,
- Doğru test araçlarının kullanılabilirliği,
- Test ürünü için uygun ortamın bulunması,
- Konfigürasyon yönetimi gibi aktivite desteğinin verilmesi,
- Tedarikçinin verimi.

Bu alanlardan birinde olan potansiyel problemler proje riski olarak adlandırılır. Bir sonraki bölümde incelenecek olan PRISMA yaklaşımı ürün riski için olan risk yönetimine ve riske yönelik teste odaklanır. Proje risk yönetimi ile ilgili tartışılan konular uygulanabilir. Yukarıda bahsedilen iki risk tipi bağımsız değildir. Ürün riskleri proje risklerine yönlenebilir. Yetersiz geliştirme yapılan üründe daha fazla hata bulunur ,daha fazla hata giderildiği için daha fazla tekrar testi yapmak gerekecektir bu yüzden projenin bitiş tarihi gecikir. Eğer yazılımcıların yeterli bilgi ve becerileri yoksa ürün hataya meyilli olarak ortaya çıkar.

Paydaşların, özellikle iş temsilcilerinin riskler üzerinde daha önce kazandıkları deneyimler vardır. Başarısızlıkla sonuçlanacak olan ürün risklerini görmek istemezler. Bir şekilde daha fazla ödeme yaparak bu risklerden kurtulmayı tercih ederler. Paydaşlar risk sorumluluğu alınmasında en önemli rolü üstlenirler ve bu yüzden risk tanımlama aşamasına, analizlere, önceliklendirmelere risk azaltma uygulamalarına dahil olmak zorundadırlar. Paydaşlar sistemin gerçek ortamdaki hareketlerinden de sorumludurlar.

Sistemin kurulması için son kararın verilmesi ve test ortamından canlı ortama taşınması kolay olmaz. Genelde bütçenin büyük bir kısmı pazarlamaya ve yeni ürünlerin geliştirilmesine ayrılır ya da mevcut ürünler şirketin önümüzdeki birkaç yıl içindeki pozisyonunu belirleyecektir. Sistem test ortamındayken hataların bulunup giderilmesi için fırsat vardır. Ancak sistem bir kere canlı ortama çıktığı zaman bu kararı veren kişi kendi üzerine dikkatleri çekmiş olur. Ürünün canlı ortama çıkma kararını testçi vermez. Testçi iş açısından paydaşlara bilgi sağlar. Testçiler kendi yönetimlerini sunar ve diğer paydaşları bağımsız bir şekilde bilgilendirirler. Bir ürünün yeteri kadar iyi olup olmadığı değerlendirileceğinde, testçi elde edilen bulguların faydalarının olduğunu ancak bazı ürün riskleri içerdiğini söylemelidir. Riskler izlenmeli gerekliyse tarih bilgisi tutulmalıdır. Testçiler paydaşların karar vermesine yardımcı olurlar ancak kendileri karar vermezler. Bazı ürün riskleri testçilerin bakış açısına göre sistemi kabul edilemez yapabilir eğer bu risklerin kritik olmasıyla ilgili paydaşlarla baştan anlaşmaya varıldıysa paydaşların bu risklerle ilgili algısı değişene kadar sistem kabul edilmemelidir.

Test için organizasyonun, riske olan bakış açısını algılamak önemlidir. Bazı organizasyonlar canlı ortama çıkarken kasıtlı olarak risk alırken bazıları tamamen riske karşıdır. Organizasyonlar belirli riskleri kabul etmek konusunda farklı davranışlar sergiler. Projenin ya da organizasyonun risk tutumunun anlaşılması, riske yönelik testle ilgili ya da risk yönetimiyle ilgili yaklaşımda etkili olana kadar önemlidir.

Özetle, test etmek, paydaşlara verilen bilgilerin doğruluğundan ve anlaşılabilirliğinden emin olmamızı sağlar.

5.1.1 Test aktivitelerinin sınırlamaları

a) Ayrıntılı test

Test hataların bulunmasına, ürün risklerinin azaltılmasına, ürün riskleriyle ilgili bilgi edinilmesine ve ürün kalitesinin artırılmasına yardımcı olur. Ne kadar testin yeterli olduğu ya da kesin sonuçlar için ne kadar bilginin gerekli olduğu tartışılabilir. Bu aşamada seçenekler ortaya çıkar. Sistemdeki herşeyin test edilmesi, hiçbirşeyin test edilmemesi ya da sistemin bir kısmının test edilmesi bu seçenekler arasındadır. Yöneticiler herşeyin test edilmesi konusunda hem fikirse bu, sistemin her yönünün tüm durumlarının uygulanması anlamına gelir. Ancak sistemi yüzde yüz kapsamayla test etmek imkansızdır. Ayrıca hiçbir şirket kaynakların büyük bir kısmını bu şekilde harcamak istemez.

Projede zaman ve parayla ilgili baskı, projenin gereksinimlerinin karşılanmasıyla ilgili baskı kadar fazladır. Kaynaklar her zaman limitlidir. Proje yöneticileri onlara yatırım olarak dönmelerini sağlayacak şekilde bir test süreci isterler. Proje sürümü yayınlandıktan sonra önlenecek hatalar pahalıya mal olacaktır.

b) Test sürecinin sıkıştırılması

En çok bilinen test zaman aşımı, yazılımcıların testçilere olan teslimatı geciktirmesiyle son teslimatın tarihinin aynı kalması durumunda oluşur. Tamamlama kriterleri bir sonraki aşamada ne olacağını belirler ancak genelde bu kriterler zamanında karşılanmaz. Sürümün çıkmasıyla ilgili baskı kriterlerin sağlanması şartının yerine getirilememesini sağlar.

c) Tamamlama kriterleri

Özel ve genel durumlar için sürecin tamamen bitirilmesine izin verilmesi için paydaşların anlaşmaya varmış olması gerekir. Bitiş kriterlerinin amacı henüz tamamlanmamış olan işlerin tamamlanmış gibi kabul edilmesini önlemektir. Bitiş kriterleri testin ne zaman bitirileceğinin planlanmasını sağlar.

Bazı önemli hataların giderilemesi ya da ürün risklerinin azaltılması için girişimler yapılabilir. Bitiş kriterleri tanımlanan kalite hedefleri için hatırlatıcı niteliktedir ve projenin erken evrelerinde bununla ilgili anlaşmaya varılır. Test süreci sıkıştığı zaman, bazı ürün riskleri göze çarparken ,testçiler yeni ürün

riskleri ortaya çıkarabilirler bu durumda göze çarpan ürün risklerinin hepsi kapsanmalıdır. Eğer paydaşlar ürünün bu şekilde ortaya çıkmasına karar verirlerse daha sonra ortaya çıkacak olan ürün riskleri testçilerin problemi değildir.

5.1.2 Test sürecinin risk odaklı yürütülmesi

Herşeyi test etmek mümkün olmadığı ve testin sıkıştırılması gerektiği zaman, ürünün ve projenin doğru kısmının test edilmesi için bir yaklaşıma ihtiyaç vardır. Bu testin odak noktalarının bulunmasıyla ve ürün riskleriyle ilgili testlere yönelmesiyle olur. Ürün risklerinin değerlendirilmesi ve yönetilmesi her proje için en önemli, aynı zamanda en kritik aktivitelerdendir. Ne kadar testin yeterli olacağına karar vermek için ürün ve proje risk seviyelerine göre zaman ve para kısıtlamalarının ne kadar olacağı da hesaba katılmalıdır.

Ürün risk değerlendirmesi ne kadar testin gerektiğine karar vermek için gerçekleştirilir. Riskin seviyesine ve tipine göre tanımlama yapılır, test yaklaşımı, test eforu ve gerekli testlerin dağılımı çeşitlendirilir. Test süreci, sistemin bir sonraki geliştirmeye kadar test edilmiş olduğuna paydaşları ikna etmelidir. Bütçedeki, zamandaki ve test sürecindeki kısıtlamalar ürün riskine yönelik teste öncelik verilmesi gerektiğini göstermiştir.

5.2 Ürün Risk Yönetimi

Ayrıntılı testin imkansızlığı ve test için ayrılan sıkışık takvime karşılık, risk odaklı test yaklaşımı veya yeterli olacak kadar test uygulanabilir. Projenin sınırlamaları ve testin limitleri kapsamında risk odaklı test ana riskleri kapsayan ve yapılabilecek en iyi test anlamına gelmektedir.

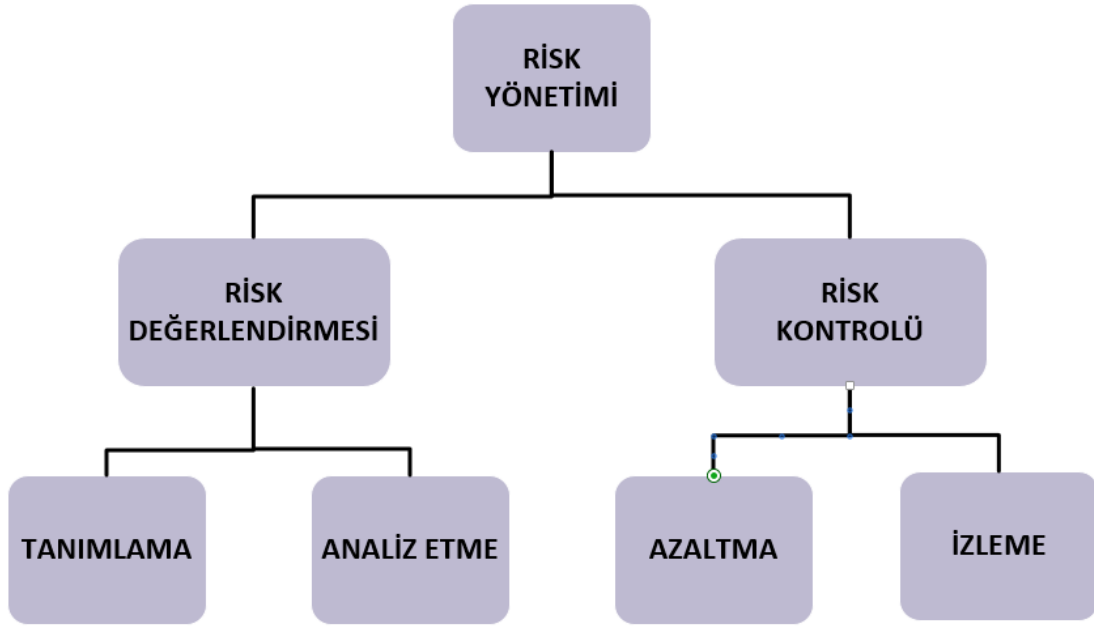
Test edilmesi gereken en önemli alanlar iş gereksinimleridir. Bunlar kullanım sıklığı ile ilgili konuların, fonksiyonların dışardan görünürlüğünün ve hataların olası etkilerinin analizi ile belirlenebilir. Başka bir açıdan fazla hata çıkması beklenen yerlerin çok test edilmesi anlamına gelebilir. Bu, sistemdeki hata yoğunluklu alanların ve karmaşıklık, yeni teknolojiler ve deneyimsiz yazılımcılar gibi başarısızlık olabilecek nedenlerin belirlenmesiyle ortaya çıkarılabilir.

5.2.1 Risk yönetiminin temel aktiviteleri

Risk, zaman, bütçe ve kalite hedefleri gibi limitler içinde koşturulan testlerin önceliklendirilmesinin ve yoğunluğunun hesaplanması için kullanılır. Testler gerçek ya da risk odaklı olarak adlandırılabilir. Bu aynı zamanda yeni ürünlerde ve bakım paketlerinde de uygulanır. Bir projede ya da organizasyonda risk odaklı testin uygulanabilmesi için ürün risk yönetim sürecine ihtiyaç vardır. Risklerin üzerinin örtülmesinden açık bir şekilde belirtilmesi gerekir böylece kişisel başarı ve kişisel kararlardaki başarıya güvenilebilir. Bunun için analize ve sonuç yaklaşımına ihtiyaç vardır. Risk yönetimi dört ana aktivite içerir;

- **Risk tanımlaması:** Hangi risklerin nerede olduğunun belirlenmesi için yöntemler aranmasıdır.
- **Risk analizi:** Tanımlanan risklerin, kritik karmaşık ve potansiyel hata eğilimli alanların altında yatan nedenleri gösterir kritik karmaşık ve potansiyel risk eğilimli alanların oranlarına bakar ve bu bilgileri önceliklendirme için kullanır.
- **Risk azaltılması:** Tatmin edici durumların yaratılması için uygun olan alternatif aksiyonların alınması sağlanır ,aksiyonlara karar verilir ve uygulanır, ürün risklerinin azaltılması için testler tasarlanır ve uygulanır.
- **Riskleri izleme:** Süreç boyunca riskler takip edilir uygulanan aksiyonların yapısı ve büyüklüğü saptanır ve risk durum raporları çıkarılır.

İlk iki aktivite risk değerlendirmesi olarak son iki aktivite de risk kontrolü olarak adlandırılabilir. Şekil 5.1 risk yönetiminin ana temalarını sistematik olarak göstermektedir. Risk kontrolleri ürün risk yönetiminde önemli bir rol almaktadır. Bu tanımlanmış ve analiz edilmiş ürün risklerinden alınan sonuçların tüm ölçümlerini içerir.



Şekil 5.1 Ürün risk yönetimi aktiviteleri.

Risk tanımlaması ve risk analiz aktiviteleri hangi yazılım geliştirme yaşam döngüsü modeli izlenirse izlensin,projenin ilk aşamalarında başlamalıdır. Risk azaltma aktiviteleri planlanmalı ve test planlama süreci boyunca uygulanmalıdır. Ana test planında ve test planlarının farklı aşamalarında proje ve ürün risklerinden bahsedilmelidir. Risklerin tipleri ve seviyeleri, ürün risklerinin belirtildiği test seviyelerinde, risklerin azaltılması için harcanan test eforu ve risklerin azaltılması için tekrarlanarak uygulanan diğer test ve tekniklerde, risklerin giderilmesi ile ilgili yeterlilik kriterlerinin belirlenmesinde etkili olacaktır. Testçiler kendi profesyonel oldukları alanlardaki tutumlara göre ürün risk yönetiminde belirli uzmanlıklar geliştirmelidir.Böylece risk tanımlamaları esnasında sürece ortak olabilirler. Aslında bir çok aşamada test yöneticisi testçilere anahtar noktalarda destek sağlayarak ürün risk değerlendirmesini yönlendirebilir.

Risk yönetimi aktiviteleri biraz öngörüye bağlı olmasına rağmen aslında tamamen pratik deneyime dayanır. Proje boyunca sürekli olan bir risk yönetimine ihtiyaç vardır. Risklerin belirlenmesi, risk analizi, risk oranlarının azaltılması ve risklerin izlenmesi aktiviteleri yinelenen aktivitelerdir. Risk yönetimi aktivitelerinin bu dört temel adımı tekrarlanarak uygulanabilir. Eğer organizasyonun test stratejisi ya da

test politikası dokümanı varsa burada ürün ve proje riskleri test ile referans gösterilebilen bir süreç içinde anlatılmalıdır. Bu dokümanlar risk yönetiminin testin tüm aşamalarıyla bütünleştiğini göstermelidir. Aşağıda bu dört bölüm özetlenmiştir:

a) Risk tanımlaması

Risk tanımlaması işlerin nerede ve nasıl yanlış gideceğinin belirlenmesiyle ve tanımlanan risklerin daha fazla analiz için dokümante edilmesiyle ilgilidir. Temel olarak daha fazla detaylı analiz için gerekli olan maddeler listesi tanımlanır. Olabildiğince tamamlanmış bir ürün riskleri listesi elde etmek için test yöneticisi sistemle direk ya da dolaylı olarak ilgili olan paydaşları da bu aşamaya dahil etmelidir. Formal bir şekilde paydaşların analizi de istenebilir.

Yeterli ve uygun paydaşların seçildiğinden emin olmak önemlidir. Bu kritik paydaşların seçilmesi için gereksinim dokümanlarındaki, proje planlarındaki ve diğer proje dokümantasyonlarındaki paydaşların referanslarını kontrol etmek iyi bir yöntem sayılabilir. Katılımda bulunmamış bir paydaş tanımlanmamış bir ürün riskine dikkat çekecek olan kişi olabilir. Bir çok durumda paydaşlar katılım sağlamayabilir ya da bu konuda istekli olmayabilirler. Bazı durumlarda da paydaşlar diğer paydaşların vekilleri gibi hareket edebilirler.

Risk tanımlaması sistemin test esnasındaki öncelikleri ile ilerletilen bir uygulamadır. Ürün risklerinin tanımlandığı listeyi oluşturmak kolaydır. Önemli olan risklerin belirlenmesidir.

Test sadece ürün risk değerlendirmesini kolaylaştırır. İşin özünde risk zaten vardır. Paydaşlar hangi sebeple olursa olsun ürün riskini belirlemekte başarısız olurlarsa bu testçilerin değil paydaşların sorumluluğundadır. Paydaşların riskin asıl sahibi olduklarını bilmeleri ve bununla yüzleşmeleri önemlidir.

Bu seviyede asıl tehlike mevcut risklerin unutulması fazlasıyla iyimser davranılması ya da risklerin etkileriyle ilgili yanlış tahminlerde bulunulmasıdır. Bu riskleri en aza indirmenin yolu ürün risk tanımlamaları ile analiz aktivitelerinin ayrılarak risk tanımlamalarının belirlenmesidir. Genellikle bunlar birlikte yürütülür ve ürün riskleri tamamlandıktan sonra analiz aktiviteleri hızlı bir şekilde yapılır. Risklerin belirlenmesi ve risk analizi iki ayrı ama birbirine de bağlı aktivitelerdir.

Riskler her zaman belirgin değildir. Bilgi sahibi ve deneyimli paydaşlarla çalışıldığı zaman risk tanımlama tekniklerinden yararlanmak kullanışlı olabilir. Bu tekniklerin uygulanabilirliği geliştirme sürecinin tipine, personelin deneyim ve yeteneklerine, zamanın uygunluğuna ve yaşam döngüsünde risk tanımlamasının yapılmasına bağlıdır. En çok kullanılan teknikler aşağıda maddeler halinde belirtilmiştir.

- i. Gereksinimlere Dayalı
- ii. Beyin Fırtınası
- iii. Uzman Görüşlerden Yararlanılarak
- iv. Bağımsız Risk Değerlendirmesi
- v. Kontrol Listesi
- vi. Risk Seminerleri
- vii. Çıkarılan Dersler(Geçmişe Yönelik Toplantılar)

Ürün risklerinin tanımlanmasında etkinliğin artırılması için yukarıdaki tekniklerin karıştırılarak kullanılması tavsiye edilir. Test yöneticisinin ürün risklerini tanımlamak için kullandığı tekniklerin kombinasyonları, ürün riskleri üzerinde bütün halde bir görüş elde etmeye yarar. Risk kayıtları ne kadar eksiksiz ise etkili test o kadar iyi yapılır.

Risk tanımlama aşamaları yönlendirilirken ürüne farklı seviyelerde aşina olan personelin de olması yardımcı olacaktır. Ürüne olan aşinalıklar farklı görüş açıları sağlarken çok açık ortada olan riskler gözden kaçırılabilir. Geliştirme sürecine ya da ürüne aşinalığın az olması risk değerlendirme sürecinde tarafsız görüşler yapılabilmesini sağlar.

i. Gereksinimlere dayalı risk tanımlaması

Her gereksinim bir şekilde ürün riskleri ile ilişkilendirilene kadar tanımlanan gereksinimler ürün risklerinin belirlenmesi için bir başlangıç noktası oluşturabilir. Ortaya çıkan üründe gereksinimlerin karşılanmamış olması projede beklenen ve şimdiki çıktılar arasında bir sapma olarak algılanabilir bu nedenle bu da bir risktir. Bir gereksinimin karşılanmamış olması canlı ortamın verimli ve etkili olmadığını ve kaynakların boşa harcandığı anlamına gelir. Bir çok gereksinim kolaylıkla ürün riski olarak düzenlenebilir. Ürün riskleri listesinin kolay yönetilebilmesi için gereksinimler gruplanabilir.

Projelerde her zaman ürün risklerinin türetilbileceği eksiksiz ve kesin bir gereksinim listesi olmayabilir. Sonuç olarak gereksinimlere dayalı ürün risklerine genel bakış için diğer tamamlayıcı teknikler gerekebilir. Gereksinime dayalı yaklaşım ve özel risk tanımlama tekniklerinin çıktılarını aynı anda kullanmak ürün risklerini dikkate almak için oldukça iyi bir yaklaşımdır.

ii. Beyin fırtınası ile risklerin tanımlanması

Beyin fırtınası bir ya da daha fazla toplantı üzerinden iletişimin yürütüldüğü bir yöntemdir. Hiç kimse bir sistemin nasıl çalıştığı ile ilgili genel resmi göremez. Geniş bilgi ve deneyim yelpazesine sahip uzmanlardan oluşturulmuş bir takım sürecin kalitesi için gereklidir, böylece projede çalışan kimsenin görüşü tek başına ele alınmaz. Takımın objektifliği geniş perspektifin ve sistemin tüm görüşleri üzerindeki algının sağlanması ile ölçülebilir.

Beyin fırtınası toplantıları iki saatten fazla olmamalı ve risk tanımlamalarına odaklı değildir. Daha sonra bilginin pekiştirilmesi için analiz aktiviteleri başlar. Katılımcıların eleştiri konusunda kendilerini rahat hissettikleri destekleyici bir ortam sağlanması çok önemlidir. Beyin fırtınası yöntemi temel olarak aşağıdaki maddeleri içermelidir.

- Diğer katılımcıların önerilerinin desteklenmesi,
- Risklerin kısa başlıklar halinde ve kaynaklardaki belirtiler kullanılarak kaydedilmesi,
- Eleştiriye yargılamaya beyin fırtınası esnasında yer verilmemesi,
- Değerlendirme ve analizin sonraya bırakılması,
- Takımın disiplin ve deneyim konusunda geniş yelpazeye sahip olması,
- Beyin fırtınası aşamasında bir toplantı yöneticisinin belirlenmesi,
- Sistemin kapsamının ve amacının beyin fırtınası oturumundan önce iyi tanımlanması,
- Tanımlanmış ürün risklerinin üzerinde durulması için alt gruplar oluşturulması.

iii. Uzman görüşlerden yararlanılarak risklerin tanımlanması

Bilgi sahibi olan paydaşlar beyin fırtınası için ya da diğer risk tanımlamaları için uygun değilse uzman kişiler ile planlama yapılabilir. Yöneticiler, dışardan gelen danışmanlar, deneyimli proje ya da test yöneticileri uzmanlarla olan

görüşmelere katılım sağlayabilirler. Bu görüşmelerin 30-40 dakika arasında tutulması uzmanların dikkati açısından önemlidir.

Görüşülen uzmanlar ifadelerinde ,proje amaçları ve paydaşların istekleri gibi müdahalelerden etkilenmemelidirler. Projenin durumu ve karakteristikleri ile ilgili iyi ve objektif bilgilendirme yapmalıdırlar. Bu görüşmelerin en büyük dezavantajı çapraz görüşlerden yoksun olmasıdır. Bir diğer dezavantaj ise bu görüşmelerin beyin fırtınası oturumlarından daha az yaratıcı olmasıdır. Beyin fırtınası oturumları gibi görüşmelerin amacı analiz çıktılarından ziyade verilerin miktarlarının belirlenmesidir.

iv. Bağımsız risk değerlendirmesi

İç kaynaklarla birlikte ya da iç kaynaklar yerine dışardan uzmanlara danışılabilir. Bağımsızlık, uzmanların organizasyon dışından olması gerektiği anlamına gelmez. Ancak her kim bu değerlendirmeyi yapacaksa o kişi ürün risk tanımlaması konusunda deneyimli olmalı ama ürün üzerinde ticari kazancı olmamalıdır.

Değerlendirme bir dizi rehber doküman eşliğinde düzenlenmeli ve bu alınan kısımlar ürün hakkında öncelik ve kullanım gibi ilgili alanlara odaklanacak şekilde yeterli bilgilendirmeyi yapmalıdır. Karşılaştırma için benzer bir üründen referans alınabilir böylece eksperler tarafından ürünün fonksiyonlarının anlaşılması kolaylaşır.

v. Kontrol listesi

Süreci kısaltma girişiminde kontrol listeleri tüm risk tanımlama süreci kapsamında kullanılabilir.Risk tanımlamaları için olan kontrol listeleri daha önceki projelerde elde edilen deneyimlerin bir havuzda toplanmasını sağlar.Bu deneyimlerin aynı organizasyon içinde olmasına gerek yoktur. Liste sistem üzerindeki genel alanlara odaklı olmalıdır. Amaç unutulmuş alanlara doğru yönelirken aynı zamanda sık sık oluşan ürün riskelerine odaklanması, problem tiplerinin belirtilerini görmektir.

Genellikle kontrol listeleri çok güçlü araçlardır. Yanlış gidebilecek olan durumların kontrol listesinden ilerleyerek kontrol edilmesi yararlı olur. Kontrol listesinin dezavantajı kullanıcının kontrol listesi üzerine odaklanarak kendi yaratıcı bakış açısına zarar vermesi olabilir. Kontrol listeleri doğru seviyede

olmalı ve çok fazla detay içermemelidir. Kısa kontrol listeleri kullanmak tercih edilir. Beyin fırtınası ya da risk seminerleri gibi diğer tekniklerin kontrol listesi ile birlikte kullanılması yararlı olabilir. Kontrol listesi ile diğer tekniklerin birleştirilmesi etkinliği arttıracaktır. Eğer kontrol listeleri proje deneyimlerine paralel olarak düzenli olarak güncellenirse daha etkili olacaktır ve geliştirilebilecektir. Kontrol listeleri aşağıdaki konuları içermelidir.

- Standart ve düzenlemelere uygunluk,
- Donanıma, yazılıma ya da insan kaynaklarına gelebilecek olan iç ve dış tehditler,
- Performans kriterleri, kullanılabilirlik ve güvenilirlik ölçütleri gibi amaç ve kabul kriterleri,
- Personel seviyesinin azaltılması gibi uzun soluklu faydalar.

vi. Risk seminerleri

Risk seminerleri bir toplantı yönetici tarafından yönetilen etkinliklerdir ve devamı gelen toplantılar şeklinde gerçekleştirilir. Amaç serbest beyin fırtınası toplantılarının bir grup kontrol listesiyle birleşmesinden oluşan risk faktörlerinin genel olarak incelenmesidir. Risk seminerlerinin odağı anahtar noktalar üzerinde düşünmek olarak özetlenebilir. Farklı kaynaklardaki gruplarla ve farklı bakış açılarıyla yapılan risk seminerlerinde ürün riskleri etkinliğe göre gelişecek ve tek bir bakış açısı yerine genel resmin görülmesini kolaylaştıracaktır.

Seminerlerin katılımcıların hepsinin ürünün durumuyla ilgili yeterli bilgiye sahip olması gerekir. Seminer başlangıcında seminer takvimiyle ilgili briefing alırlar ve hepsinden 30 dakika içinde bağımsız bir ürün risk listesi çizmeleri istenir. Riskler not olarak yazılır ve kategorilerine göre düzenlenir. Tüm yazılan riskleri sonradan kategorisine göre tartışılır ve benzer olanlar silinerek birleştirilir. Bir çok ürün riski bu aşamada bulunur ve tartışma esnasında eklenir.

vii. Çıkarılan dersler(Geçmişe yönelik toplantılar)

Deneyim bağımsız ya da kurum deneyimi olarak iki seviyede kazanılabilir. Bağımsız kazanılan deneyimler öngörü ve bilginin yansımalarıdır. Kurum içerisinde kazanılan deneyim daha çeşitlidir ve gözden geçirme ve analiz

etmek için daha disiplinli yaklaşıma ihtiyaç duyar. Bu teknik geçmiş projelerdeki kayıtların varlığına dayalıdır. Eğer hiç kayıt yoksa elverişli bir teknik olmayacaktır ama mevcuttaki ve gelecekteki projeler için kayıt tutmaya başlamak iyi bir başlangıç olabilir.

Hata sayıları, hata tipleri, hata tipine bağlı çıkan hata sayıları, sistemin çökme süresi, onarım için gereken maliyet ve objektif risk örneklerinin sağlanacağı tahmini iş maliyeti gibi ürün riskleri ile ilgili verilerin kaydına tutulmalıdır. Bu verilerin analizi gelecekteki risk tanımlama aktivitelerine girdi oluşturacaktır ve kontrol listeleri güncellenecektir. Çıkarılan dersleri gözden geçirme toplantılarında henüz tanımlanmamış ancak belirli risk tipleri kesin oluşacak olan alanlara odaklanılmalıdır.

b) Risk analizi

Risklerin tanımlanması yeterli değildir. Riskler tehditlere karşı verdikleri tepkiye göre hesaplanarak analiz edilmeli ve eğer bu tehditlerle ilgili birşey yapılabilirse bunlar belirlenmelidir. Risk tanımlaması olabildiğine çok uygun alakalı risklerin tanımlanmasıyla ilgiliyken risk analizi bu tanımlanmış risklerin gruplanması ve her risk ile ilgili olasılıkların ve etkilerin hesaplanması üzerinde yoğunlaşır. Risk analiz sonucu tüm ürün risklerinin önceliklendirildiği, tanımlandığı, oluşma olasılık tahminlerinin listelendiği ve risk ortaya çıktığında görülebilecek olan zararı içeren kapsamlı bir risk envanteri içerir.

i. Gruplama

Risklerin gruplanması her riskin uygun bir kategori ya da risk tipi altına yerleştirilmesidir. Bir çok sistemde ürün risklerinin gruplanması için kalite karakteristikleri olan güvenilirlik, güvenlik, kullanılabilirlik ve bakım kolaylığı kategorileri kullanılır. Bazı organizasyonların kendi kalite karakteristik grupları vardır. Çok fazla tanımlanmış ürün riski varsa gruplandırma yönetilebilir bir bakış açısı sağlaması açısından kullanışlıdır. Gruplama, risk tipi direk olarak performans ya da güvenlik testi gibi bir test tipiyle alakalıysa belli kategorilerdeki ürün riskleri ile alakalı test tipleri uygulanarak risk hafifletme aktivitelerine de yardımcı olur. Risk tanımlanması sırasında destek olarak kontrol listesi kullanılıyorsa gruplama test tanımlanması esnasında oluşabilir.

ii. Risk tipi

Risk kümeleri kalite niteliği, nedeni, lokasyonu ya da riskin potansiyel etkileri gibi bir ya da birden fazla gruba ayrılabilir. Riski azaltacak test tipiyle alakalı risk tipleri de gruplanabilir.

iii. Önceliklendirme

Ürün riskleri tanımlandıktan ve kategorize edildikten sonra önceliklendirilmelidir böylece test aşamasında öncelikli ve çaba gerektiren bölgelere odaklanılabilir. Önceliklendirme, her risk maddesini, oluşma olasılığını ve risk oluştuğundan sonraki etkisini içeren değerlendirme sonucu oluşan risk seviyelerine dayalıdır. Risk seviyelerinin hesaplanması için nicel olasılık ve etkiler referans alınabilir. Ürün riskleri risk seviyeleriyle ilişkili olarak sıralanır. Bu sıralama aktivitesi ürün risklerinin önceliklendirilmesi için ilk aşamadır.

iv. Gerçekleşme olasılığı

Gerçekleşme olasılığı sistemin test esnasında ortaya çıkan potansiyel problemlerinin olasılığı olarak yorumlanabilir. Başka bir deyişle bu, programlama dili ya da yazılımcının deneyimi gibi teknik bir riskten doğabilir. Bu aktivite olasılıkların hesaplanması için ürün risklerinin nedenlerinin izlenmesi aktivitesidir. Projede çalışanlar risklerin olasılığını belirtmek için en etkili kişilerdir.

Teknik riskleri içeren faktörler aşağıda belirtilmiştir;

- Teknolojinin karmaşıklığı
- Geliştirme yapan organizasyonun coğrafi dağılımı
- Yeni geliştirmeye karşı geçmişte kalan güncellemeler
- Zaman, kaynak ve yönetim baskısı
- Personelde yeterli bilgi ve deneyimin eksikliği
- Ürün kalitesine projenin erken dönemlerinde dikkat edilmemesi
- Arayüz ve entegrasyon maddeleri

v. Etkisi

Oluşma etkisi kullanıcılar, müşteriler ya da diğer paydaşlar üzerindeki etkinin şiddeti olarak yorumlanabilir. Diğer bir deyişle problemten kaynaklanan

ekonomik kayıp ya da problemden etkilenen kullanıcı sayısı gibi iş risklerinden doğabilir. Kullanıcılar risklerin olasılığını belirtmek için en etkili kişilerdir.

İş risklerini içeren faktörler aşağıda belirtilmiştir;

- Etkilenen özelliklerin kullanım sıklığı
- İmajın zarar görmesi
- İşin potansiyel kayıpları
- Finansal, ekolojik ya da sosyal kayıplar ve sorumluluk
- Mantıklı çözümlerin eksikliği
- Dış dünyada hataların görünürlüğü

vi. Nicelik ve nitelik

Testçiler risk seviyesini nicel ya da nitel olarak saptayabilirler. Eğer olasılık ya da etki nicel olarak anlaşılabilirse maruz kalınan durumun maliyeti iki değer kullanılarak hesaplanabilir. Bu belirli bir risk maddesiyle ilişkili beklenen bir kayıptır. Çoğu zaman oluşma olasılığı ve maruz kalınan etki tam olarak nicel değildir.

Sonuç olarak risk seviyeleri sadece nitel olarak anlaşılabilir. Nicel değerlendirmede risk, sınıflar içinde derece olarak tanımlanır. Olasılık çok yüksek, yüksek, orta ya da düşük olarak söylenebilir ancak kimse yüzde oranı veremeyecektir. Değerlendirmenin kesinliği insanların değerlendirme deneyimine ve tercihen farklı insanların bağımsız değerlendirme yapmasına bağlıdır. Nitel yaklaşım nicel yaklaşımın yanında alt seviyede görülmemelidir aslında nicel yaklaşım uygunsuz olarak kullanıldığı zaman riskin anlayan ve riski yöneten paydaşları yanlış yönlendirebilir. Olasılık ve etkinin nitel dereceleri düşük, orta ve yüksek olacak kadar basittir. Şekil 5.2'deki matris, ürün risklerini seviyelerine göre izlemek ve kayıt etmek için kullanılabilir. Riske dayalı test yaklaşımı için ürün risk matrisi kullanmak test hakkında yeterli bilgisi olmayan paydaşlarla iletişim açısından kolay bir yöntemdir.

| | | | | |
|------------|--------|--------|------|--------|
| Olasılık → | Yüksek | | | |
| | Orta | | | |
| | Düşük | | | |
| | | Düşük | Orta | Yüksek |
| | | Etki → | | |

Şekil 5.2 Ürün risk matrisi örneği

vii. Karar verme süreci

Risk analizi, kapsamlı ve istatistiksel olarak geçerli risk verilerine dayandırılana kadar risk analizinin nitel ya da nicel olduğuna bakılmadan olasılık ve etki faktörlerine göre değerlendirilir. Bu, çalışanların bu faktörlerle ilgili kişisel algı ve düşüncelerinin risk seviyesinin hesaplanmasında etkisi olduğu anlamına gelir. Proje yöneticilerinin, programcılarının, kullanıcıların, analistlerin, mimarların ve testçilerin farklı bakış açıları vardır bu yüzden her risk maddesi için risk seviyesinde farklı görüşler olması mümkündür. Risk analizi sürecince uzlaşmacı bir yol izlenmelidir. Test içeren risk azaltma aktiviteleri için sadece geçerli ve kabul edilen risk seviyeleri rehber olarak kullanılabilir.

Risk seviyesinin tanımlanması için toplu verilen kararlar bireysel alınan kararlardan daha doğru ve sağlıklıdır. Ürün risk analizi yapılırken toplu verilen kararlar aşağıdaki faydaları içerir.

- Bir grup bilgi paylaşımının etkili olması ve bilgi tiplerinin toplanması açısından bir kişiden daha etkili olur,
- Bir grup içinde fikir paylaşımı yapmak kişisel önyargıların oluşması ve önlenmesi için iyi bir yoldur,
- Karar verme sürecine yapılan katılım kişisel sözlerin tutulmasına teşvik eder,

- Bireysel fikirlerin genel karar ile nasıl ilişkili olabileceğinin anlaşılmasını sağlar ve kişisel fikirlerin öneminin görülmesinde fayda sağlar.

c) Risk azaltma aktiviteleri

Risk tanımlanıp analiz edildikten sonra ortaya çıkan risk faktörünü ele almanın dört yolu vardır;

- Önleyici ölçütler kullanarak riskin oluşma olasılığının ve etkisinin azaltılması,
- Eğer risk gerçekleşirse etkisini azaltmak için acil durum planlarının oluşturulması,
- Risk faktörünün ele alınması için ayrı bir ekip oluşturulması,
- Hiçbirşey yapılmaması ancak problemin oluşup oluşmadığına bakılması.

Bu seçeneklerden birini seçmek her bir seçeneğin faydalarına ve fırsatlarına göre, ilişkili oluşabilecek risklere ve maliyetlere göre değerlendirilebilir. Oluşabilecek ürün risklerinin yok sayılması dışında farklı seçenekler de vardır ancak risklerin oluşmayacağını düşünmek en büyük risktir. Bununla birlikte tüm riskleri öngörerek hiç risk faktörü içermeyen bir proje ortaya çıkarmak neredeyse imkansızdır. Risk yönetiminin özü, ürün kalite güvence kriterlerine ulaşmak ile bu kriterlerin karşılanma oranlarını maksimuma çıkarmak arasında makul bir denge kurmaktır. Kabul edilen risk durumuna bağlı olarak, birinin riskin hangi seviyede olduğuyla ilgili bilinçli bir karar vermesi gerekir.

i. Test aktiviteleri ile risklerin azaltılması

Riske dayalı teste, risk tanımlanması, risk analizi ve risk azaltma aktivitelerinin belirlenmesi ana test planının ve diğer test planlarının bir parçasıdır. Her risk maddesiyle ilgili olan risk seviyesi, her risk için harcanan test eforuna bakılarak hesaplanır.

Ürün riskleri göz önünde bulundurularak test etmek bu riskleri azaltmanın bir yoludur. Hataların bulunması kapsamında testçiler hataların farkında olunmasını sağlayarak ve sürümden önce hatalarla uğraşma fırsatı yakalayarak riski azaltabilirler. Benzer durumlar altında sistemin doğru çalışmasına bakarak riskin azaltıldığından emin olunur. Aynı zamanda ürün riskleri statik test aktiviteleri ile de azaltılabilir. Örneğin gereksinimler iyi

yazılmamışsa gözden geçirme aktivitesi riski azaltmak açısından daha etkili bir yöntem olacaktır.

ii. Önceliklendirme

Risk seviyesi aynı zamanda testlerin önceliklendirilmesi için de kullanılabilir. Bazı durumlarda tüm yüksek riskli testler daha düşük riskli testlerden önce uygulanır (Derinlik öncelikli test). Bazı durumlarda ise tanımlanmış riskler kullanarak, risk ağırlıklandırılması yapılması için test kümeleri ortaya çıkarılır. Test esnasında en az bir kere her riskin kapsandığından emin olunur (Sığ öncelikli test).

Risklerin önceliklendirilmesi, tasarımın, takvimin ve test uygulamalarının anahtarıdır. Önceliklendirme için olan ihtimaller yazılım geliştirme takvimine bağlı olabilir. Test uygulamalarının öncelik sırası çıkarılabilir. Eğer alt sistemler ve bileşenler farklı bir sırada geliştiriliyorsa, bu test takvimini ve test uygulamasının geçerliliğini önemli ölçüde etkiler. Bu da test aktivitesinin neden projenin erken aşamalarında olması gerektiğini anlatan bir diğer sebeptir. Yalnızca testte edinilen bilgilerin önceliklendirilmesi uygun olduğu zaman, test yöneticisi, proje yöneticisi bunu analiz edebilir ve gelişmiş, bütünleşik, test edilebilir risk modelini sağlayan en verimli planı belirlerler.

Risk odaklı test; testçiye kalan risk seviyesi açısından yönetime raporlama yapmaya yardım ederken, yönetimin ise testin uzamasına veya kalan riskin kullanıcı, müşteri, yardım masası, teknik destek veya işlemsel personele transfer edilmesine karar verilmesinde yardımcı olur.

iii. Test için risk azaltma yöntemleri

Testçiler tarafından önceliklendirmeye ek olarak kullanılan diğer risk azaltma aktiviteleri aşağıdadır.

- Uygun test tasarım tekniğinin belirlenmesi,
- Denetleme ve gözden geçirilmenin düzenlenmesi,
- Test tasarımındaki gözden geçirilmenin düzenlenmesi,
- Tecrübeli personelin kullanılması,
- Tekrar edilecek test seviyelerinin belirlenmesi,
- En uygun regresyon test senaryosunun belirlenmesi.

Test yaklaşımı, gösterilmiş test eforu ile gerekli riskleri kapsama arasında optimal dengeyi bulmaya yönelik olmalıdır. Hataların test esnasında bulunup doğrulanması, ürün ortaya çıktıktan sonra hatanın bulunup düzeltilmesinden daha az maliyetli olacaktır. Test maliyetleri ve test faydaları arasında kesin bir optimal nicelik ilişkisi olmamasına rağmen, test yöneticileri bu genel prensibi her zaman kişisel kararlarında akıllarında tutmalıdır.

iv. Değerlendirme

Genelde, test için risk azaltma yaklaşımı geliştirilip, paydaşlara sunulduğunda, çok istekli olurlar. Tüm teknik detayları anlayamazlar ancak test yönetimine karşı güven kazanırlar. Bununla birlikte, tanımlanmış test yaklaşımı için gerekli olan kaynakları gördüklerinde bu değişebilir. Genelde bunun pahalı olduğu düşünülür. Derinlemesine bir değerlendirme yapmak paydaşların bakış açısını değiştirmek için önemlidir. Paydaşlara, doğru değerlendirmeler ile yapılan risk azaltma yaklaşımının ürün ortaya çıktığında oluşacak olan maddi hasardan kaçınılması için doğru aktivite olduğunun anlatılması gerekir.

d) Risklerin izlenmesi

İzleme genellikle, sistemin durumun farkında olmak demektir. Proje uygulama açısından doğru ürün risklerinin tanımlandığından emin olunur ve planlanan etkilerin gerçekleştiği görülür. Risk azaltma aktivitelerinin ve risk seviyeleri ürün risk seviyelerinin geçerli seviyede azalıp azalmadığını takip ederek proje boyunca izlenir.

Temel risk tanımlaması ve analiz aktiviteleri yapılır ve risk azaltma aktivitelerinin yerine getirilmesi için hangi risk maddelerinin ne oranda azaltıldığı ölçülebilir. Bu test senaryolarının ve ürün riskleriyle ilgili bulunan hataların takibi ile yapılabilir. Testler uygulandıktan sonra ve hatalar bulunduktan sonra testçiler kalan ve artan risk seviyelerini denetleyebilirler. Bu değerlendirmenin çıktısı olarak test yöneticisi test kapsamının büyütülmesine, testin durulmasına ya da farklı test ölçütlerinin uygulanmasına karar verebilir.

Bu tip izleme risk odaklı testin temel aşamasıdır ve ürünün piyasaya sürülmesi için doğru zamanın hesaplanmasına yardımcı olur. Test raporu bilinen ve yeni ortaya çıkan ürün risklerini içerirken aynı zamanda sağlanan avantaj ve dezavantajları da kapsmalıdır. Çoğu yönetici risk odaklı raporları tercih eder

çünkü raporlar, ürünün durumunu ve test sürecinin görülmesi için daha net bir görüş açısı sağlar. Bu raporlar, proje içinde ürün risk değerlendirilmesinin yapılması için kaynaklanan erken dönemde elde edilmesine teşvik sağlar. Risk odaklı bilgilerin güvenilirliği, ürün risk değerlendirilmesinin kalitesine ve paydaşların projeye olan katkısına bağlıdır.

i. Ürün Risklerinin Gözden Geçirilmesi

Testçiler, tanımlanmış olan riskin kapsandığından emin olmak zorundadırlar. Testin ilerlemesi ile birlikte kapsam da izlenir. Çıkan risk seviyelerine ve bu risklerin azaltılması için gereken uygun kapsam seviyelerine karar vermek kolay bir iş değildir. Riskler değiştikçe bunlara bağlı testlerin değiştiğinden emin olmak da zordur.

Eğer çoklu test döngüleri varsa gelecekteki test döngüleri ürün risk değerlendirilmelerine göre güncellenmelidir. Ürün risk değerlendirmesi yeni ya da değişen ürün risklerine test sırasında bulunan hata yoğunluklu alanlara, çözümlenen hatalara uyum sağlayacak şekilde güncellenmelidir.

Her yeni ya da ek test döngüleri oluşturulacaksa her riskin girdi olacağı bir analiz planı kullanılmalıdır. Risk tanımlaması ve risk analizi projenin erken aşamalarında yapılması gereken bir görev değildir. Risk tanımlaması ve analizi aktiviteleri proje boyunca devam etmelidir. Böylece yeni ortaya çıkan ürün riskleri de kapsanır. Periyodik olarak projelerde ürün riskleri ve risk seviyeleri yeni bilgilere göre yeniden değerlendirilir.

ii. Raporlama

Risk odaklı testin en büyük avantajı, yönetime sistemin piyasaya hazır olup olmadığı konusunda kapsamlı bir bilgi sunmaktır. Bu test disiplininin en önemli amacıdır. Bu bilgiyi sağlamanın anahtarı her testi bir ürün riski ile referans vermektir. Bunun için bir “izlenebilirlik matrisine” ihtiyaç vardır .

Tablo 5.1 Ürün risk maddelerinin izlenebilirlik matrisi örneği

| | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | ... | Test n |
|----------------|--------|--------|--------|--------|--------|--------|--------|-----|--------|
| 1.Risk Maddesi | x | x | x | | | | | | |
| 2.Risk Maddesi | x | | | x | x | | x | | |
| 3.Risk Maddesi | x | | x | | | | | | |
| 4.Risk Maddesi | | | | | x | x | x | | |
| | | | | | | | | | |
| m.Risk Maddesi | | | | | | | | | |

Tüm testçiler belli bir riske ait test senaryosunu test ortamında geçirdiğinde, bu risk maddesi silinir. Ürün riskleri çoklu test seviyelerini boydan boya gösterir. İzlenebilirliğin kapsamı tüm test seviyelerini karşılamalıdır. Bu tip risk odaklı raporlama kolayca grafik formatında sunulabilir.

Risk odaklı test uygulamalarındaki tipik problem, projedeki test raporlarının ürün riskini azaltmaya odaklı olmayıp, uygulanmış test durumları ve hata sayılarına odaklı olmasıdır.

Tipik proje yöneticisi azaltılmış proje risklerinden ziyade ne kadar test durumunun çalıştırıldığı ile ilgilenir. Risk odaklı test, risk odaklı raporlamayı kapsamaktadır. Risk paydaşlar için raporları daha kolay ve anlaşılır hale getiren iş dilidir. Test raporu, testin en önemli çıktılarından biridir. Bu, bizim projenin bir sonraki gelişim aşamasına ilerlemek veya sistemi piyasaya sürmek için paydaşlara sunduğumuz bilgidir. Kim için test raporu yazdığınızı ve bu raporu onlar için nasıl faydalı hale getirileceği unutulmamalıdır. Proje sonrası geçmişe yönelik proje toplantılarında paydaşların test raporunun hangi kısmını beğendiklerini hangilerini anlamadıklarını sorulabilir. Test raporu ana çıktı yerine geçebilir. Risk odaklı raporlama kullanımı, doğru istikamette atılan büyük bir adımdır.

Yüksek seviyeli risk odaklı raporlama örneği Tablo 5.2 olarak verilmiştir. Bu örneklemede, risk değerlendirmesi ,ürün risk tanımları ve uygulama durumları ile belirtilmiştir. Sonrasında testler tanımlanır, tasarlanır ve ürün risklerine göre

belirtilir. Ürün riskleri ile testler izlenebilirlik matrisine göre kontrol edilir. Test uygulama safhası başlangıcında bütün ürün riskleri bu aşamada belirtilmiştir.

Tablo 5.2 Test uygulama durumu tablosu

| | Risk Seviyesi | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | Test 8 | ... | Test n |
|-----------------------|---------------|--------|--------|--------|--------|--------|--------|--------|--------|-----|--------|
| 1.Risk Maddesi | Kritik | ✓ | ✓ | ✓ | | | | | | | |
| 2.Risk Maddesi | Kritik | ✓ | | | x | x | | ? | | | |
| 3.Risk Maddesi | Yüksek | ✓ | | ✓ | | | | | | | |
| 4.Risk Maddesi | Orta | | | | | x | ✓ | ? | | | |
| ... | | | | | | | | | | | |
| m.Risk Maddesi | Düşük | | | | | | | | ✓ | | |

6. PRISMA(PRACTICAL RISK BASED TESTING) SÜRECİ

PRISMA yaklaşımı kullanılmadan önce PRISMA sürecinin tanımlanması önemlidir. Bu bölümde ürün risk değerlendirmesi PRISMA yaklaşımı kullanılarak ortaya çıkarılacaktır.

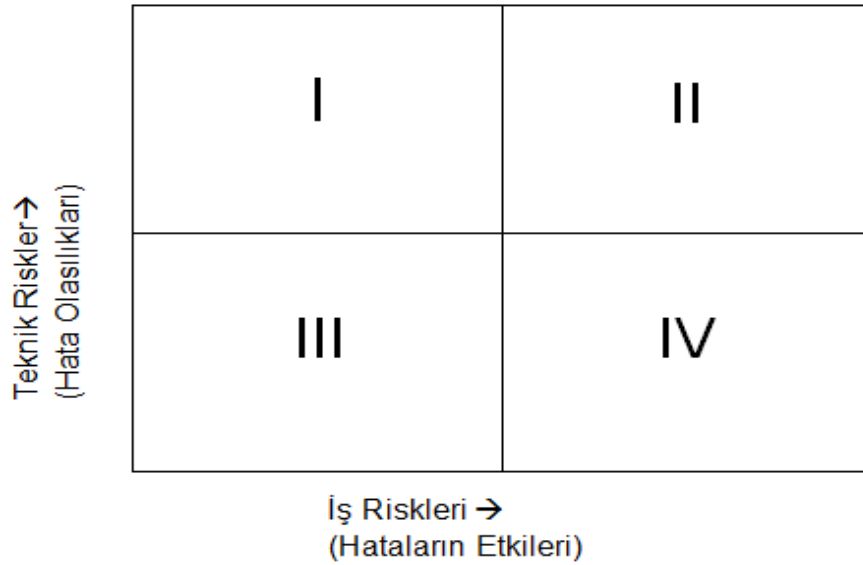
6.1 Sürece Genel Bakış

Bu bölümde PRISMA süreci ve süreç ayrıntıları genel anlamda anlatılacaktır.

6.1.1 Ürün risk matrisi

Prisma sürecindeki temel konu ürün risk matrisinin oluşturulmasıdır. Her ürün riski için bu tanımlanır. Bu riske bağlı oluşabilecek olan muhtemel hataların etkileri ve bu hataların oluşma olasılıkları hesaplanır.

Nümerik değerler bu etki ve olasılıklara atanarak bir ürün riski ürün risk matrisine yerleştirilir. Standart bir ürün risk matrisi dört alana ayrılır ve her bir alan farklı bir seviye ve tipteki riskleri kapsar. Farklı seviye ve tipteki riskler test planında belgelenmiş farklı bir test yaklaşımı anlamına gelmektedir. Ürün risk matrisi, projede uygulanmış tüm testlerin temeli olarak kullanılabilir.



Şekil 6.1 Ürün risk matrisi bölgeleri.

Ürün risk matrisi paydaşlara ve karar mekanizmasında bulunan yöneticilere ürün riskleri ile ilgili bir öngörü sağlar. Anlaşılabilirliği liste haline getirilmiş olan halinden çok daha nettir ve iletişim kolaylığı sağlar.

6.1.2 Farklı test seviyeleri

Geliştirme bölümü, genellikle risk odaklı test ve bileşen üzerinde PRISMA ya da bileşen entegrasyon testlerini uygular. Geliştirme testlerine gelene kadar, değişken veya karmaşık alanlarla alakalı yüksek risk alanları oluşacaktır. PRISMA, uygulanan testlerde işlevselliğin ticari kritikliğinden çok teknik risklere odaklı olmalıdır. Ticari olarak ve kullanıcı bakış açısından risk odaklı test ve PRISMA yaklaşımı sistem testlerinde ve kabul testlerinde uygulanabilir. Ticari açıdan teknik riskler daha az dikkate alınır. Sistemin gereksinimleri karşılama, kullanıcı işlevselliğinin doğruluğuna ve ticari faaliyetleri garanti altına almasına odaklanılır. PRISMA yaklaşımı uygulanırken sistemde ve test aşamasında odaklanılan kısım, gelişmekte olan birleşenlerin teknik sorunlarından çok ticari riskler üzerindedir. Çeşitli seviyelerde ürün risk matrisi uygulandığında öncelikler farklılık gösterir.

6.1.3 Risklerin tanımlanması : Gereksinimler ve beyin fırtınası

PRISMA bünyesinde; risk tanımlaması için iki ana teknik yöntem kullanılır: Beyin Fırtınası ve Gereksinimlere Dayalı Risk Tanımlaması. Ürün riskleri başta gereksinim dokümanlarında ve beyin fırtınası zamanlarında tanımlanır. Bu iki teknik bir bütündür biri işleme dayalı dokümana, diğeri takım egzersizinde insana bağlıdır. Bu iki bakış açısı birleştirilerek kusursuz ürün gözden geçirme riskleri ortaya çıkar

Gereksinim odaklı risk tanımları bazen iyi algılanmamasına ve yeterli risk tanımlarını oluşturulmamasına rağmen, temel riskleri oluşturduğundan kullanışlıdır. Diğer tekniklere göre test yöneticisinin farklı kaynaklara ihtiyaç duymamasından dolayı büyük avantaj sağlar. Test yöneticisi, proje aşamalarını planlarken, birincil küme olarak proje risklerini belirler. Birincil kümelere bir de , beyin fırtınası gibi tamamlayıcı tekniklerin liste olarak uygulanmasını sağlamaktır. PRISMA beyin fırtınası oturumları , test yöneticisi tarafından tanımlanan gereksinim odaklı ürün risklerinin artırılması ve doğrulanması için kullanılır. Özellikle en kötü senaryolardaki beyin fırtınası, bu amaç için oldukça başarılıdır. Çoğu gereksinim

odaklı uygulamalarda ürün riskleri tanımlandığından beri, beyin fırtınası çalışmasına daha fazla odaklanılır Bu, risk tanımlarını daha verimli yapar. Sonuç olarak , beyin fırtınası, tanımlı ürün risklerini günceller ve bir sonraki aşamadaki işlemler için kullanılmasını sağlar.

6.1.4 Risk analizi: Etki ve olasılık faktörleri

PRISMA yaklaşımı ile kullanılan başka bir önemli kavram ise etki ve olasılıkların hesaplanması için kullanılan faktörlerdir.Son kullanıcılara herhangi bir ürün riskinin önemli olup olmadığını sorarsanız bir çok ürün riskini önemli olarak etiketler. O yüzden bu soru belirlenen ürün riskinin olma olasılığı nedir ve olursa oluşacak olan etkileri nelerdir şeklinde sorulmalıdır.

6.2 Başlangıç Aşaması

Başlangıç aşamasının amacı kuralların tanımlanması ve şablonların oluşturulmasıdır.Başlangıç aşaması aşağıdaki görevleri içerir.

- Standart sürecin tanımlanması,
- Faktörlerin belirlenmesi,
- Her faktör için ağırlıklandırma yapılması,
- Puanlama aralıklarının belirlenmesi,
- Kuralların tanımlanması.

6.2.1 Standart sürecin tanımlanması

PRISMA yaklaşımını uygularken organizasyon içinde kullanılacak olan standart süreç bu aşamada tanımlanır. Bu organizasyon içinde farklı durumlar için uygulanabilecek süreç varyasyonlarını da içerir.Organizasyon içinde bu süreç belgelenmeli ve el kitabı olarak dağıtılmalıdır. Süreç belgelendikten sonra iyileştirilmesi ve denetimi daha kolay olacaktır.

Ürün risk matrisi formatı bu kısımda belirlenir. Şekil 6.1 içinde gösterildiği gibi Normalde matriste dört çeyrek vardır ancak isteğe bağlı olarak beş alan da kullanılabilir.

6.2.2 Faktörlerin belirlenmesi

Başlangıç aşamasında ilgili olasılık ve etki faktörleri belirlenir. Belirlenen faktörler tüm paydaşlar tarafından anlaşılacak şekilde olmalıdır. Bu aşama PRISMA takım liderinin görevleri arasındadır ancak farklı faktörler için test lideri, geliştirme lideri ya da diğer paydaşlarla beyin fırtınası yapılabilir. Grubun deneyimine göre bir faktörün test için neden önemli olduğu konuşulmalıdır.

6.2.3 Her faktör için ağırlıklandırma yapılması

Bir faktörü referans alarak diğer faktörün vurgulanması yöntemi kullanılabilir, yani daha etkili olacak bir faktör daha az etkili bir faktörün iki katı şeklinde atanabilir. Ancak ağırlık değerleri ataması maddelerin ürün risk matrisi üzerindeki pozisyonunu doğrudan etkileyeceği için daha hassas davranılması yararlı olacaktır.

6.2.4 Puanlama aralıklarının belirlenmesi

Risklerin değerlendirilmesi tanımlanmış ürün riskleri için faktörlerin puanlanması şeklinde olur. Burada likert ölçeği* kuralı uygulanır. Puanlama aralığı kritiklik durumuna göre 1-3 aralığında düşük(1), orta(2), yüksek(3) şeklinde, 1-5 aralığı şeklinde ya da 1-9 aralığı şeklinde belirlenebilir. En sağlıklı 1-5 değer aralığının kullanılmasıdır. Düşük(1-2), orta(3-4), yüksek(5).

6.2.5 Kuralların belirlenmesi

Son olarak puanlama ve oranlama için kurallar belirlenir. Dağılım kuralı, paydaşların puanlama aralığındaki tüm değerleri bir faktör için kullanması anlamına gelir. Yani bir paydaş aynı faktör için tüm risk maddelerine tek değeri atayamaz. Puan kuralı ise her paydaşın kendisine atanan her faktör için her risk maddesini puanlaması anlamına gelir. Puanlama yapmadığı zaman uyarılır.

Likert Ölçeği: Likert ölçeği çok düşüğe kadar değişen yanıt seçeneklerini kullanarak fikirleri ölçer. Basit bir "yüksek/ düşük" seçeneği yerine, fikir derecelerinin ayrıntılı ortaya çıkmasını sağlar. İstenilen durum ve istenmeyen durumun ortaya konmasını amaçlayan olumlu ve olumsuz durumların eşit sayıda madde ile ifade edildiği ölçek çeşididir, çoğunlukla 5 seçeneklidirler ve puanlamaları 1'den 5'e yapılır.

6.3 Planlama Aşaması

Planlama aşamasının amacı , yönetilen ve yüksek kaliteli ürün risk değerlendirmesinin gerçekleştirilebildiğinden emin olmaktır.

Planlama aşamasında, kapsam, paydaşlar ve ürün riskleri tespit edilir ve tanımlanır.Yapısal bir planlama aşaması başarının anahtarıdır.Planlama aşaması aslen test yöneticisi tarafından yürütülür ve aşağıdaki görevleri içerir:

- Kapsamın tanımlanması,
- Uyum faktörleri ve kuralların belirlenmesi,
- Belgelerin bir araya getirilmesi,
- Ürün risk maddelerinin tanımlanması,
- Paydaşların tanımlanması ve seçilmesi,
- Faktörlerin paydaşlara ayrılması.

6.3.1 Kapsamın tanımlanması

PRISMA ürününün risk değerlendirme alanının çok iyi tanımlanması gerekir.Ürünün açık ve kesin tanımı, ürünün payı, veya çalışılması gereken gereksinimlerin sırası ekipteki herkes tarafından uygun ve anlaşılır olmalıdır.Tercihen, ekibin geliştirme mimarisi ile ilgili dokümanlara erişimi olabilmelidir.Mimarinin anlaşılması teknik risklerin tanımlanmasının desteklenmesine yardımcı olacaktır.

Paydaşların, kapsamı kafalarında netleştirmeleri için, onlara ilk toplantıda fırsat verilmesi gerekmektedir.Net olarak tanımlanmış bir kapsam PRISMA ürününün risk değerlendirme aşaması sırasında, ürünün ekip tarafından yanlış anlaşılmasını engeller.

6.3.2 Uyum faktörleri ve kuralların belirlenmesi

Başlangıç adımında , organizasyonla genel anlamda ilişkili olan olasılık ve etki faktörleri tanımlanır.Planlama adımında, bu faktörler test edilmekte olan sistem ile ilişkili olup olmadığına karar verilmesi için tekrar gözden geçirilir.Bazen bir faktörün sistemle ilişkili olduğu düşünülebilir fakat faktörle ilişkili henüz herhangi bir bilgi yoksa bu aşamada o faktör kullanılmaz.Örneğin; ekibin deneyimine bağlı teknik

risk, ekip tanımlanmadığı zaman değerlendirilemez.Genellikle, başlangıç adımında faktörlerin listesi tanımlanır ve planlama adımında faktörler liste dışına çıkartılır. Prensip olarak ise sadece test edilmekte olan sistemle ilişkili ve organizasyonda daha önce hiç başvurulmamış faktörleri de eklemek mümkündür.

Faktörlere ek olarak, tüm kuralların değerleri ve değişken faktörlerin ağırlıkları ayarlanmalıdır.Kural ve faktörlerin öncelik sıralamasını belirlemek için geçmişteki projelerden geri bildirimler alındığında , sıralama daha güvenilir olur ve daha az yenileme gerektirir.

6.3.3 Belgelerin bir araya getirilmesi

Planlama aşamasında kullanılacak belgelerin sırası katılımcılar tarafından kararlaştırılır ve toplanır.Bu belgeler genellikle aynıdır ya da en azından test dokümanları ile yakından ilgilidir.Uygun olan girdi belgelerinin çeşidi büyük ölçüde ürün risk değerlendirmesinde uygulanan test seviyesine dayanır.PRISMA yaklaşımında, takım lideri ihtiyaç duyulan kalite seviyesindeki girdilerinin bu süreç için gerekli bilgi içerip içermediğini kontrol etmelidir.Belgeler son belge veya onaylı belge olmak zorunda değildir; öte yandan ekibin ürün risk değerlendirmesini etkili kullanabilmek için istikrarlı olmaları gerekir.Risk değerlendirmesi sırasında adı geçen herhangi bir belgenin (gereksinim dokümanı veya yazılım mimarisi dokümanı gibi) belge sahibine rapor edilmesi gerekir.Eğer girdi belgesi gerekli kalite seviyesinde değil ise , ürün risk değerlendirmesine devam edilip edilmeyeceği organizasyon tarafından düşünülmesi gerekir.

PRISMA sürecindeki opsiyonel adım sayılan “genişletilmiş risk tanımı” adımı bu aşamada zorunlu hale gelmiş olabilir.Böyle bir durumda ürün risk maddeleri beyin fırtınasıyla tanımlanacaktır.Girdi olarak kullanılan belgeler gereksinim dokümanlarının ya da tasarım dokümanlarının bir çeşididir(ürün değerlendirme,test seviyesi, entegrasyon aşamasındaki bir ürün risk değerlendirmesi, bileşen test seviyesi gibi)

6.3.4 Ürün risk maddelerinin tanımlanması

Gereksinimlere ve olası girdilere dayalı ürün riskleri test yöneticisi tarafından tanımlanır. Bu liste beyin fırtınası yapılarak çoğaltılabilir. Çevik yaklaşımı

projelerde gereksinim kümeleri bir sonraki projeler için belgelenecek kullanıcı senaryosu olarak kullanılabilir. Eğer varsayım yapılacak olursa daha fazla araştırma, belge sahibi ve paydaşlarla görüşme gibi unsurlara ihtiyaç duyulabilir.

Genellikle her gereksinimi ayrı bir ürün risk maddesi olarak düşünmek daha sonra gereksinim madde numaraları çoğalmasında durumunda karışıklık yaratacağından sağlıklı değildir. Bir ilke olarak süreci yürütebilmek için 35 ila 40'dan fazla tanımlanmış ürün risk maddesinin olmaması gerekir. Bu genellikle girdi belgesi olarak belirtilen belgelerin birimlere ayrılması anlamına gelir. Proje ekibi daha sonraki aşamada girdi olarak yüksek seviye ürün risk değerlendirmesini kullanarak ve bileşen ve altyapı aracılığıyla başlangıç gereksinimlerini değerlendirerek test stratejisine dayalı daha ayrı, detaylı bir risk değerlendirmesine karar verebilir.

Web projelerinde ya da arayüz ağırlıklı projelerde her bir arayüz bir risk maddesi olarak tanımlanabilir. Böylece test edilecek arayüzler üzerinden risk önceliklendirmesi yapılabilir ve arayüzler birer test maddesi olarak atanabilir.

Tespit edilen ürün risk maddeleri tek olarak tanımlanmalı (etiketlenmeli) ve paydaşlar için anlaşılabilir bir yolla ifade edilmelidir. Referans numarası bir rakam olabilir ama genellikle bir kod ya da kısaltma proje için daha anlamlı olur. Her bir risk maddesi için bir işaret, girdi belgelerinin ilgili parçaları olarak sağlanmalıdır. Bu, daha sonra paydaşlar geçmiş bilgilere ihtiyaç duyduklarında ürün risk maddelerinin takibine yardımcı olacaktır.

Bir maddeyi tanımlamada her risk maddesinin bir satırlık tanımı da yardımcı olabilir. Paydaşlara takibini yaptıkları risk maddesini daha kolay puanlamalarını sağlar.

Bazen ürün risk maddelerinin saptamasının test edilmekte olan sistemin fonksiyonelliğine öyle odaklanmıştır ki arayüzler ve veri transfer özellikleri gözden kaçırılır. Proje temsilcileri, teknik ve mimari bilgilerin bir sistemden diğerine geçişinden haberdar olmayabilir. Bunun nedeni, mimari dokümanların ve tasarım dokümanlarının (özellikle entegrasyon test seviyesi ya da bileşen entegrasyon test seviyesinde) ürün risk maddelerini tanımlarken göz önünde bulundurulmasının tavsiye edilmesidir.

6.3.5 Paydaşların tanımlanması ve seçilmesi

Normalde test yöneticisi PRISMA sürecinden sorumlu olsa da bütün PRISMA süreçleri ekip odaklıdır. Ekip odaklı olmasının amacı süreç içerisine çeşitli deneyimler ve bakış açıları getirmeyi bir takım uygulaması yapmaktır. Bir ekip için en uygun olan 6 ila 4 katılımcı arasındadır, fakat minimum katılımcı rakamı ürün tarafından etkilenen alanların sayısına göre belirlenebilir. Ekibi yaklaşık 10 kişiyle sınırlandırmak faydalıdır. Genelde sonuca ulaşan ekip, proje ekibinden daha geniş bir yapıya sahip olur. Tüm ekiplerin de dahil olduğu toplantılar aşağıdaki yararları sağlar:

- Eğer tüm ekipler katılım sağlarsa, kararlara bağlılık sağlanır ve sonuçların kabul ettirilmesi kolaylaştırılır;
- Ekipler katılım sağlamazsa süreç ve çıktılar sağlıklı olmaz;
- Ekipler arasındaki iletişim yükselir, ekip PRISMA yaklaşımını benimsedikten sonra birbirlerinin düşüncesine, amaçlarına, hareketlerine karşı anlayış kazanır;
- Ekip olarak çalışmak belirsiz ve riskli durumlarda, bireysel işe kıyasla grup çalışmasının yararını kullanır.

Bu adımda ürün risk değerlendirmesinde bulunan paydaşlar belirlenir ve seçilir. Genellikle çeşitli rolleri temsil eden paydaşlar seçilir. Paydaşlar, proje yöneticilerinden, geliştiricilerden, yazılım mimarlarından, pazarlamacılardan, tüketicilerden, iş yöneticilerinden seçilebilir. Paydaşlar ürünü kuran, kullanan ve ondan etkilenenlerdir. Paydaşları tanımlarken şunu akılda tutmak gerekir ki paydaş tanımı ürünü kullanacak olanlarla sınırlandırılmamalıdır. Kar amacıyla paydaşların riskleri tanımlamaya yardımcı olma gibi bir sorumluluğu vardır. Prensipte olarak, test etme paydaşların yararına yapılır.

Paydaşları tanımlarken test yöneticisi departmanlara önem vermeyebilir. Sponsorla veya ürün sahibiyle yapılan başlangıç toplantısında, paydaşlar genellikle tanımlanır. Proje büyüklüğüne, ürünün kritikliğine, gelişip gelişmediğine dayanarak ek bir paydaş analizi gerekebilir. Gereksinim dokümanından çıkarım yapılan paydaş listesi, mühendislik sürecinde yetersiz olduğu tahmin edilen ek paydaşları tanımlamada kullanılabilir.

Paydaşların tanımı ve seçimi, “Unutulan paydaş, tanımlanmayan ürün riskleri anlamına gelir” şeklinde açıklanan önemli bir aktivitedir.Çok iyi bir test süreci uygulanabilir fakat canlıya çıktığında, en küçük hatada, tanımlanmayan her ürün riski fark edilecektir. Bir proje için, tüm paydaşların tanımlanmasına yardımcı olmak için olabilecek paydaşların bir listesi oluşturulur.

6.3.6 Faktörlerin paydaşlara atanması

Bu aşamada puanlama yapacakları faktörler paydaşlara atanır. Paydaşlara, her faktör ile ilgili puanlama yapmak isteyip istemedikleri sorulabilir ancak pratikte faktörlerin, paydaşların ilgi alanlarına ve rollerine göre dağıtılmaları daha kullanışlı bir yöntem olacaktır. Bu ayrıca paydaşlar için de daha motive edici olur. Böylece bilgi sahibi olmadıkları faktörler ile ilgili ayrıca çaba sarfetmek zorunda kalmazlar. Genelde etki faktörleri iş ile ilgili paydaşlara atanırken, olasılık faktörleri teknik taraftaki paydaşlara atanır.

Bu aşamanın sonucunda ürün risk değerlendirmesi için bir plan ortaya çıkmalıdır. Bu plan, süreç tanımını, parametreleri, analiz edilecek ürün risklerinin kümelerini, seçilmiş olan paydaşları ve onların rollerini içermelidir.

6.4 Başlangıç Toplantısı Aşaması

Başlangıç toplantısının amacı, yapılacak olan ürün risk değerlendirmesi hakkında katılımcıların bilgilendirmek ve gerekli sözleri almaktır.

Başlangıç toplantısı sırasında test yöneticisi tüm katılımcılara ürün değerlendirme işlemini açıklar.Başlangıç toplantısından sonra süreç , bireysel roller ve yapılması gereken aktiviteler tüm katılımcılarda net olmalıdır.Bu toplantının bir başka amacı herkesi ürün risk değerlendirmesine odaklamak ve herkesin katılımı için gerekli olan zamana bağlı kalmalarını sağlamaktır. Başlangıç toplantısı, PRISMA sürecine aşına olmayan paydaşlar için de bir eğitim niteliği taşır.Bu adım opsiyonel olmasına rağmen, başlangıç toplantısının ürün risk değerlendirmesine etkisi çok büyüktür.

Başlangıç toplantısının başarılı olabilmesi için test yöneticisi tarafından bir hazırlık yapılması gerekir. Amaçlar ve yapılacak işler tanımlanmalıdır. Başlangıç toplantısı aşaması aşağıdaki görevleri içerir:

- Sürecin açıklanması,
- Ürünün açıklanması,
- Anlaşmaya varılması.

6.4.1 Sürecin açıklanması

Başlangıç safhası, risk odaklı test yaklaşımının anlaşılmasını, ürün risk matrisinin değerlendirmesini sağlar. Ortak görüşe varılması daha iyi motivasyon sağlanmasına ve uygulamanın daha sağlıklı ortaya çıkmasına sebep olur. Sürecin kullanılabilirliği ve faydaları hakkındaki tartışma süreçten sonra değil başlangıç toplantısında yer almalıdır. Ürün risk değerlendirmesine dayalı test sürecinin işletilmesinin önemi, özellikle bu toplantıda açıklanmalıdır. Bireysel hazırlığın yapılması, kullanılacak araçların tanımının yapılması ve uygulanabilir son teslim tarihini konuşmak bu toplantıda önemlidir. Katılımcılara puanlama yapacakları faktörler detaylı bir şekilde açıklanır. Faktörlerin anlamları ve yapılacak puanlama için(kuralları ve mantığı) değer kümeleri paydaşlar tarafından yorumlanmalıdır.

Test yöneticisi sürecin geri kalanı için genel bakış açısını anlatmalıdır. Paydaşlar bu toplantıdan ne beklentileri gerektiğini ve değerlendirmenin sonuçlarını anlamalıdır. Test yöneticisi katılımcılara süreçteki rollerinin ne olduğunu ve proje içindeki katkılarının test yaklaşımını nasıl etkilediğini anlatmalıdır.

6.4.2 Ürünün açıklanması

Test yöneticisi tarafından bir önceki adımda yapılandırılan ürün risk maddelerinin listesi genellikle detaylı açıklama gerektirir. Gerekli olan açıklama seviyesi katılımcıların proje ve ürüne olan yakınlık derecelerine göre belirlenir. Genellikle ürün veya proje hakkında bilgiye sahip olan katılımcılar seçilir fakat tüm katılımcılar bu seviyede bilgiye sahip olamazlar. Projenin kısa bir tekrarı, amaçları, gereksinim dokümanı ve sistem mimarisi, ürün bilgisini katılımcılara gerekli olan seviyede sağlar. Paydaşlardan güvenilir sonuçlar elde edebilmek için ortak bir anlayış belirlenmeli ve ürün risk maddeleri tanımlanmış olmalıdır.

6.4.3 Anlaşmaya varılması

Başlangıç toplantısının sonunda, paydaşların sürece katılım sağlayacaklarının ve beklentileri yerine getireceklerinin sözü alınmalıdır. Rollerin anlaşılması , takvime uyulması son tarihlere uyulması tamamen paydaşların sorumluluğundadır. Test yöneticisi bu yaklaşımın benimsenmesini sağlamalıdır. Rollerin toplantıda açık bir şekilde dile getirilmesi, sorumluluklara uyulmasını ve sorumlulukların uygulanmasını kolaylaştırır.

Bu aşamanın sonucunda ürün risk değerlendirme sürecinde ve üründe özellikle risk maddeleri hakkında tüm paydaşlar üzerinde ortak bir görüş oluşması gerekir.

6.5 Ayrıntılı Risk Tanımlaması

Risk tanımlaması yapmanın amacı;

- Ürün risk kümesini genişletmek,
- Başlangıçtaki ürün risk kümesini doğrulamak,
- Gereksinimler ve ürün riskleri arasındaki uyumsuzlukları tanımlamak.

Tek başına gereksinimler bazen ürün risklerini tanımlamak için yetersiz olabilir. Planlama aşamasında gereksinimler başlangıçtaki ürün risklerinin kümesini tanımlamakta kullanılır. Yeni bir sistem yada var olan bir sisteminin ürün risklerini tanımlamak sadece gereksinimler üzerinden yapılamayabilir çünkü ürün risk değerlendirme başlangıcında tüm gereksinimler bilinmeyebilir.

Bu adımda beyin fırtınası oturumları, ürün risklerinin başlangıç kümesini doğrulamak ve genişletmek için kullanılır. Beyin fırtınası oturumlarında, her paydaştan bu yeni sistemle ilgili 2 ya da 3 en kötü senaryoyu sunmaları istenir. Öngörülen risklerin başlangıçtaki risk kümesi ile örtüşüp örtmeyeceği gözden geçirilir. Gereksinimleri ürün risklerine bağlayarak, bu yeni tanımlanmış ürün risklerinin eksik olup olmadığı ya da birbirleriyle çatışıp çatışmadığını anlaşılabilir. Yüksek öncelikli ürün riskleri ile ilgili gereksinimler eksik yada belirsiz olduğunda bu gereksinimler üzerinden ilerlenmelidir.

Beyin fırtınası oturumunda ana sorunlardan biri nerede durulacağını bilinmesidir. “Ne zaman ürün listesi yeteri kadar iyi olur” sorusuna cevap verebilmek önemlidir.

Planlama aşamasında temel olarak gereksinimleri alarak ürün risklerinin başlangıç kümesini oluşturmak sonrasında en kötü senaryoyu düşünerek beyin fırtınası oturumunu sınırlandırmak PRISMA sürecinde sorununun üstesinden gelmenin en pratik yoludur.

Bu adım, geliştirmenin devam ettiği sistemin kritikliğine dayanır ve opsiyonel bir adımdır ancak aynı zamanda gereksinim anlaşmasının kalitesine ve geçerliliğine de bağlıdır. Kritik bir sistem için ürün risklerinin listesi olabildiğince hızlı tamamlanmalıdır ve böyle durumlarda bu adım opsiyonel değil zorunlu olmalıdır. Gereksinimlerin kalitesi yeterli değilse tanımlanan ürün risklerinin listesini genişletmek için ek ürün risklerinin tanımlanacağı bir beyin fırtınası oturumuna ihtiyaç duyulabilir.

Ayrıntılı risk tanımlama aşamasının sonucu olarak, tanımlanmış ürün riskleri kümesi geliştirilir. Bu sürecin yan ürünü olarak, gereksinim hataları dokümante edilir ve ilgili kişilere sunulur, böylece tüm proje için yararlı olacak güçlü bir gereksinim dokümanı ortaya çıkmış olur.

6.6 Bireysel Hazırlık Aşaması

Bireysel hazırlık aşamasının amacı;

- Her ürün riski için olasılık ve etki faktörlerini analiz ederek ürün risk analiz sürecine bağımsız girdiler sağlamak,
- Her ürün risk maddesi için atanmış faktörleri puanlamak ve test yöneticisine sunmak.

Bireysel hazırlık aşamasında, katılımcılar her kayıtlı risk maddesinin etki faktörlerini ve olasılıklarını puanlayarak kaydederler. Katılımcılar, risk maddelerini faktörlerle eşleştirerek algılanan riske en çok uyan değeri vererek puanlamayı yaparlar. Puanlama sürecinde katılımcıların diğer katılımcılardan etkilenmemesi için bu adım mutlaka bağımsız yapılmalıdır. Bu adım kağıt üzerinde yapılabilir fakat genellikle katılımcılara çeşitli puanlama kuralları içeren bir Excel dosyası gönderilir.

Bireysel hazırlık aşaması aşağıdaki görevleri içerir:

- Her risk maddesi için faktörlerin puanlanması,
- Tahminlerin dokümantasyonu,
- Kişisel gözden geçirme,
- Test yönetimi.

6.6.1 Her risk maddesi için faktörlerin puanlanması

Bireysel hazırlık yapmanın ve takımdan bağımsız puanlama yapmanın iki avantajı vardır. En önemlisi paydaşlara risk faktörlerinin algılanmasında bağımsız ve önyargısız bir bakış açısı sağlar. Puanlama sürecinde paydaş, değerlendirme toplantısında baskın olabilecek kişilerden etkilenmez. İkinci avantaj ise katılımcı ne zaman istekli olursa ya da ne zaman uygun olursa o zaman puanlamayı yapabilir. Test yöneticisi, paydaşların bireysel çalışması ile tüm katılımcıların katıldığı bir toplantı ayarlama görevinden kurtulmuş olur.

Puanlama, paydaşın projenin başındaki olasılık ve etki faktörlerini algılamasına bağlıdır. Bunun sonucu olarak belirsizlikler olabilir. Faktörler hakkında daha fazla bilgi edindikten sonra projenin ileri safhalarında bu belirsizlikler azalacaktır. Bu aynı zamanda değişen şartlar ve ek bilgiler sonucu faktör puanlama sonuçlarının düzenli aralıklarla gözden geçirilmesini gerektirir.

Puanlamanın satır satır değil de sütun sütun yapılması çok önemlidir. Faktörleri birbirleriyle olan ilişkilerine göre puanlamak gerekir. Karmaşıklığı puanlarken “En karmaşık risk maddesi nedir?” ve “En az karmaşık risk maddesi nedir?” sorularının sorulması gerekir.

Bu sorular ancak karmaşık risk maddelerini birbirleriyle karşılaştırarak cevaplandırılabilir. Etki faktörleri için de aynı şey geçerlidir, risk maddesinin iş açısından önemi puanırken “İş açısından en önemli risk maddesi nedir?” ve “İş açısından en az önemli risk maddesi nedir?” soruları sorulmalıdır.

6.6.2 Tahminlerin belgelenmesi

Paydaşlar tarafından verilen puanlar paydaşların riskleri algılamasına dayanır. Paydaşlar bir şeylerin yanlış gidebileceğini ya da iş açısından bazı maddelerin önemli olduğunu öngörebilirler. Algılanan riskler bireysel tahminlere ve paydaşlar

tarafından kullanılan gerekçelere bağlıdır. Bu tahminler ve gerekçeler belgelenmelidir. Toplantı sırasında, beklenmeyen sonuçlar ve aykırılıklar olursa bu bilgiler kullanılabilir.

6.6.3 Kişisel gözden geçirme

Başlangıç toplantısı sırasında açıklanan ve tanımlanan kurallar çerçevesinde paydaşlar kendi puanlarını tekrar kontrol edecektir. Paydaşlar faktör ve risk maddeleri üzerinden olası puan değerlerini uygun şekilde paylaşır ve paylaşmadığına ve her ürün risk maddesi için kendisine atanan faktörlere bir puanlama yapıp yapmadığına bakmalıdır.

Gözden geçirme bittikten ve paydaşlar sonuçlardan tatmin olduktan sonra ilerleme sağlanması için bireysel puanlar test yöneticisine sunulur.

6.6.4 Test yönetimi

Bu aşamada, test yöneticisinin daha az deneyimli paydaşları desteklemesi gerekir. Test yöneticisi süreci açıklamalı, kurallar ve faktörleri detaylı şekilde anlatmalı ve puanlama çalışmasında katılımcılara yardımcı olmalıdır.

Sürecin yavaş işlemesi dikkat kaybına neden olur o yüzden bu süreç bir hafta ile sınırlandırılmalıdır. Proje ve test planlama aşamasının kendi ilerleme hızları vardır ve ürün risk değerlendirme süreci bu hıza ulaşmalıdır. Sürecin bitmesine yakın test yöneticisi katılımcılara puanları göndermeleri gerektiğini hatırlatır. PRISMA aracı kayıtların takip edilmesine ve paydaşların hazırlanma sürecinin ilerlemesine olanak sağlar. PRISMA sürecinin bir sonraki adımına geçilmeden önce projedeki rollerden ya da her bir gruptaki paydaştan bir yanıt alınmalıdır. Test yöneticisinin, en azından iki paydaşın her bir faktöre bir puanlama yapıp yapmadığını kontrol etmesi gerekir. Aksi takdirde puanlama paydaşlardan birinin öngörüsüne dayanacaktır.

Bu aşamada sonuç olarak her katılımcıdan üzerlerine atanan faktörlerin puanlanmasını içeren dokümanlar elde edilir. Sonuçlar kurallara uygunluğu kanıtlanmış , tamamlanmış puan tabloları şeklinde olmalıdır.

6.7 Bireysel Puanların İşleme Alınması

Bireysel puanların işleme alınmasının amacı;

- Tüm katılımcıların puanlama yapıp yapmadığını doğrulamak.
- Bireysel puanları taslak halindeki ürün risk matrisiyle birleştirmek.
- Kabul edilen kurallara yapılan uygunsuzlukları tanımlamak ve bunları liste haline getirmek.
- Katılımcılar için görüş toplantısı ayarlamak.

Bu aşamada test yöneticisi, paydaşlardan bireysel sonuçları toplar,onları işleme koyar ve paydaş görüş toplantısına hazırlanır.Bireysel puanların işleme alınması aşaması aşağıdaki görevleri içerir:

- Puanlamanın kontrol edilmesi,
- Bireysel puanların işleme alınması,
- Konu listesi.

6.7.1 Puanlamanın kontrol edilmesi

Paydaşlardan bireysel puanların alınması sırasında test yöneticisi ilk olarak puanlamanın doğru yapılıp yapılmadığını kontrol eder. Eğer tanımlanan ve karşılaştırılan kurallara uyulmadıysa test yöneticisi bunu paydaşlarla çözmek zorundadır.Faktörün veya puanlama değerinin anlamı çok net anlatılmamış ya da bireysel puanlama süreci anlaşılmamış olabilir.

Bu noktada test yöneticisi, paydaşın tahminlerini özellikle yüksek veya alçak puanlama değerlerine göre yapıp yapmadığını kontrol eder. Eğer gerekli ise, paydaşın bireysel hazırlığı tekrar etmesi gerekebilir. Eğer kurallara uyulmadıysa ya da diğer konular paydaş ve test yöneticisi arasında çözülemiyorsa,bu konu görüş toplantısında tartışılmalı ve konu listesi içinde yer almalıdır.

6.7.2 Bireysel puanların işleme alınması

Test yöneticisi bireysel puanları kullanarak ortalama puanları hesaplar. Puanlar her risk maddesi için işlenirken aşağıdaki gibi durumlar oluşabilir.

- Kimse riski, ürüne potansiyel tehlike olarak tanımlamaz.

- Bazı paydaşlar, riski kritik olarak tanımlar ama beklenen olasılık ve etkiler üzerindeki görüşler değişebilir, bu da konu listesine ekleme yapmaya sebep olur.
- Her paydaş bir risk maddesini az çok eşit derecede kritik olarak düşünür.

Her risk maddesinin tüm olasılık ve etki faktörleri için ortalama puanı çıkarılır. Her ürün risk maddesi için olasılık faktörlerinin ortalama değerleri toplanır ve sonuç olarak olasılık faktörleri için toplam puanlama taslağı oluşur. Aynı şekilde her ürün risk maddesi için etki faktörlerinin ortalama değerleri toplanır ve sonuç olarak etki faktörleri için toplam puanlama taslağı oluşur. Hem olasılık, hem de etki faktörleri için toplam puanlar uygun olduğu zaman, her risk maddesi PRISMA ürün risk matrisinde konumlandırılabilir.

Ortaya çıkan sonuç paydaş görüş toplantısında sunulacak ve tartışılacak taslak halindeki ürün risk matrisidir.

6.7.3 Konu listesi

Test yöneticisi tartışılmaya açık konuları görüş toplantısında bir liste haline getirir, Konu listesi tanımlanmış kural kümesine uymayan tüm risk maddelerini içermelidir. Kontrol listesinin boyutu paydaş görüş toplantısını etkisiz hale getirmeyecek şekilde olmalıdır. Eğer test yöneticisinin odaklanması gereken daha önemli konular varsa diğer konuları dışarıdan bir ya da birden çok paydaşla toplantıdan önce çözmeye çalışır.

Kurallara uymama örnekleri aşağıdakilere benzer olabilir.

- Bir risk maddesinin bir faktör için verilen tüm puan değerlerinin dağılımı önceden belirlenmiş eşiği aşarsa yani A katılımcısı en yüksek puanı verir ve B katılımcısı aynı faktör için en düşük puanı verirse bu çözümlenmemiş olarak nitelendirilir. Bu ürün risk seviyesinin sonucu olarak verilen risk seviyelerinde net bir anlaşılma ya da anlaşma olmadığı için güvenilir değildir.
- Risk maddeleri tüm çeyrek alanların bir araya geldiği ürün risk matrisinin merkezine çok yakın konumlanmışlarsa katılımcılar bazı kurallara uymamış olabilir.

Eşik değeri ve diğer kurallara başlangıç aşamasında ki proje seviyesinde karar verilir. Planlama safhası sırasında test yöneticisi tarafından ayarlama yapılabilir. Test yöneticisi ve katılımcılar kural uyumsuzluğu olmadığı halde ürün risk analizinde büyük ölçüde önemi olan konuları(kaybolan ürün riskleri, faktörlerin ağırlıkları, uygulanabilirlikleri) toplantıda konuşabilirler.

Bu adımda ortaya olasılık ve etki puanlarını taslak olarak içeren ve ürün risk seviyelerini geçici olarak gösteren bir ürün risk matrisi ve ürün risk maddeleri listesi çıkar. Aynı zamanda paydaş görüş toplantısında tartışılması gereken maddelerin listesi de bir çıktıdır.

6.8 Paydaş Görüş Toplantısı

Paydaş görüş toplantısının amacı:

- Paydaşları potansiyel risklerin algılanması konusunda yüzleştirmek,
- Tanımlanan risk maddelerinin olasılık ve etkisini tartışmak,
- Bir önceki adımda test yöneticisi tarafından tanımlanan konuları konuşmak,
- Ürün risklerinin ürün risk matrisindeki son yerlerine ve ürün risk seviyelerindeki ortak görüşe mümkün olduğunca ulaşmak.

Paydaş görüş toplantısı, test yöneticisinin toplantının amaçlarını ve takip edilen süreci açıklamasıyla başlar. Test yöneticisi aynı zamanda taslak ürün matrisini sunar. Bu toplantının sonunda ürün riskleri ve risk seviyeleri konusunda ortak bir sonuca varılmalıdır. Sonuç olarak paydaşların anlaşmaya vardığı bir ürün risk matrisi ortaya çıkmalıdır. Tamamlanmış olan tüm puanlar üzerinde ortak görüşe varmak imkansız olabilir. Sonuç olarak her katılımcı kendi rol ve deneyimlerine dayanan belirli risk maddelerinin önemini hesaba katar ve kendi görüşüne sahip olur.

Bazı projelerde görüş toplantısı yapılmadan risk değerlendirmesi uygulanmaya çalışılır. Bu olduğu zaman test yöneticisi bireysel puanları ve ürün risk matrisini işleme koyar. Bunu yaparak test yöneticisi bireysel olarak risk almış olur. Bu yaklaşım tabi ki bu yüzden tavsiye edilmez.

Paydaş görüş toplantısı aşağıdaki görevleri içerir:

- Konu listesini işleme koymak,
- Matris alanını analiz etmek,
- Risk seviyelerini doğrulanması,
- Toplantının özetlenmesi.

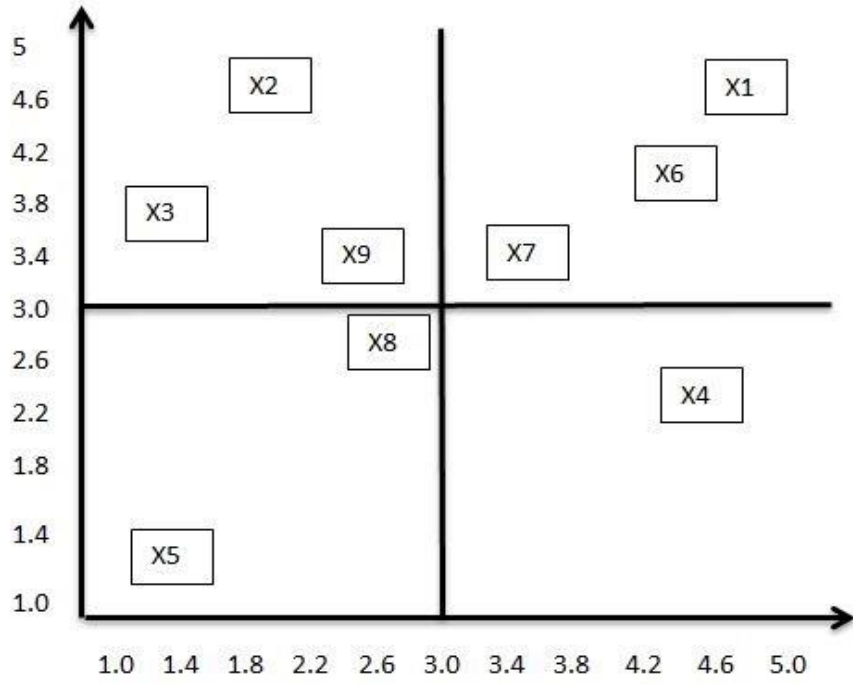
6.8.1 Konu listesini işleme koymak

Paydaşlar, görüş toplantısı sırasında test yöneticisi tarafından hazırlanan konu listesi üzerinden birbirlerinin tartışmalarını anlamak ve ortak görüşe ulaşmak amacıyla ilerlerler. Tartışma, dokümanite edilmiş her faktör için her risk maddesinin tahminlerini içerir. Paydaşların, genellikle en yüksek ve en düşük puanları daha fazla tartışmaya gerek kalmadan başlangıçta açıklamaları istenir. Her iki tarafın açıklamalarından hemen sonra tartışma başlar. Bazen gereksinimlerin farklı anlaşılması puanlamayı etkileyebilir. Bu belirsizliği ortadan kaldırmak için gereksinimler için değişiklik talebi istenebilir.

Toplantıda ortak bir fikre varıldığı zaman puanlar güncellenir ve test yöneticisi acil olarak risk maddesinin pozisyonunu ürün risk matrisinde günceller.

6.8.2 Matris alanını analiz etmek

Aşağıda sunulan Şekil 6.2'de ürün risk matrisiyle dağıtılmış olan ürün riskleri gözlenebilir. Birkaç risk maddesi matrisin uç köşelerindedir. 5 numaralı risk maddesi gibi düşük etkili ve düşük olasılıklı maddeler test sırasında göz ardı edilebilir, 1 numaralı risk maddesi gibi yüksek etkili ve yüksek olasılıklı maddelere test sürecinde çok daha fazla dikkat edilmelidir. Matrisin orta bölgesine yakın olan risk maddeleri ile ilgili karar vermek çok daha zordur. Çünkü hangi bölgeye ait olduklarını saptamak zor olabilir. Burada risk azaltma yaklaşımının uygulanmasından önce bazı analizler gerekli olabilir. Normalde çember içindeki tüm risk maddeleri test yöneticisinin konu listesine dahildir ve tartışılması gerekir. Analizden sonra hangi çeyreğe oturtulmaları gerektiğine karar verilmelidir.



Şekil 6.2 Risk maddelerinin yerleştirilmiş olduğu ürün risk matrisi örneği.

6.8.3 Risk seviyelerinin doğrulanması

Toplantının sonunda final puanlarına karar verilir ve sonuç olarak ürün risk matrisi sunulur. Bu matris paydaşlar tarafından matrisin beklenildiği gibi çalışıp çalışmadığını görmek ve sürpriz sonuçların çıkmasını engellemek için birkez daha doğrulanmalıdır. Eğer sonuçlar paydaşların beklentileriyle eşleşmiyorsa o zaman daha fazla tartışmaya ihtiyaç duyulabilir.

6.8.4 Toplantının özetlenmesi

Her toplantının sonunda, test yöneticisi sonuçları özetler ve amaçlanan hedeflerin karşılanıp karşılanmadığını kontrol eder. Eğer gerekirse sonrasında yapılan işlerin takibi için bir toplantı organize edilebilir. Bazen daha küçük bir grup için belirli bir tartışma toplantısı daha etkili olabilir.

Bu aşamanın sonucunda risk seviyeleri belirlenmiş ürün risk matrisindeki yerleri üzerinde anlaşmaya varılmış ürün risk maddeleri listesi çıkar. Toplantı sırasında yapılan açıklamaları da kapsayan bir karar listesi de ortaya çıkmalıdır.

Ortak görüş tüm paydaşların ekip kararını destekleyebilmesi anlamına gelir. İdeal olarak ürün risk değerlendirme ekibindeki herkes etki,olasılık ve her risk maddesi için risk seviyesinin uygun olduğu konusunda anlaşmaya varmalıdır.Yine de geliştirme sürecinden ve her paydaşın ürüne bakış açısından kaynaklanan bazı anlaşmazlıklar olacaktır. Bu anlaşmazlıklar konu listesinde ayrıntılı bir şekilde belgelenmiştir.Test yöneticisi ortak görüşte birleşebilmek için öneriler sunar ve anlaşmazlıkla başa çıkmak için ayrı bir süreç uygulanır. Ortak görüşe ulaşmak için bazı yaklaşımlar aşağıda tanımlanmıştır.

a) Ekip oylaması: Oylama, olasılık ve etki üzerine ortak görüşe ulaşmaya yardımcı olan bir araçtır. Puanlamada bir anlaşmazlık olduğunda,puanlamaları güçlü olan paydaşların gerekçelerini ve tahminlerini ekibin geri kalanına sunmaları gerekir.

b) Uzman görüşü almak:Eğer uzman takımda değilse,test yöneticisi, PRISMA puanlarını gözden geçirmesi ve risk madde ve faktörlerinin nasıl puanlanacağına dair fikir vermesi için uzmanı toplantıya davet edebilir.Uzman puanlama sürecindeki son sözü söylemek yerine farkında olmadıkları veya bilmedikleri bilgileri ekibe sağlayabilir.

c) Ekip üyelerinden birinin kararına uymak:Ekipte eğer ürün veya süreç üzerine uzmanlığı olan veya risk maddesinin sahibi olan biri varsa son kararı vermesi için ekibin o üyesini görevlendirilebilir.

d) Yüksek derecelendirmeyi kullanmak:Eğer ekip ortak görüşe varamadıysa, en yüksek puanı kullanmayı seçebilir.Bu yaklaşım, madde için riskin azaltılmasından daha çok zaman kaybedilmesine neden olur.

6.9 Farklılaştırılmış Risk Odaklı Test Yaklaşımı Tanımlamak

Farklılaştırılmış risk odaklı test yaklaşımı tanımlamanın amacı;

- Üzerinde anlaşmaya varılan ürün risk matrisine uygun bir test yaklaşımı tanımlamak.
- Hem etkili(en önemli ürün riskleri üzerine odaklanarak) hem etkin(az önemi olanların düşünüldüğü ürün risklerini detaylı olarak test ederek eforu boşa harcamayarak) bir test yaklaşımı kurmak.

- Tüm test yapanların test aktivitelerini ürün risklerine göre uyguladığından emin olmak.

Test yaklaşımı, belirli bir proje için test stratejisinin hayata geçirilmesidir. Projenin amacına ve risk değerlendirmesine, uygulanması gereken test tasarım tekniklerine, yapılması gereken test tiplerine dayalı kararları içerir.

Sonuç olarak test yaklaşımını tanımlamak PRISMA uygulamasını yapmanın ana nedenlerinden biridir. PRISMA yaklaşımı test yöneticisine sınırlı sayıdaki test kaynaklarının olabildiğince etkili kullanımının sağlanması konusunda rehberlik eder. Pratikte birçok test yöneticisi farklılaştırılmış test yaklaşımını tanımlarken risk seviyeleriyle eşit şekilde ilgilenmede zorluk yaşarlar. Prensip olarak dört çeyrek bölgedeki ürün risk maddeleri farklı şekilde test edilmeli ve bu yüzden en azından dört belirgin test yaklaşımı test planı içinde dokümante edilmelidir. Bir maddenin test edilmesinden kimin sorumlu olduğu, nasıl ve ne zaman hangi maddenin test edilmesi gerektiği ve ürün risk değerlendirme sonuçlarıyla örtüşüp örtüşmediği dokümante edilmelidir.

Test yaklaşımı etkili ve etkin test etmeyi sağlamakta önemli bir aşamadır. Test yaklaşımının iki ana özelliği vardır: uygulanacak testin yoğunluğu (test senaryolarının sayısı) ve testin planlama ve yerine getirilmesinde uygulanan önceliklerdir. Farklılaştırılmış risk odaklı test yaklaşımı tanımlama aşaması aşağıdaki görevleri içerir:

- Önceliklendirmelerin yapılması,
- Test tasarım tekniklerinin belirlenmesi,
- Geliştirme yaklaşımının belirlenmesi.

6.9.1 Önceliklendirmenin yapılması

Ürün risk matrisindeki her bir risk maddesinin lokasyonuna bağlı olarak, risk maddeleri önceliklendirilir. Test için sıraya konan tüm risk maddelerinin sonucunda; en önemli olan madde en riskli olan madde demektir ve o maddenin önce test edilmesi gerekir. Bu en ideal durumdur. Geliştirme gerçekleştirilirken, risk odaklı yazılım mimarisi yapmak mümkün olmayabilir. Diğer taraftan test öncelikleri ile ilgili bilgi zamanında verilirse, test yöneticisi ve proje yöneticisi bunu analiz

edebilir ve geliştirme, entegrasyon ve hizmet takviminin önceliklerini belirlemek için bu bilgiyi kullanabilir.

6.9.2 Test tasarım tekniklerinin belirlenmesi

Test etme yoğunluğu farklı test tasarım teknikleri kullanılarak çeşitlendirilebilir. Örneğin yüksek riskli test maddelerinde karar ağacı tekniği kullanılarak ve daha düşük seviyeli risk maddeleri için eşdeğer aralık tekniği kullanılarak çeşitlendirme yapılabilir. Risk, (hem risk seviyesi ve risk tipi) test tasarım tekniğini seçmedeki tek yaklaşım ya da bir test tasarım tekniğinin içinde bir değişken olmalıdır.

Risk en önemlisi olsa da kullanılacak test tasarım tekniklerini seçmede tek faktör değildir. Karar, iç ve dış faktör sayılarına göre verilebilir.

İç faktörler ;

- Kullanılan modeller
- Testçilerin bilgi ve deneyimleri
- Beklenen hata tipleri
- Yapılan dokümantasyon
- Yaşam döngüsü modeli
- Yaşam döngüsü aşaması (yeni geliştirme ya da bakım)

Dış faktörler ;

- Müşteri/sözleşme gereksinimleri
- Sistem Tipi
- Gereksinimlerin düzenlenmesi
- Zaman ve bütçe

Yukarıda test tasarım tekniklerinin sonucu olarak, farklılaştırılmış bir risk odaklı test tanımlamada diğer opsiyonları araştırmak önemlidir. Pratikte ürün risk matrisine dayalı farklılaştırılmış yaklaşımı tanımlamanın birçok başka yolu olduğunu belirttik. Diğer risk azaltma aksiyonları önceliklendirme ve test tasarım tekniklerine ek olarak aşağıda açıklandığı gibi farklılaştırılmış test yaklaşımı tanımlanırken kullanılabilir.

a) Gözden geçirme ve kontroller

Tanımlanan ve analiz edilmiş ürün risklerine dayanarak birisi daha fazla gözden geçirme ya da daha kökten gözden geçirmeler yapmaya karar verebilir. Teknik (olasılık) ürün risklerinde, daha tekniksel gözden geçirmeler yararlı olurken, daha işe yönelik (etki) ürün riskleri için, kullanıcı temsilcileri ile izlenecek yolları belirlemek yararlı olabilir.

b) Test tasarım toplantıları

Yeni bir test tasarımına başlandığı zaman pratik gelişim içinde alınan derslerden kısa bir toplantı organize edilir. Toplantı son kullanıcı, uzman, testçi, yazılım geliştirici ve paydaşları içerir. Toplantı sırasında test maddesi için test düşünceleri ve kritik konular genellikle beyin fırtınası kullanılarak tanımlanır. Bu test yapana, daha iyi bir test tasarımına ulaşmasında yardımcı olur.

c) Test tasarımlarını gözden geçirme

Önemli ürün riskleriyle test tasarımları paydaşlarla ya da diğer test yapanlarla gözden geçirilebilir. Risk seviyesi ile ilişkili olarak yaklaşım şöyle olabilir:

- Kritik risk: Hem iş hem geliştirmeye ilgili paydaşlarla test tasarımının formal olarak gözden geçirmesi.
- Büyük risk: Test ekibiyle test tasarımının formal olarak gözden geçirilmesi.
- Orta risk: Başka bir test yapan tarafından test tasarımının gözden geçirilmesi.
- Düşük risk: Test tasarımının gözden geçirmesi yok.

d) Test senaryolarının detay seviyesi

Ne kadar dokümantasyonun yeterli olduğuna karar vermek zor olabilir. Gereken dokümantasyon seviyesine test tasarımı için gerekli olan dokümantasyon seviyesine bakılarak ya da risk seviyesindeki test senaryoları ilişkilendirilerek karar verilebilir. Kritik risk maddeleri için test senaryolarının gözden geçirmelerini kolaylaştırmak için daha fazla dokümantasyon gerekebilir.

e) Bağımsızlık seviyesi

Test tasarımını yapmak için test senaryoları ve test prosedürlerini tanımlamayı içeren bir testçiye sahip olmak ve daha sonra test prosedürünü uygulayan

başka bir testçiye sahip olmak mümkündür. Testi bağımsız koşturan kişi test senaryolarına karşı daha eleştirci olma eğilimindedir ve sonuç olarak daha çok hata bulmak olasıdır.

f) Deneyimli Kişi

Daha deneyimli testçileri sistemin daha kritik alanlarında görevlendirmek iyi bir önlemdir. Genellikle bu iyi bir fikir olmasına rağmen kişi böcek paradoksundan (*pesticide paradox*) haberdar olmalıdır. Eğer deneyimli testçiler hep aynı yaklaşımı uygularlarsa yeni hataları göremeyeceklerdir.

g) Regresyon testi

Ürün risk değerlendirmesinin sonucu, regresyon test kümesinin içeriğine de yüksek risk alanlarının iyi korunmasını sağlayarak rehberlik etmelidir. Bu yaklaşım otomatikleşen test olaylarını seçerken de kullanılabilir. Sadece düşük riskli test maddeleriyle ilgilenen otomatik test kümesine sahip olmak çok elverişli değildir.

6.9.3 Yenilenen geliştirme yaklaşımı

Ürün risk değerlendirmesinin sonucu geliştirme yaklaşımını ve süreci etkilemek için de kullanılabilir. Örneğin dış kaynaklı olarak geliştirilen maddeler tanımlanan risk seviyesi, yüksek ürün riskleriyle ilişkilendirilen bileşenler sonucunda daha sonra içerden bir yazılımcı tarafından geliştirilebilir. Geliştirme yaklaşımı ve süreçte yapılan değişiklikler oturmuş ürün risklerinin üzerinde, özellikle hataların benzerliğinde sonradan etkili olacaktır.

Bu aşamanın sonucunda ürün riskleri (ürün risk değerlendirmesi sırasında tanımlanmış ve analiz edilmiş) sıraya konarak kabul edilmiş ve tanımlanmış bir test yaklaşımı ortaya çıkar.

Önceden belirtildiği gibi bir çok test uzmanı, farklılaştırılmış bir risk odaklı test yaklaşımını tanımlamada zorluk çekerler. Bu yüzden test stratejisinin bir parçası olabilen bazı ana düşüncelerin, böyle bir test yaklaşımında dokümanite edilerek ana hatlarını oluşturmak büyük oranda tavsiye edilir.

Belirli bir test seviyesindeki test teknikleri tanımlanırken hedeflenen hataları, sınırlamaları ve teknikleri anlamak önemlidir. Test teknikleri tipik olarak güçlere ve

zayıflıklara sahiptir ve genellikle hata çeşitlerinden bir veya bir kaçını tanımlamaya odaklanır.

6.10 PRISMA Sürecinin Avantajları, Dezavantajları ve Dikkat Edilmesi Gereken Unsurlar

Proje içinde PRISMA yaklaşımını tanıtmak zaman alır ve organizasyon içinde iyi hazırlanmış bir giriş gerektirir. Genellikle giriş ürün risk yönetiminin öneminin tüm paydaşlara anlatılmasıyla olur. Organizasyon PRISMA yaklaşımını mevcut risk yönetim yaklaşımıyla ,standartlarıyla, test etme metotlarıyla ve çalışma prosedürleriyle bütünleştirebilirse sürecin kabul edilebilirliği artar.Giriş kısmına pilot bir proje ile başlamak daha yararlı olacaktır.Pilot proje sırasında standart sürece adaptasyonların daha kolay olacaktır.

Başlangıç uygulamasının önemli bir parçası da seçim ve yönlendirme yapan test danışmanının veya test yöneticisinin eğitimi, sorulara cevap vermede bir mekanizma kurulması ve destek sağlanmasıdır.

Tanımlanan önermeler ve PRISMA süreci belirli bir proje çeşidi ya da gelişmiş yaşam döngüsüyle sınırlandırılmamıştır. Projede PRISMA kullanımında gereken eforun test için harcanacak efora uygun oranda olması önemlidir. Organizasyon içinde süreç kullanım sırasında geliştirilebilir ve geliştirilmelidir. Süreç değişen şartlara adapte edilmelidir.Test proje değerlendirmesinin sonuçları ve geriye dönük toplantılar bunun için girdi sağlamalıdır.

PRISMA süreç tanımından, tek bir ürün risk matrisinin tüm proje için yaratıldığı izlenimine kapılınabilir. Geniş projelerde katlanan yani birden fazla ürün risk matrisleri yaratılabilir. Test planı ya da strateji seviyesinde ürün risk değerlendirmesine yön veren ana tek bir ürün risk matrisi ise daha kullanışlı olacaktır

Test yöneticilerinin çoğu PRISMA yaklaşımının kullanışlı olduğunu düşünür. Kullanışlılık hesaplamaları, PRISMA yaklaşımının test sürecine sağladığı yararları açığa çıkarır. Alınan geri bildirimlerden PRISMA sürecinin yararları aşağıdaki gibi sıralanabilir.

- Proje zaman baskısı altında olduğu zaman doğru kararlar almayı destekler,
- Risk iş üzerinden bir konuşma şeklidir ve bu yüzden PRISMA paydaşlarla iletişimi geliştirmenin bir aracıdır,
- Testle ilgili öngöründe bulunmada ve farklılaştırılmış risk odaklı test yaklaşımını tanımlamada iyi bir temel sağlar,
- Proje sırasında ürün riskleri ve test gelişimini görüntülemek için bir sistem sağlar,
- Bileşen ve entegrasyon testi de dahil olmak üzere test aktivitelerine net bir bakış açısı sağlar,
- Ürünün en önemli kısımlarının projenin bir sonraki aşamasına geçilmeden önce test edilmesini garanti eder,
- PRISMA yaklaşımını kullanmanın sonucu olarak test sonucu bulunan hataların öncelik seviyesi artmıştır,
- Ürün risklerini etkileyen faktörler hakkındaki bilinci artırma ve süreç geliştirme aktiviteleri için girdi olarak kullanılabilir.

Alınan geri bildirimlerden PRISMA yaklaşımının kolaylıkları ve zorlukları aşağıdaki gibi sıralanabilir.

- Farklı yaklaşımlarda doğru seviyede risk maddelerini tanımlamak ve yönetilebilir madde sayısında gruplamak genellikle bir zordur. PRISMA yaklaşımını kullanmak bu açıdan kolaydır;
- Bazen paydaşları tanımlamak ve onları özellikle ilk seferde işin içine sokmak zordur;
- Bazı paydaşlar için seçim yapmak zordur; her zaman her şeyin test edildiğini düşündüler;
- Çeşitli faktörlerin anlaşılması her zaman kolay değildir;
- Geliştirme takvimi her zaman test öncelikleriyle aynı çizgide değildir , en önemli risk maddeleri sürecin sonuna doğru geliştirilir;
- Risklere dayalı farklılaştırılmış bir test yaklaşımı tanımlamak zordur; test mühendislerinin yetenek ve bilgi seviyesine de dayanır.

PRISMA sürecinin bir diğer avantajı genel resmin görülmesini sağlamaktır. Risk değerlendirme sonuçlarını ürün risk matrisi şeklinde sunmak çok daha etkili

olacaktır. Liste ve rakamlar içeren bir tablo çok hızlı şekilde okunamaz ve genellikle paydaşlar kendilerini rakamlar üzerinden ilerleyen bir tartışma içinde bulurlar.PRISMA ürün risk matrisinde etken faktörünün x ekseninde , benzerlik faktörünün ise y ekseninde olması ve dört çeyrek dairenin her birinin bir seviye ve risk çeşidini temsil ettiği bir matris risk analizinin sonuçlarını net bir şekilde ortaya koyar.Bu ürün risklerinin tartışılması için çok daha iyi bir temel sağlar.

6.11 Ürün Risk Matrisi İçin Bir Örnek Hesaplama

Beş risk maddesi göz önünde bulundurularak üç paydaş üzerinden örnek bir ürün risk matrisi oluşturulmuştur.Bu paydaşlar;

- Proje lideri; tüm faktörler için değerlendirme yapmıştır.
- Yönetimden bir paydaş; ilgili faktörlerin etkileriyle ilgili değerlendirme yapmıştır.
- Yazılım Mimarı; ilgili faktörlerin olma olasılıklarıyla ilgili değerlendirme yapmıştır.

Risk değerlendirme puan aralıkları 1 ile 5 arasında olacak şekilde belirlenmiştir. Aşağıdaki tablolarda da görüldüğü gibi paydaşlar 5 puan üstünde puanlama yapamamışlardır. Yüksek riskli olan maddelere en fazla 5 puan verilmiştir.

Etki faktörleri için iş açısından önem ve kullanım sıklığı faktörleri hesaba katılırken, olasılık faktörleri için karmaşıklık ve projedeki zaman baskısı hesaba katılmıştır.İş açısından önem ve zaman baskısı faktörleri diğer faktörlere göre daha önemli olduğundan katsayılarını 2 olarak ağırlıklandırılmıştır. Kullanım sıklığı ve karmaşıklık faktörlerinin katsayılarına ise 1 diyelim. Aşağıda proje yöneticisinin, yönetimden seçilen paydaşın ve yazılım mimarının örnek puanlama tabloları vardır.

Tablo 6.1 Proje yöneticisi örnek puanlama tablosu

| Proje Yöneticisi | Etki Faktörleri | | Olasılık Faktörleri | |
|-------------------|-------------------|------------------|---------------------|---------------|
| Faktörler | İş Açısından Önem | Kullanım Sıklığı | Karmaşıklık | Zaman Baskısı |
| Katsayılar | 2.0 | 1.0 | 1.0 | 2.0 |
| 1. Risk Maddesi | 5 | 3 | 5 | 5 |
| 2. Risk Maddesi | 3 | 5 | 5 | 1 |
| 3. Risk Maddesi | 3 | 2 | 1 | 5 |
| 4. Risk Maddesi | 4 | 1 | 2 | 2 |
| 5. Risk Maddesi | 1 | 2 | 3 | 3 |

Tablo 6.2 Yönetimden seçilen paydaşın örnek puanlama tablosu

| Yönetim | Etki Faktörleri | | Olasılık Faktörleri | |
|-------------------|-------------------|------------------|---------------------------------------|---------------|
| Faktörler | İş Açısından Önem | Kullanım Sıklığı | Karmaşıklık | Zaman Baskısı |
| Katsayılar | 2.0 | 1.0 | 1.0 | 2.0 |
| 1. Risk Maddesi | 4 | 2 | Bu faktörler yöneticiye atanmamıştır. | |
| 2. Risk Maddesi | 3 | 5 | | |
| 3. Risk Maddesi | 4 | 3 | | |
| 4. Risk Maddesi | 5 | 1 | | |
| 5. Risk Maddesi | 2 | 3 | | |

Tablo 6.3 Yazılım mimarı örnek puanlama tablosu

| Yazılım Mimarı | Etki Faktörleri | | Olasılık Faktörleri | |
|-----------------|---------------------------------------------|------------------|---------------------|---------------|
| Faktörler | İş Açısından Önem | Kullanım Sıklığı | Karmaşıklık | Zaman Baskısı |
| Katsayılar | 2.0 | 1.0 | 1.0 | 2.0 |
| 1. Risk Maddesi | Bu faktörler yazılım mimarına atanmamıştır. | | 4 | 5 |
| 2. Risk Maddesi | | | 4 | 2 |
| 3. Risk Maddesi | | | 1 | 5 |
| 4. Risk Maddesi | | | 3 | 1 |
| 5. Risk Maddesi | | | 5 | 3 |

Puanlama tabloları alındıktan sonra her risk maddesi için ortalama puanlar hesaplanır. Örneğin 1. Risk maddesi'nin olasılık faktörünü sadece proje yöneticisi ve yazılım mimarı puanlamıştır. Yani bu madde için puanlar toplamı, 4+5=9 ,iki kişi puanlama yaptığı için de ortalama puan değeri $9/2=4,5$ 'dur. Tablo 6.4 'de tüm ortalama puan değerleri hesaplanarak tabloya yerleştirilmiştir.

Tablo 6.4 Örnek ortalama puanlar tablosu

| | Etki Faktörleri | | Olasılık Faktörleri | |
|-----------------|-------------------|------------------|---------------------|---------------|
| Faktörler | İş Açısından Önem | Kullanım Sıklığı | Karmaşıklık | Zaman Baskısı |
| Katsayılar | 2.0 | 1.0 | 1.0 | 2.0 |
| 1. Risk Maddesi | 4.5 | 2.5 | 4.5 | 5.0 |
| 2. Risk Maddesi | 3.0 | 5.0 | 4.5 | 1.5 |
| 3. Risk Maddesi | 3.5 | 2.5 | 1.0 | 5.0 |
| 4. Risk Maddesi | 4.5 | 1.0 | 2.5 | 1.5 |
| 5. Risk Maddesi | 1.5 | 2.5 | 4.0 | 3.0 |

Bu tablolar oluşturulduktan sonra test yöneticisi etki ve olasılık faktörleri için ortalama puanları hesaplar. Aynı zamanda puanların faktörler ve risk maddelerinin varyasyonlarına göre bağımsız puanlar olduklarını kontrol eder. Tablo 6.5'de final puanları görülmektedir.

Tablo 6.5 Final puanlama tablosu

| | Etki Faktörleri | Olasılık Faktörleri |
|------------------------|----------------------------------------------|----------------------------------------------|
| 1. Risk Maddesi | $(4.5 \times 2.0) + (2.5 \times 1.0) = 11.5$ | $(4.5 \times 1.0) + (5.0 \times 2.0) = 14.5$ |
| 2. Risk Maddesi | $(3.0 \times 2.0) + (5.0 \times 1.0) = 11.0$ | $(4.5 \times 1.0) + (1.5 \times 2.0) = 7.5$ |
| 3. Risk Maddesi | $(3.5 \times 2.0) + (2.5 \times 1.0) = 9.5$ | $(1.0 \times 1.0) + (5.0 \times 2.0) = 11.0$ |
| 4. Risk Maddesi | $(4.5 \times 2.0) + (1.0 \times 1.0) = 10.0$ | $(2.5 \times 1.0) + (1.5 \times 2.0) = 5.5$ |
| 5. Risk Maddesi | $(1.5 \times 2.0) + (2.5 \times 1.0) = 5.5$ | $(4.0 \times 1.0) + (3.0 \times 2.0) = 10.0$ |

Ürün risk matrisi eksenlerindeki ölçeklendirmeler minimum ve maksimum puanlara göre değerlendirilir.

Yatay ekseninde etki faktörleri için maksimum puan $10(5 \times 2.0)$ [iş açısından önem maksimum puanı]+ $5(5 \times 1.0)$ [kullanım sıklığı maksimum puanı] 15 e eşittir.

Yatay ekseninde etki faktörleri için minimum puan $2(1 \times 2.0)$ [iş açısından önem minimum puanı]+ $1(1 \times 1.0)$ [kullanım sıklığı minimum puanı] 3 e eşittir.

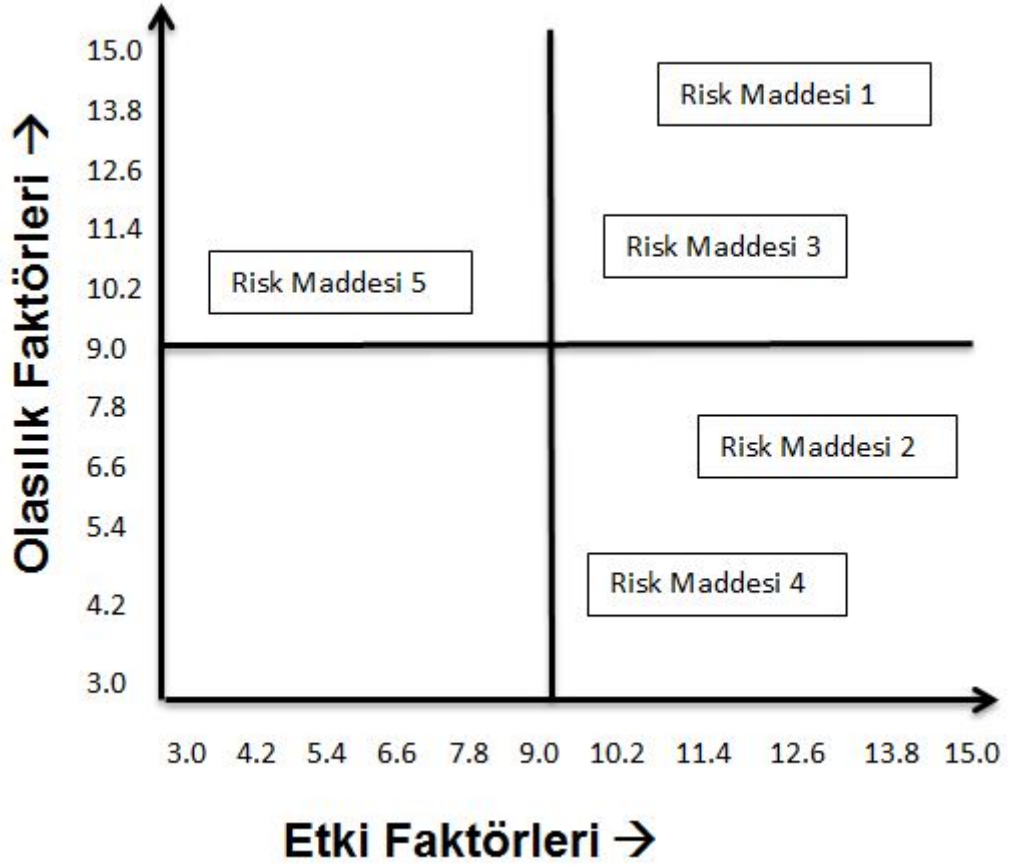
Bu durumda orta hattın $(15+3)/2=9$ da olması gerekir.

Dikey ekseninde olasılık faktörleri için maksimum puan $5(5 \times 1.0)$ [karmaşıklık maksimum puanı]+ $10(5 \times 2.0)$ [zaman baskısı maksimum puanı] 15 e eşittir.

Dikey ekseninde olasılık faktörleri için minimum puan $1(1 \times 1.0)$ [karmaşıklık minimum puanı]+ $2(1 \times 2.0)$ [zaman baskısı minimum puanı] 3 e eşittir.

Bu durumda orta hattın $(15+3)/2=9$ da olması gerekir.

Beş risk maddesi, Şekil 6.3'de görüldüğü gibi ürün risk matrisi üzerine yerleştirilmiştir.



Şekil 6.3 Puanlamalar sonucu oluşturulan ürün risk matrisi örneği

Ürün risk matristen anlaşıldığı üzere risk değerlendirmesinde birinci ve üçüncü risk maddeleri yüksek seviyede risk içermektedir. Hem olma olasılıkları yüksektir hem de oluşmaları durumunda etkileri oldukça fazla olacaktır. Test planlaması aşamasında bu risk maddelerinin önceliklendirilmeleri faydalı olacaktır.

7. TMMi VE PRISMA YAKLAŞIMLARININ TEST SÜRECİ ÜZERİNDE UYGULANMASI İÇİN BİR ÖRNEK

Turizm alanında dünya çapında faaliyet gösteren ve teknolojik gelişmeleri yakından takip eden bir turizm organizasyonunun geliştirmiş olduğu GDS(Global Distribution System) projesinin test süreci üzerinde TMMi modeli içinde PRISMA yaklaşımı uygulanacaktır*.

7.1 GDS Tanımı(Global Distribution System)

GDS Global dağıtım kanalları; insanların ulaşım ve konaklama konusunda her türlü bilet ve hizmet satın alma ya da kiralama ihtiyaçlarını daha hızlı ve ucuz olarak karşılamaları için oluşturulan dinamik veri bankalarıdır.

Bu veri bankaları çok yönlü bir ağ oluşturarak hizmet sağlayıcının tüketiciye fiyat ve kontenjan sunması için özel giriş modülleri oluşturur. Böylece bir otel, havayolu, araba kiralama ya da diğer ulaşım firmaları tüm fiyat ve uygunluklarını GDS kanallarına aktarabilmektedir. Hizmet sağlayıcıdan GDS kanalına yürütülen bilgi akışı aracısız ya da birden fazla aracılı olabilir. Sadece otel sistemleri geliştirmiş olan firmalar sadece otel ile GDS kanalının bağlantısını yapabilirler. Böylelikle otel ya da diğer firmalar her bir kanal ile ayrı ayrı görüşmek ve anlaşmak zorunda kalmaz, tek bir sisteminden birkaç farklı GDS kanalına fiyat ve müsaitlik gönderebilirler.

GDS kanalları yoğun olarak öncelikle dünya genelinde 600.000'in üzerinde faaliyet gösteren ve ofis satışı ile belirli kitlelere hitap eden acentelere sistem alt yapısı sağlayarak, çeşitli ürünleri tek bir bağlantı ile sunar. Acenteler sistemleri üzerinden sorgulama yaparak müşterisi için online olarak rezervasyon gerçekleştirir.

7.2 Problem Tanımı

Projeyi geliştiren organizasyonun halen benimsediği, düzenli ilerleyen ve etkili bir test süreci yoktur. Proje takvimine uyulması için test sürecinin verimli ve etkili ilerlemesi, projenin canlıya hatasız çıkabilmesi ve bütçe aşımı olmaması organizasyon için önemlidir.

*25 sayılı kaynakta söz konusu edilen PRISMA yazılımı uygulamada kullanılmıştır.

Bu proje için ayrı bir test ekibi oluşturulmuştur ancak test sürecinin düzgün ilerlemesinde tek sorumluluk testçilerde değildir. Etkili bir test yaklaşımının benimsenmesi ve bu yaklaşımın geliştirme sürecine paralel uygulanması proje takviminin gecikmesini engelleyecektir. Bu sebeple TMMi modeli bu organizasyonda kullanılmaya üzere seçilmiştir. TMMi modelinin süreç alanları projenin test süreci içinde birebir uygulanmıştır. Ancak GDS sistemi içinde bulunan, birçok modül ve arayüz vardır, bu modüllerin bazıları diğerlerine göre daha karmaşık ve daha çok kullanılan modüllerdir. Örneğin uçuş ve otel arama arayüzleri, araç kiralama ve arama arayüzlerinden çok daha fazla kullanılmaktadır, bununla birlikte araç kiralama arayüzleri çok daha karmaşık bir kod yapısına sahiptir. Bazı modüller ise iş açısından yüksek seviyede risk içermektedir. Ödeme arayüzlerinde herhangi bir hata olması büyük maddi zarara neden olabilir. Bu sebeple projedeki test sürecinde dikkat edilmesi gereken bir çok farklı unsur ve faktör vardır. Testçiler gereksinimler üzerinden test senaryolarını oluştururlar ancak sistemin karmaşıklığı ya da riskli alanlar gereksinimlerde ayrıntılı olarak anlatılmayacağı için testler ile ilgili önceliklendirme yapamazlar. Testlerin önceliklendirilmesi ve yüksek riskli olabilecek alanların projenin erken evrelerinde ve çok daha farklı test varyasyonları kullanılarak test edilmesi, hem test süreci açısından, hem de geliştirme süreci açısından önemlidir. Bu sebeple TMMi modelinin Test Planlama aşamasında test önceliklerinin belirlenmesi gerekmektedir. Test Planlama aşamasında ise test maddelerinin ne şekilde önceliklendirileceği ve hangi faktörlerin göz önünde bulundurulacağı belirsizdir. PRISMA yaklaşımının TMMi modeli içinde Test Planlama aşamasında kullanılması, test sürecinde yüksek riskli alanların ortaya çıkmasını ve önceliklendirilmenin yapılmasını sağlayarak, riskli alanların projenin bitiminden çok önce farkedilerek hataların giderilmesini sağlayacaktır.

7.3 TMMi Yaklaşımı Süreç Alanlarının Uygulanması

GDS sistemini geliştiren organizasyonun uyguladığı bir test süreci olmadığı için bu proje üzerinde TMMi modelinin 2. seviyesi referans alınmıştır. TMMi modelinin yapısı gereği bir organizasyon 2.seviyeyi uygulamadan diğer seviyelere geçememektedir. TMMi 2. seviye içinde bulunan Test Planlama süreç alanında PRISMA yaklaşımı kullanılmıştır. Bu seviyede bulunan diğer süreç alanları özet

olarak sunulacak ve Test Planlama süreç alanı içerisinde PRISMA yaklaşımına ağırlık verilerek bir örnek uygulama yapılacaktır.

TMMi modelinin 2. seviyesinde aşağıdaki süreç alanları dikkate alınmalıdır.

- Test Politikası ve Stratejisi
- Test Planlaması
- Testin İzlenmesi ve Kontrolü
- Test Ortamı
- Test Tasarımı ve Uygulaması

7.4 Test politikası ve test stratejisi süreç alanının uygulanması

TMMi modeli 2. seviyedeki ilk süreç alanı Test Politikası ve Test Stratejisinin belirlenmesi aşamasıdır. Bu süreç alanında sırasıyla aşağıdaki adımlar uygulanmıştır.

- Test politikasının belirlenmesi,
- Test stratejisinin belirlenmesi,
- Test performans göstergelerinin belirlenmesi.

Test politikasının belirlenmesi aşamasında, belirlenen iş gereksinimlerine yönelik test hedefleri konularak bu hedeflerin kalite programı ile örtüşüp örtüşmediğine bakılmıştır. Ürünün kullanıma uygunluğunun doğrulanması, hataların ürün ortaya çıkmadan önce önlenmesi, ürünün standartlara uygunluğunun doğrulanması, ürün kalitesinin görünür şekilde iyi olması ve test sürecinin projenin bitiş tarihinden önce sona ermesi bu proje konulan test hedefleridir. Konulan test hedefleri seçilen paydaşlar ile gözden geçirilmiştir. Tanımlanan test hedeflerine yönelik test politikası yine paydaşlar ile mutabık kalınarak oluşturulmuştur. Test tanımı yapılmış ve ulaşılması gereken kalite seviyesi proje çalışanları ile paylaşılmıştır, proje ekibinden tamamen bağımsız test uzmanları atanarak test organizasyonunun bağımsızlığı kesinleşmiştir. Konulan test hedefleri ve belirlenen test politikası tüm paydaşlara dağıtılmıştır ve iyice anlaşılması için sunumlar yapılmıştır. Son olarak paydaşlar ile birlikte projenin devreye alım planı oluşturulmuştur.

Test stratejisinin belirlenmesi aşamasında genel bir ürün risk değerlendirmesi yapılmıştır ve test için tipik olabilecek kritik alanların üzerinden ayrıntılı olmadan geçilmiştir. Ürün risk değerlendirmesine katkıda bulunabilecek olan paydaşlar için ön hazırlık yapılmıştır. Test stratejisi, test tipleri(bileşen testi, entegrasyon testi, sistem testi, kullanıcı kabul testi) belirlenip tanımlanarak ortaya çıkmıştır. Her tip test için amaçlar sorumluluklar, ana işler, başlangıç ve çıkış kriterleri belirlenmiştir. Bileşen testlerinde, bileşenlerin tasarım belgesinde anlatıldığı şekilde çalıştığı doğrulanmalıdır. Entegrasyon testlerinde ise tüm bileşenlerin ortak halde de doğru çalıştığı doğrulanmalı ve arayüzlerin arayüz dokümanında tanımlandığı şekilde görüldüğü doğrulanmalıdır. Sistem testlerinde, sistemin gereksinim dokümanına uygun olduğu doğrulanmalı ve sistem gereksinimlerinin tümünün kapsamı kapsamadığına bakılmalıdır. Kullanıcı kabul testlerinde ise sistemin kabul kriterlerine uygunluğuna bakılmalı, sistemin kullanıma uygunluğu doğrulanmalı ve tüm iş gereksinimlerinin kapsamı kapsamadığına bakılmalıdır. Test stratejisinin belirlenmesi aşamasında aynı zamanda test senaryolarının yazım teknikleri, otomasyona uygun olabilecek olan alanlar, kullanılacak standartlar ve başlangıç/bitiş kriterleri de belirlenmiştir.

Kalitenin ölçülebilir olması için test performans göstergeleri belirlenmeli ve verilerin toplanması, saklanması, analizi ve prosedürlerin raporlanması yapılmalıdır. Seçilen projede test performans göstergeleri olarak test için harcanan çaba ve maliyet, test teslimatı için gereken zaman, bulunan hata sayısı, hataların bulunma yüzdesi, testin kodu kapsama yüzdesi ve test olgunluk seviyesi, test performans göstergeleri olarak belirlenmiştir. Belirlenen test performans göstergelerinin verileri düzenli olarak güncellenmiş, paydaşlarla gözden geçirilmiş ve raporlanmıştır.

7.5 Test planlaması süreç alanının uygulanması

TMMi modeli 2. seviyedeki ikinci süreç alanı Test Planlamasının yapılması aşamasıdır. Bu süreç alanında PRISMA yaklaşımı ele alınarak ayrıntılı bir biçimde süreç incelenmiştir.

Test planlama aşamasında sırasıyla aşağıdaki adımlar uygulanır.

- Ürün risk değerlendirmesinin yapılması,
- Bir test yaklaşımının belirlenmesi,
- Test ölçüm ve tahminlerinin yapılması,
- Test planı geliştirilmesi,
- Geliştirilen test planına bağlı kalınması.

İlk maddeyi göz önünde bulundurursak risk maddeleri tanımlanmalı ve değerlendirilmelidir. Ürün risk değerlendirmesi test için kritik alanların belirlenmesini sağlar. Ürün risk değerlendirmesinin yapılması aşamasında ürün riskleri belirlenmiş ve analiz edilmiştir. Bu aşamada riskler kategori kategori ayrılabilir(fonksiyonel riskler, mimari riskler, fonksiyonel olmayan riskler vb.) ancak seçilen projede test edilecek olan arayüzlerin her biri bir risk maddesi olarak ele alınmıştır. Test maddelerinin her biri için risk değerlendirmesi yapılmıştır. Ürün riskleri tanımlandıktan ve analiz edildikten sonra ikinci adımda bir test yaklaşımı belirlenmelidir. Belirlenen test yaklaşımı PRISMA yaklaşımıdır.Üçüncü adımda zaman ve bütçe aşımı olmaması için test maddeleri üzerinde tahmin yürütülerek önceliklendirmeler yapılmalıdır. Bu aşama için PRISMA yaklaşımı sonucunda ortaya çıkacak olan ürün risk matrisi bize yardımcı olacaktır.Dördüncü adımda test planı ortaya çıkarılmalı ve beşinci adımda test planı gözden geçirilerek belirlenen plana uyulmalıdır. Bir önceki adımda ürün risk matrisi sonucunda yapılan test maddesi önceliklendirmelerine göre bir test planı oluşturulmuştur ve proje boyunca plan haftalık olarak gözden geçirilmiştir.

PRISMA yaklaşımı referans alınarak bu aşamalar uygulanacaktır.

7.6 PRISMA Yaklaşımının TMMi Süreci İçinde Uygulanması

Altıncı bölümde anlatıldığı gibi PRISMA yaklaşımında sırasıyla başlangıç aşaması, planlama aşaması , başlangıç toplantısı aşaması, ayrıntılı risk tanımlaması aşaması, bireysel hazırlıkların yapılması aşaması, bireysel puanların işleme alınması aşaması, ortak görüş toplantısı ve farklılaştırılmış risk odaklı test yaklaşımı aşamaları uygulanmıştır.

7.6.1 Başlangıç aşamasının uygulanması

Altıncı bölümde anlatıldığı gibi PRISMA yaklaşımının ilk aşaması Başlangıç aşamasıdır. Bu aşamada standart süreç tanımlanmış ve kurallar belirlenmiştir.

a) Standart sürecin tanımlanması

Seçilmiş olan organizasyon içinde sürecin uygulanabilir olması için kullanılacak olan standart süreç bu aşamada tanımlanmıştır.PRISMA süreç adımları dokümanite edilmiş ve süreç içinde rol alan her çalışana iletilmiştir. Süreç iyileştirilmesi ve denetimi bu doküman üzerinden yapılmıştır.Ürün risk matrisi formatı test yöneticisi tarafından belirlenmiştir.Matriste dört çeyrek alan kullanılmasının projenin kapsamı açısından daha sağlıklı olacağına karar verilmiştir.

b) Etki ve olasılık faktörlerinin belirlenmesi

Bazı risk analiz tekniklerinde risk seviyesi etki ve olasılık faktörlerinin çarpılmasıyla belirlenir. Ancak etkisi yüksek olacak ancak olma olasılığı düşük bir risk için yüksek seviyede puanlama yapmak anlamsızdır. Etki faktörleri genellikle iş riskleri ile ilgiliyken, olasılık faktörleri daha çok teknik riskleri içerir. Seçilen proje üzerindeki etki ve olasılık faktörleri aşağıdaki şekilde belirlenmiştir.

Kullanılan etki faktörleri;

- i. Kritik alanlar
- ii. Görünür alanlar
- iii. En çok kullanılan alanlar
- iv. İş açısından önem
- v. Tekrar çalışma maliyeti

i. Kritik alanlar

Kritik alanlara karar vermek için sorumlu kişinin, sistem kullanımını tüm çevresi içinde analiz edip,sistemin başarısız olabilecek durumlarını bulması gereklidir. Bu işlem tamamlandığında başarısızlık hallerinin olası sonuçları yada en kötü durum senaryoları analiz edilmelidir. Acil durumda başvurulabilecek yedek olanakların, kullanıcılar tarafından olası elle kontrollerin ve analistlerin hesaba katılması önemlidir. Zarar,ürüne bağlı bir çok farklı şey anlamına gelebilir,bazı

ürünler için güvenlikle ilgiliyken , bazısı için sadece maddi zararlar ilişkilidir. Seçilen projede ise birçok alan maddi açıdan kritik iken kredi kartı ile ödeme yapılan alanlarda güvenlik çok önemlidir. Bu maddenin ağırlık değeri 5.0 olarak belirlenmiştir.

ii. Görünür alanlar

Görünür alanlar, eğer yanlış giden bir şey olursa birçok kullanıcının tecrübe edeceği alanlardır. Hesaba katılan bu faktör de bu tür problemlere kullanıcıların toleransındır. Eğitimli olmayan kullanıcılara yönelik yazılımlarda ve halk tarafından kullanılmak için tasarlanan yazılımlarda , kullanıcı ara yüzüne özen göstermek gerekir. Sağlık ana fikir olabilir. Dış bakış (proje sınırların dışında) ve iç bakış (sadece içerden kullanıcıların problemi tecrübe etmesi) arasında ayırım yapılmalıdır. Uçuşların seçildiği ve listelendiği, hotel rezervasyonlarının yapıldığı alanlar bu proje için en görünür alanlar arasındadır. Seçilen projede test maddeleri bu faktör için görünür olma durumlarına göre ağırlıklandırılmıştır. Bu maddenin ağırlık değeri 5.5 olarak belirlenmiştir.

iii. En çok kullanılan alanlar

Bazı fonksiyonlar her gün kullanılırken diğer fonksiyonlar arada bir kullanılır. Bazı fonksiyonlar birçok kullanıcı tarafından kullanılırken, bazıları sadece birkaç kullanıcı tarafından kullanılır. Birçok kişi tarafından her zaman kullanılan fonksiyonların başarısız olmasının etkisi çok büyük olur. Test önceliği, sıklıkla ve ağırlıkla kullanılan fonksiyonlara verilmelidir. Gün başına yapılan iş sayısı önceliklere karar vermede yardımcı olabilir.

En çok kullanım alanları aşağıdaki şekilde tanımlanabilir.

- Kaçınılmaz alan: Ortalama kullanım oturumu sırasında birçok kullanıcının kullandığı ürün alanı,
- Sıklık: Her oturum sırasında olmasa da çoğu kullanıcının sonunda kullanmak zorunda kalacağı ürün alanı,
- Ara sıra: Ortalama bir kullanıcının hiç ziyaret etmeyeceği fakat ara sıra daha profesyonel ve deneyimli bir kullanıcının fonksiyonlarla ilgilendiği bir ürün alanı,

- Ender:Çoğu kullanıcının hiç ziyaret etmediği ve sadece çok çalışılmamış basamaklar yaparken kullanıcıların ziyaret ettiği ürün alanı.

Seçilen projede sisteme giriş yapılan, uçuş ve otellerin aratılıp ve listelendiği, kredi kartıyla ödemenin yapıldığı alanlar en çok kullanılan alanlardır. Acente kullanıcılarının komisyon hesaplarının yapıldığı alanlar ise en az kullanılan alanlar arasındadır. Bu maddenin ağırlık değeri 6.5 olarak belirlenmiştir.

iv. İş açısından önemi

Etkiye bakmanın bir diğer yolu da maddenin iş seviyesindeki önemidir. İş konusunda tanımlandığı gibi gelişmekte olan sistemin temel amacıyla direk olarak ilişkilendirilip ilişkilendirilmeyeceğine ve sistemin iş değerinin bu özelliği taşımayarak azalıp azalmayacağına bakılması gerekir. Seçilen projede ise kullanıcının ödeme yaptığı ve kart bilgilerini girdiği alanlar, komisyon hesaplamasının olduğu alanlar iş açısından oldukça önemlidir. Bu maddenin ağırlık değeri 7.5 olarak belirlenmiştir.

v. Tekrar çalışma maliyeti

Ürün riskinin etkisini belirtirken seçilen başka bir faktör iş tekrarının maliyetidir. İş tekrarının maliyeti görünür bir şekilde ürün riskine bağlı olarak sapabilir. Bazı alanlarda güncelleme, maliyetli olabilecek iken ,bazı alanlarda müşteri tarafından kurulan basit bir yazılım güncellemesi yeterli olabilir. Bazı alanlarda ise sistemi güncellemek için ve hatayı düzeltmek için sistem mühendisi gerekli olabilir. En kötü durum senaryosunda sistemin(lerin)geri yüklenmesi gerekli olabilir. Seçilen projede ise organizasyon tarafından hazırlanmış olan diğer uygulamalar ile olan entegrasyon alanlarında sorun çıkması en çok maliyeti arttıracak alanlardır. Ağırlıklandırma bunlar göz önüne alınarak yapılmıştır. Bu maddenin ağırlık değeri 5.7 olarak belirlenmiştir.

Kullanılan olasılık faktörleri aşağıdaki gibidir;

- vi. Karmaşıklık
- vii. Değişiklik yapma sıklığı
- viii. Yeni teknolojiler ve metotlar
- ix. Zaman baskısı
- x. Arayüz
- xi. Boyut

- xii. Hata gemiři
- xiii. Gereksinimlerin kalitesi

vi. Karmařıklık

Hata oluřumunda karmařıklık nemli bir faktrdr. Birok deęiřik karmařıklık ltleri mevcuttur ve yksek karmařıklık ltleri genellikle problemliler alanlar belirtebilir. Karmařıklık, ok sayıda deęiřken kullanılmasıyla, karmařık mantık ve karmařık kontrol yapısından kaynaklanabilir. nk karmařıklık deęerlendirmesi farklı karmařıklık analizleri, problemliler alanlar bulmak gibi birok karmařıklık grřne dayanır. Karmařıklığı faktr olarak kullanırken hangi ltn kullanılacağını tanımlamak ve karmařıklığın risk deęerlendirmesine katılan paydařlar tarafından nasıl tanımlanacağına bakmak gerekir.

Seilen projede listeleme arayzleri en karmařık arayzlerdir. Bunlarla birlikte benzer fonksiyonlar gerekleřtiren arayzlerin karmařıklık deęerleri de aynıdır. rneęin uuř arama sayfası ile oteller arama sayfası aynı mantıkta alıřır. Kod yapıları aynıdır. Bu yzden de karmařıklık aısından yakın deęerler alacaklardır. Bu maddenin aęırlık deęeri 5.0 olarak belirlenmiřtir.

vii. Deęiřiklik yapma sıklığı

Deęiřiklik nemli bir hata reticisidir. Bunun bir sebebi deęiřikliklerin bireysel olarak kolay olarak anlařılmaması ve bu yzden etkilerinin tamamıyla analiz edilmemesidir. Bařka bir neden de genellikle baskı altında yapılması ve analizin tamamen yapılmamasıdır. Bu genellikle beklenmeyen yan etkilerle sonulanır. Yapılan deęiřiklikler gnlk olarak tutulmayabilir. Bu konfigrasyon ynetim sisteminin bir parasıdır. Sık deęiřiklikler bařlangıtan beri kt tasarlanan kodun belirli bir alanıyla alakalı olabilir ya da birok deęiřiklik tarafından zarar gren orijinal tasarımın kt tasarlanmasına neden olabilir. Birok deęiřiklik yetersiz gereksinim analizinin de genellikle bir belirtisidir. Byk miktarda deęiřtirilmiř alanlar, kullanıcı beklentileriyle uyulařmayabilir.

Projede ise geliřtirme ařaması bařladıktan sonra mřteriden birok deęiřiklik isteęi gelmiřtir. zellikle arayz tasarımları ok fazla deęiřmiřtir. Puanlamalar da bu ynde yapılmıřtır. Bu maddenin aęırlık deęeri 5.8 olarak belirlenmiřtir.

viii. Yeni teknoloji ve metotlar

Yeni aygıtlar, metotlar ve yeni teknolojiler kullanılarak geliştirilen ürünler yüksek sayıda hata ile sonuçlanan bir deneyim kazandıracaktır. Yeni programlama dilleri mevcut programlama alıştırmaların uyarlamasına sıkça gereksinim duyar. Herhangi bir yeni araç ya da teknik yeni problemleri getirebilir. Araç değişikliklerine ek olarak, metot ve modellerde değişiklikler yüksek sayıda hatayla sonuçlanabilir. Özellikle metot olgunlaşmamışsa ya da takımın metotla ilgili deneyimi düşükse bu önerme doğrudur. Eğer geliştirmede sabit metotlar kullanılırsa sistemin oldukça güvenli olması beklenir. Diğer bir taraftan eğer yeni ya da kanıtlanmayan metotlar ve modeller kullanılırsa, sistem daha az güvenilir olabilir.

Projede görev alan yazılımcılar kullandıkları “framework” hakkında bilgi sahibi olmadıkları için iki aylık bir öğrenme sürecine ihtiyaç duymuşlardır. Bunun dışında kullanılmış olan çok fazla yeni teknoloji olarak adlandırılabilir bir faktör yoktur. Bu maddenin ağırlık değeri 3.5 olarak belirlenmiştir.

ix. Zaman baskısı

Zaman baskısı insanların kısa kesintiler almasına sebep olur. İnsanlar genellikle kalite kontrol aktivitelerini atlayarak her şeyin yolunda gideceğine dair olumlu düşünerek problemi çözmeye konsantre olurlar. Zaman baskısı mesai dışı çalışmaya da neden olabilir. Uzun süren iş saatleri sonunda daha fazla hata yapılması ve insanların konsantrasyon kaybı olma eğilimi içinde olmaları bilinen bir gerçektir. Yersiz iyimserlik ve mesai dışında da çalışma, yüksek seviyelerde hata yoğunluğuna sebep olabilir. Geliştirme sürecinde zaman baskısı hakkında veri proje planları yapılarak programcı veya yöneticilerle konuşularak zaman hesaplaması yapılmalıdır.

Projenin ilerleyen safhalarında analiz dokümanında müşterinin belirtmediği ancak sonrasında talepte bulunduğu bir çok yeni istek olmuştur. Buna rağmen projenin bitiş tarihi güncellenmemiştir. Haliyle çalışanlar üzerinde ciddi bir zaman baskısı oluşmuştur. Bu maddenin ağırlık değeri 6.8 olarak belirlenmiştir.

x. Hata gemiři

Hataların giderilmesi genellikle yeni hatalara sebep olur ve bu yzden hata eęilimli alanlar genellikle deęiřmez. Gemiřte bileřenlerin sahip oldukları hatalara gelecekte de sahip olmaları muhtemeldir. Eęer tasarım ve kodlama analizleri ve bileřen ve entegrasyon testi hata istatistikleri mevcutsa o zaman öncelikler bunu izleyen test safhalarından görüntülenebilir. Bu maddenin aęırlık deęeri 7.8 olarak belirlenmiřtir.

xi. Gereksinimlerin kalitesi

Belirsizlikle tanımlanan gereksinimler projenin problemlere sahip olmasına neden olur. Gereksinimler metin olarak, kullanıcı senaryosu olarak ya da faktör olarak tanımlansa da tanımlanmasa da bu faktör daha sonra ki proje ařamasında hataların bir kaynaęıdır. İyi tanımlanan ve sabit olan gereksinimler genellikle kullanıcıların beklentileriyle tanımlanan bir çözümlerle sonuçlanır. Kötü tanımlanan ve istikrarsız gereksinimler genellikle birçok sorun için kaynak ve bu alanlarda ihtiyaç duyulabilecek test eforunun bir iřareti olarak düşünülebilir. Bu maddenin aęırlık deęeri 5.9 olarak belirlenmiřtir.

xii. Arayüz

Birçok hata, bileřenlerin, altyapının, donanımın veya yazılımın arayüzlerle olan baęlantı noktalarında ortaya çıkmaktadır. Bu genellikle birimler arası iletişim ve zamanlama problemlerinden kaynaklanır. Ara yüzleri fazla olan alanlar genellikle hataya daha eęilimlidir. Bu açıdan genellikle iç ve dış ara yüzler arasında ayırım yapılır. Kullanıcı ara yüzleri genellikle dış arayüz olarak adlandırılır.

xiii. Boyut

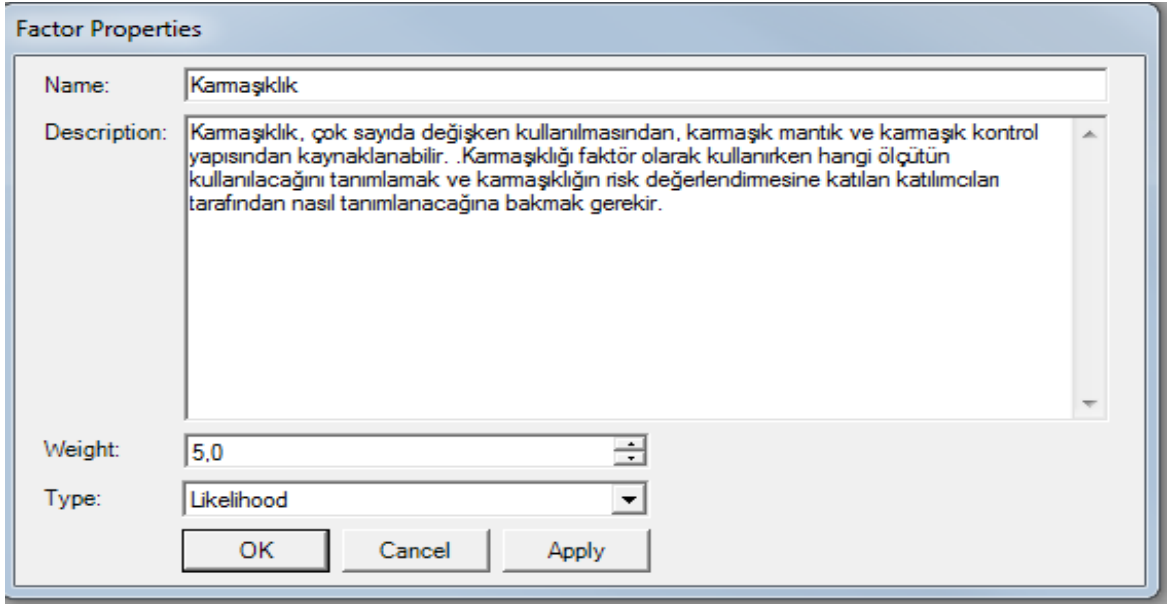
Bazı zamanlar, insanlar çok büyük olan bileřen ya da alt yapıları anlamakta güçlük çekerler. Bu yzden büyük bileřenler ortalama büyüklükteki bileřenlerden daha fazla hata eęiliminde olabilirler. Bu yzden bu faktörle ne anlatılmak istendięi ve karşılařtırmalı puanları elde etmenin nasıl ölçüleceęini paydařlara açıklamak önemlidir.

c) Her faktör için aęırlık deęeri atanması

Aęırlık deęeri, ürün risk matrisindeki ürün risklerinin pozisyonlarına büyük etkisi olacak bir deęer olarak kullanılır. Aęırlıklandırma güncel veriler yoluyla

yapılacağı gibi proje değerlendirme sonuçlarına ve geçmiş projelere dayanarak da yapılabilir. Standart ağırlık değerlerinden çok sapılmaması önerilir. Tutarlılık için organizasyon seviyesinde aynı ağırlık değerlerinin ortaklaşa belirlenmesi ve kullanımı önerilir.

Aşağıda seçilmiş olan projedeki faktörlerin PRISMA aracı üzerindeki tanımlama ekranı gösterilmiştir. Belirlenen faktörün adı yazıldıktan sonra açıklaması yapılmıştır. Bu faktörden etkilenen risk maddesinin ürün risk matrisindeki pozisyonunun belirlenmesi için her faktöre “Weight” alanından bir ağırlık değeri verilmiştir. “Type” alanından faktörün olasılık faktörü mü yoksa etki faktörü mü olduğu seçilmelidir. Projeyi yürüten organizasyon tarafından ağırlık değerleri 1 ile 10 arasında olacak şekilde belirlenmiştir.



Şekil 7.1 Faktör tanımlama ve ağırlıklandırma ekranı

d) Puanlama aralıkları ve kuralların belirlenmesi

Bu aşamada kurallar test yöneticisi tarafından belirlenerek paydaşlara duyurulmuştur. Belirlenen ilk kural puanlama aralığı kuralıdır. Puanlama aralığı olarak 1-5 aralığı seçilmiştir. Paydaşlar risk maddelerine 1-5 arasında puan verebileceklerdir. İkinci kural ise dağılım kuralıdır. Paydaşlar her test maddesine aynı puanı veremezler hatta verdikleri puanlarda eşit dağılım yapmalıdırlar yoksa test yöneticisi tarafından uyarılırlar. Son kural ise her

madde ile ilgili puanlama yapma zorunluluğudur. Paydaşlar üzerlerine atanan her faktörle ilgili sorumluluk sahibidirler ve puanlama yapmak zorundadırlar.

7.6.2 Planlama aşamasının uygulanması

Altıncı bölümde anlatıldığı gibi PRISMA yaklaşımının ikinci aşaması Planlama aşamasıdır. Bu aşama test yöneticisi tarafından yürütülmüştür. Öncelikle kapsam tanımlanmıştır ve tüm proje ekibiyle paylaşılmıştır. Sonrasında etki ve olasılık faktörleri belirlenmiştir. Proje içerisinde ürün risk değerlendirmesine yardımcı olabilecek dokümanlar toplanmıştır. Katılımda bulunacak ve puanlama yapacak paydaşlar belirlenmiştir. Etki ve olasılık faktörleri de seçilen paydaşların organizasyon içindeki rollerine göre dağıtılmıştır.

a) Kapsamın tanımlanması

Seçilmiş olan GDS projesi bir web projesidir ve çevik yöntemle geliştirilecek olan bir projedir. Projenin içeriği test yöneticisi tarafından tüm paydaşlara ve proje çalışanlarına ayrıntılı bir şekilde anlatılmıştır. Gereksinimler kullanıcı senaryosu şeklinde yazılmıştır ve her bir kullanıcı arayüzü bir risk maddesi olarak atanmıştır. Yazılımcıların sorumluluk alanları da kullanıcı arayüzleri üzerinden verilmiştir haliyle test süreci de bu şekilde işleyecektir. Testçiler ve yazılımcıların tüm mimari dokümanları görme yetkisi vardır böylece oluşabilecek teknik riskler ile ilgili ön bilgi sahibi olmuşlardır.

b) Uyum faktörleri ve kuralların belirlenmesi

Başlangıç adımında , organizasyonla genel anlamda ilişkili olan olasılık ve etki faktörleri tanımlanmıştır. Bu aşamada faktörler sistemle ilişkili olup olmadığına karar verilmesi için test yöneticisi, yazılım mimarları ve proje yöneticisi ile tekrar gözden geçirilmiştir. Olasılık faktörleri için yazılım mimarları Arayüz ve Boyut faktörlerinin kullanılmasına gerek olmadığı yönünde görüş bildirmişlerdir. Bu faktörler olasılık faktörleri listesinden çıkarılmıştır. Faktörlere ek olarak, tüm kuralların değerleri ve değişken faktörlerin ağırlıkları gözden geçirilmiştir.

c) Belgelerin bir araya getirilmesi

Planlama aşamasında kullanılacak belgelerin sırası paydaşlar tarafından kararlaştırılmıştır. Girdi dokümanlarının çeşidi büyük ölçüde ürün risk değerlendirmesinde uygulanan test tiplerine(bileşen testi, entegrasyon testi, kullanıcı kabul testi vb.) göre seçilmiştir. Takım liderleri ve kalite ekibi seçilen belgeleri kontrol etmiştir. Gerekli kalite seviyesinde olmayan belgeler rapor edilerek doküman sahiplerine iletilmiştir. Seçilen dokümanlar; gereksinim dokümanı, yazılım mimarisi dokümanı, yazılım test dokümanı, analiz dokümanı ve kullanıcı arayüzü dokümanıdır.

d) Ürün risk maddelerinin tanımlanması

Kullanıcı arayüzleri, gereksinimler ve olası girdilere dayalı ürün riskleri test yöneticisi tarafından tanımlanmıştır. Bu liste beyin fırtınası yapılarak belirlenmiştir. PRISMA çok büyük ölçüde gereksinimler üzerindeki ürün risk değerlendirmelerinin tanımlanmasına dayansa da seçilen projede gereksinimler de kullanıcı arayüzleri gruplanarak oluşturulduğu için her kullanıcı arayüzü bir risk maddesi olarak belirlenmiştir. İletişim, başarılı bir proje için gereklidir. Ürün riskleri, paydaşlar tarafından anlaşılacak şekilde tanımlanmalıdır. Bir risk gerçekleştiği zaman bunun ne anlama geldiği açık olmalıdır. Risk odaklı test yaklaşımında test süreci test kaynaklarının etkili bölüşümüne izin vermek için net bir şekilde tanımlanmalıdır. Eğer böyle bir yaklaşım ayrıntılı şekilde tanımlanmazsa ve bir ürün riski için ne kadar testin yeterli olacağına karar vermek testçinin inisiyatifine kalırsa, risk odaklı test yaklaşımının tüm yapısı çöker. Aşağıda PRISMA aracı kullanılarak tanımlanmış bir örnek risk maddesi tanımlama ekranı vardır.

Name: Flight Searching

Description: End user will use this screen to search the flights. Return or One Way options going to be options users has to choose one of the option. Then user will enter Departure date, departure airport and arrival airport. If its return flight return date will be enable on the screen and user will enter the return date. There will be optional fields for users choiche Direct and search range for the flight if the users flight dates are flexible then he can choose a range for search. Default flight class option will be Economy. For the airport selection there is an option for autocomplete. If the user enters moscow the search will be include all airports in moscow similar as Istanbul. If there is an one airport in that city it will be searched for that specific airport.

Quality: Functionality

Label: 2

OK Cancel Apply

Şekil 7.2 Risk maddesi tanımlama ekranı

Toplamda 17 adet risk maddesi tanımlanmıştır. Test maddeleri proje içindeki sıralarına göre numaralandırılmıştır. Her biri için açıklama alanına ilgili test maddesinin neden riskli olabileceği tam olarak hangi fonksiyonu gerçekleştirdiği yazılmış ve kalite açısından nasıl etkili olacağı “Quality” alanından etkinlik,işlevselliği,bakım kolaylığı, taşınabilirlik, güvenilirlik,kullanılabilirlik seçeneklerinden biri seçilerek belirlenmiştir.

Tanımlanan risk(test) maddeleri numaraları ile Tablo 7.1’de gösterilmiştir. Test maddeleri ürün risk matrisi üzerinde verilen numaraları ile görüntülenecektir.

Tablo 7.1 Puanlama yapılacak olan test maddeleri

| | |
|-------------------------------------------|--------------------------------------------|
| 1. Login/New User | 3. Hotel Searching |
| 1.1 I forgot my password | 3.1 Hotel Listing |
| 1.2 Order History | 3.2 Hotel Purchasing/Information |
| 1.3 Messages Page | 3.3 Hotel Payment/Confirmation |
| 2. Flight Searching | 4. Transfer Search |
| 2.1 Flight Listing/Booking | 4.1 Transfer Listing/Add To Basket |
| 2.2 Flight Confirmation/View Order | 4.2 Transfer Booking/Paym/Ticketing |
| 2.3 Flight Payment/Ticketing | 4.3 Transfer Ticketing/View Order |
| 5. Backoffice | |

e) Paydaşların tanımlanması,seçilmesi ve faktörlerin paydaşlara atanması

Bu adımda ürün risk değerlendirmesinde bulunan paydaşlar belirlenmiştir. Paydaşlar risk analizleri ve tanımlama için gerekli bilgiyi sağladığından,doğru paydaşları seçmek önemlidir. Paydaşlar, sistemden sorumlu olanlar(ürün sahibi,proje yöneticisi, programcı),ürün doğru çalışmadığında etkilenenler(bakım,yardım masası) ve sistemi kullananlar(son kullanıcılar) arasından seçilebilir. Paydaş tanımı, risk değerlendirme sürecinin parçası olarak yapılmıştır.Sürece katılmayan her paydaşın ürün risk değerlendirmesinde eksiklik yaratacağı, paydaşlara anlatılmıştır. Şekil 7.3'de PRISMA aracı kullanılarak tanımlanmış bir paydaş tanımlama ekranı vardır.

Stakeholder Properties

Name: Ece Aşçı

Participating

Telephone:

E-Mail: ece.asci@otiholding.com

Role: Programmer

Status: Form Received

Factors

Available Factors:

- Hata Geçmişi
- Gereksinimlerin Kalitesi
- En Çok Kullanılan Alanlar
- İş Açısından Önemi
- Tekrar Çalışma Maliyeti

Assigned Factors:

- Karmaşıklık
- Değişiklik Yapma Sıklığı
- Yeni Teknolojiler ve Metotlar
- Zaman Baskısı
- Görünür Alanlar
- Kritik Alanlar

OK Cancel Apply

Şekil 7.3 Paydaş tanımlama ekranı

Bir kişi yönetim seviyesinden olmak üzere toplamda sekiz tane paydaş belirlenmiştir. Bir proje yöneticisi, bir test yöneticisi, iki yazılım mimarı, iki

programcı ve bir de test uzmanı seçilmiştir. Proje için belirlenen tüm faktörler Şekil 7.3'de görüldüğü gibi "Available Factors" alanında görülmektedir. Her paydaş her faktörü puanlamadan sorumlu değildir. Her bir paydaşın sorumlu olduğu faktörler sağ ok yardımıyla "Assigned Factors" alanına yönlendirilmiştir.

Sadece proje yöneticisi ve test yöneticisi tüm faktörleri puanlamadan sorumludurlar. Programcılar karmaşıklık, değişiklik sıklığı, yeni teknolojiler, zaman baskısı, görünür alanlar ve kritik alanlar faktörleri ile ilgili puanlamalar için atanmışlardır. Yazılım mimarları karmaşıklık, değişiklik sıklığı, yeni teknolojiler ve kritik alanlar faktörleri ile ilgili puanlamalar için atanmışlardır. Test uzmanı değişiklik sıklığı, hata geçmişi, görünür alanlar, en çok kullanılan alanlar, gereksinimlerin kalitesi ve zaman baskısı faktörleri ile ilgili puanlamalar için atanmışlardır. Yönetim seviyesinden seçilen paydaş ise, kritik alanlar, en çok kullanılan alanlar, iş açısından önem ve tekrar çalışma maliyeti faktörleri ile ilgili puanlamalar için atanmışlardır.

7.6.3 Başlangıç toplantısı aşamasının uygulanması

Altıncı bölümde anlatıldığı gibi PRISMA yaklaşımının üçüncü aşaması başlangıç toplantısı aşamasıdır. Bu aşama test yöneticisi tarafından yürütülmüştür ve paydaşlara ürün değerlendirme işlemi açıklanmıştır. Öncelikle paydaşlara PRISMA süreci açıklanmıştır. Sonrasında da sürecin uygulanacağı proje ayrıntılı olarak anlatılmıştır. Son olarak da paydaşlarla anlaşmaya varılmıştır.

a) Sürecin açıklanması

Risk odaklı test yaklaşımının paydaşlar tarafından anlaşılması, ürün risk matrisinin değerlendirilmesini sağlar. Seçilen projede tüm paydaşlara PRISMA süreciyle ilgili eğitim verilmiş ve geri bildirimler alınarak faktörler ve risk maddeleri ile ilgili ortak karara varılmıştır. Ürün risk değerlendirmesine dayalı test sürecinin işletilmesinin önemi ve faydası bu toplantıda açıklanmıştır. Bireysel hazırlıkların ne şekilde yapılacağı, PRISMA aracının kullanımı gibi konular bu toplantıda anlatılmıştır. Son teslim tarihi belirlenmiş ve tarih ile ilgili ortak karar alınmıştır. Paydaşlara puanlama yapacakları faktörler detaylı bir şekilde açıklanarak, faktörlerin anlamları ve yapılacak puanlama için belirlenen

değer aralıkları bildirilmiştir test yöneticisi her paydaşa süreçteki rolünü ve o paydaşın projeye olan katkısını anlatmıştır.

b) Ürünün açıklanması

Projede puanlama için seçilen paydaşların proje ile bilgileri vardır ancak bu aşamada analiz dokümanları, sistem mimarisi dokümanı ve gereksinim dokümanlarının da yardımıyla ürün ayrıntılı olarak açıklanmıştır. Seçilen maddelerin içerikleri, arayüz fonksiyonları, diğer sistemlerle olan bağlantı noktaları ile ilgili paydaşlara ayrıntılı bilgi verilmiştir.

c) Anlaşmaya varılması

Başlangıç toplantısının sonunda, paydaşlardan sürece katılım sağlayacaklarının ve takvime uyacaklarının sözü alınır. Test yöneticisi rolleri ve sorumlulukları toplantıda açık bir şekilde anlatmıştır.

7.6.4 Ayrıntılı risk tanımlaması aşamasının uygulanması

Seçilen projede, her test maddesi bir ürün riski olarak tanımlanmıştır. Planlama aşamasında gereksinimler başlangıçtaki ürün risklerinin kümesini tanımlamakta kullanılmıştır ancak ürün risk değerlendirme başlangıcında tüm gereksinimler bilinemeyeceği için süreç burada devam ettirilerek sonlandırılmıştır. Paydaşlarla beyin fırtınası yapılarak sistem ve arayüzlerle ilgili 3 tane en kötü durum senaryosu oluşturulmuştur. Bu senaryonun risk maddelerinin ne kadarını kapsadığına bakılmıştır ve risk maddeleri bu doğrultuda güncellenmiştir.

7.6.5 Bireysel hazırlık aşamasının uygulanması

Altıncı bölümde anlatıldığı gibi PRISMA yaklaşımının beşinci aşaması bireysel hazırlık aşamasıdır. Bu aşamada her ürün riski için paydaşlar olasılık ve etki faktörlerini analiz ederek ürün risk analiz sürecine bağımsız girdiler sağlamıştır. Paydaşlar her ürün risk maddesi için kendi üzerlerine atanmış faktörleri puanlayarak, test yöneticisine sunmuşlardır. Paydaşların birbirlerinden etkilenmemeleri için bu aşama bağımsız yürütülmüştür.

a) Her risk maddesi için faktörlerin puanlanması

Paydaşlar risk maddelerini sütunlardaki faktörlere göre değerlendirerek puanlama yapmışlardır. Puanlama aralığı 1 ile 5 olarak belirlenmiştir. 1 en

düşük riskli, 5 ise en yüksek riskli değer olarak alınmıştır. Paydaşların 1 ile 5 arasındaki puanlarda eşit dağılım yapmaları için bir kural tanımlanmıştır. Yani paydaşlar tüm maddeleri 1 ya da tüm maddeler 5 olarak puanladıklarında uyarı alacaklardır. Aşağıda örnek bir puanlama tablosu görülmektedir. Soldaki kolonda risk maddeleri vardır. Sağ tarafta ise sırasıyla olasılık ve etki faktörleri paydaşın sorumluluk alanına göre sıralanmıştır. “Distribution” alanında yazan “OK” yazısı dağılımın o faktör için doğru yapıldığını göstermektedir. Bu alanın “NOT OK” olduğu faktörler için paydaş eşit dağılım yapmamış demektir. Bu durumun olduğu paydaşlar test yöneticisi tarafından uyarılmıştır ve puanlamayı tekrar gözden geçirmişlerdir.

b) Tahminlerin dokümantasyonu

Paydaşlar tarafından verilen puanlar, paydaşların olasılık ve etki faktörlerine bakış açısına göre değişir. Algılanan riskler bireysel tahminlere ve paydaşlar tarafından kullanılan gerekçelere bağlıdır. Bu tahminler ve gerekçeler daha sonraki toplantıda tartışılmak üzere, dokümanite edilmiştir.

c) Kişisel gözden geçirme

Başlangıç toplantısı sırasında açıklanan ve tanımlanan kurallar çerçevesinde paydaşlar kendi puanlarını kontrol etmiştir. Paydaşlar faktör ve risk maddeleri üzerinden olası puan değerlerini uygun şekilde paylaştırıp paylaştırmadığına ve her ürün risk maddesi için kendisine atanan faktörlere bir puanlama yapıp yapmadığına bakmalıdır. Gözden geçirme bittikten sonra paydaşlar belirlenen tarihe kadar puanlamalarını yapıp test yöneticisine göndermişlerdir.

d) Test yönetimi

Bu aşamada, test yöneticisi daha az deneyimli paydaşları desteklemiş ve soruları yanıtlamıştır. Test yöneticisi süreci açıklamalı, kurallar ve faktörleri detaylı şekilde anlatmalı ve puanlama çalışmasında paydaşlara yardımcı olmalıdır.

Bu süreç faktörlerin anlaşılması aşamasını da içerdiği için 10 gün olarak belirlenmiştir. Süreç, proje ve test planlama aşamasıyla paralel sürdürülmüştür.

7.6.6 Bireysel puanların işleme alınması aşamasının uygulanması

Altıncı bölümde anlatıldığı gibi PRISMA yaklaşımının altıncı aşaması bireysel puanların işleme alınma aşamasıdır. Bu aşamada test yöneticisi puanlamaların doğru yapıp yapılmadığını kontrol etmiş ve ortalama puanları hesaplayarak bireysel puanları işleme almıştır. Olasılık ve etki faktörleri için taslak ortalama değerler hesaplanmıştır. Paydaşlardan alınan dokümanlar gözden geçirilerek ve puanlamalardaki yığılmalar da göz önüne alınarak ortak görüş toplantısı için ön hazırlık yapılmıştır.

a) Puanlamanın kontrol edilmesi

Test yöneticisi paydaşların puanlarını kontrol ederek tanımlanan ve karşılaştırılan kurallara uyulmadığı durumlarda paydaşlarla bire bir toplantı

yapmıştır.Faktörün veya puanlama kurallarını anlamayan paydaşlar için puanlama süreci tekrar anlatılmıştır.Bir paydaş tahminlerini özellikle yüksek puanlama değerine göre yaptığı için o paydaş tekrar puanlama yapmıştır. Paydaş ve test yöneticisi arasında çözülemeyen problemler görüş toplantısı için konu listesine eklenmiştir.

b) Bireysel puanların işleme alınması

Test yöneticisi bireysel puanları kullanarak ortalama puanları hesaplamıştır.Her risk maddesinin tüm olasılık ve etki faktörleri için ortalama puanı çıkarılmıştır. Her ürün risk maddesi için olasılık faktörlerinin ortalama değerleri toplanmış ve sonuç olarak olasılık faktörleri için toplam puanlama taslağı oluşmuştur. Aynı şekilde her ürün risk maddesi için etki faktörlerinin ortalama değerleri toplanmış ve sonuç olarak etki faktörleri için toplam puanlama taslağı oluşmuştur. Böylece her risk maddesi PRISMA ürün risk matrisinde konumlandırılmıştır. Şekil 7.5'de 3 numaralı test maddesinin (Hotel Searching) ortalama puanlarının hesaplandığı ekran görüntüsü vardır.

| Detailed Overview - Hotel Searching (3) | | | | | | | | | | | |
|-----------------------------------------|-------------|------------------|----------------|---------------|--------------|----------------|----------------|----------------|-----------------|-----------------|----------|
| Hotel Searching (...) | Karmaşıklık | Değişiklik Ya... | Yeni Teknol... | Zaman Bask... | Hata Geçmişi | Gereksiniml... | Görünür Ala... | Kritik Alanlar | En Çok Kulla... | İş Açısından... | Tekri... |
| Final | 4 | 3,7 | 4 | 4,2 | 4,5 | 2,5 | 4,5 | 4,1 | 2,7 | 3,8 | 3 |
| Proposed | 4 | 3,7 | 4 | 4,2 | 4,5 | 2,5 | 4,5 | 4,1 | 2,7 | 3,8 | 3 |
| Doğa Serdaroğlu | 5 | 4 | 4 | 4 | 4 | 2 | 4 | 4 | 3 | 3 | 3 |
| Simay Yılmazlar | | | | | | | 4 | 4 | 3 | 4 | 3 |
| Gülfer Men | 3 | 3 | 4 | 4 | 5 | 3 | 4 | 4 | 2 | 4 | 3 |
| Ece Açı | 4 | 3 | 3 | 5 | | | 5 | 4 | | | |
| Efecan Çetir | 3 | 5 | 4 | 5 | | | 5 | 4 | | | |
| Kemal Gültekin | 5 | 4 | 4 | | | | | 5 | | | |
| Engin Kırmacı | 4 | 4 | 5 | 3 | | | 5 | 5 | | 4 | |
| Günkut Zeybek | 4 | 3 | 4 | | | | | 3 | | | |

Şekil 7.5 Örnek ortalama puan tablosu

c) Konu listesi

Daha önceki aşamalarda, tartışılmaya açık konular görüş toplantısı için bir liste haline getirilmiştir. Konu listesi, tanımlanmış kural kümesine uymayan tüm risk maddelerini içerir. Test yöneticisi ve paydaşlar ürün risk analizinde büyük

ölçüde önemi olan konuları, sayılmayan ürün riskleri, faktörlerin ağırlıkları, uygulanabilirliklerini bu toplantıda konuşmuşlardır.

7.6.7 Paydaş görüş toplantısının yapılması

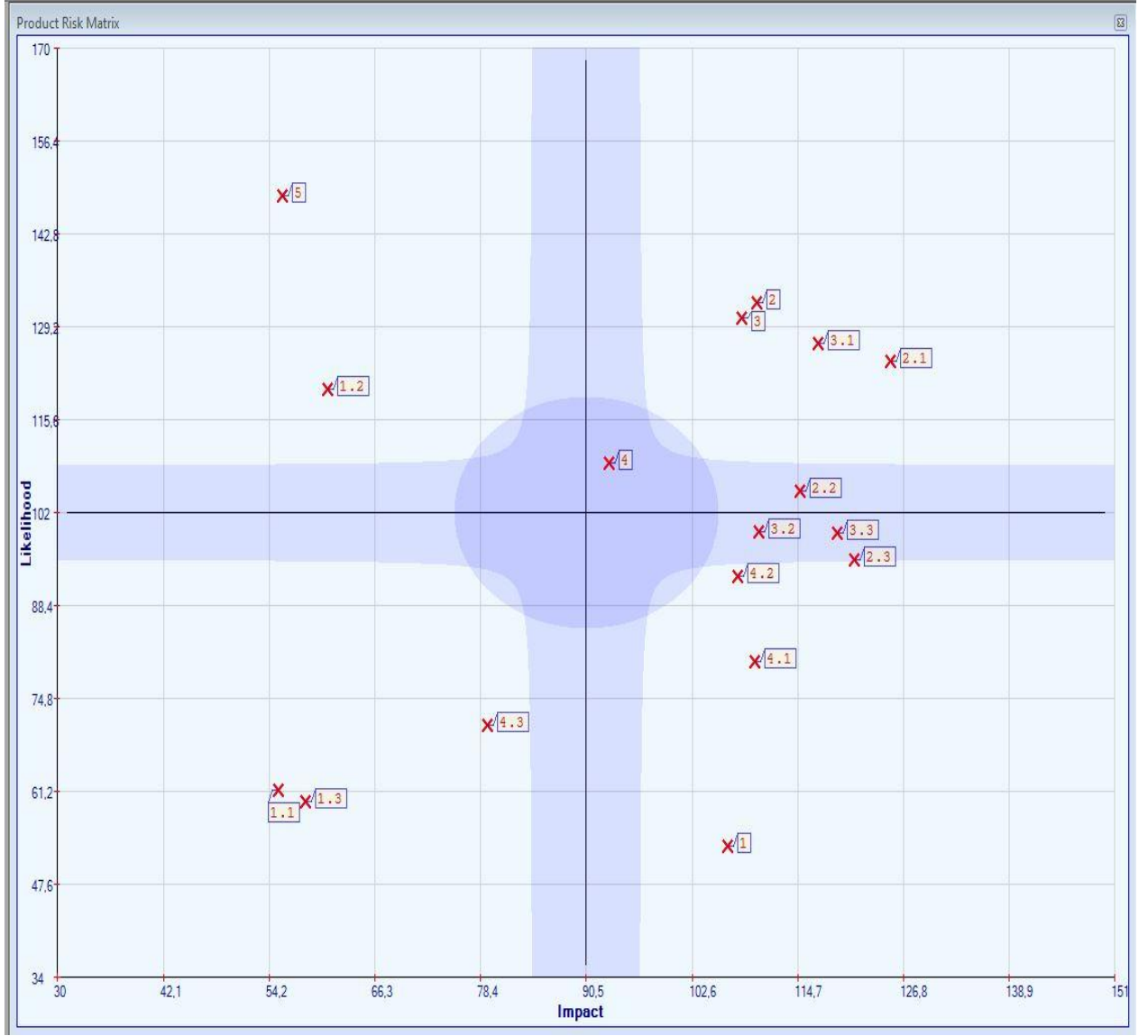
Altıncı bölümde anlatıldığı gibi PRISMA yaklaşımının yedinci aşaması paydaş görüş toplantısı aşamasıdır. Bu aşamada test yöneticisi tüm paydaşların katılım sağlayacağı bir toplantı ayarlar. Önceki aşamalarda test yöneticisinin hazırlamış olduğu konu listesinin üzerinden gidilerek karar verilmeyen konulara karar verilmiş ve eksiklikler giderilmiştir. Taslak ürün risk matrisinin üzerinden gidilerek tüm maddeler ile ilgili paydaşlar görüşlerini sunmuştur. Sonuç olarak test sürecinin ilerleyeceği, ana ürün risk matrisi ortaya çıkmıştır.

a) Konu listesini işleme koymak

Paydaşlar birbirlerinin yapmış olduğu tahminler ile ilgili yorumları bu toplantıda paylaşmışlardır. Belli risk maddeleri ile ilgili en düşük ve en yüksek puanlama yapan paydaşlar bunların nedenlerini açıklamıştır. Özellikle bu maddeler üzerinden gidilerek ürün risk matrisinde bu maddelerin pozisyonları güncellenmiştir.

b) Matris alanını analiz etmek

Şekil 7.6'da risk matrisiyle dağıtılmış olan ürün risklerini gözlenebilir. Birkaç risk maddesi matrisin uç köşelerindedir. (1.1),(1.3) ve (4.3) numaralı risk maddeleri gibi düşük etkili ve düşük olasılıklı maddeler test sırasında göz ardı edilebilir durumdadır. (2),(3),(2.1),(3.1) numaralı risk maddeleri gibi yüksek etkili ve yüksek olasılıklı maddelere test sürecinde çok daha fazla dikkat edilmelidir. Matrisin orta kısmına yerleştirilen (4) numaralı risk maddesi ile ilgili karar vermek çok daha zordur. Burada risk azaltma yaklaşımının uygulanmasından önce bazı analizler gerekli olabilir. Normalde orta eksene yakın tüm risk maddeleri test yöneticisinin konu listesine dahildir ve tartışılması gerekir. Analizden sonra hangi çeyreğe oturtulmaları gerektiğine karar verilmelidir.

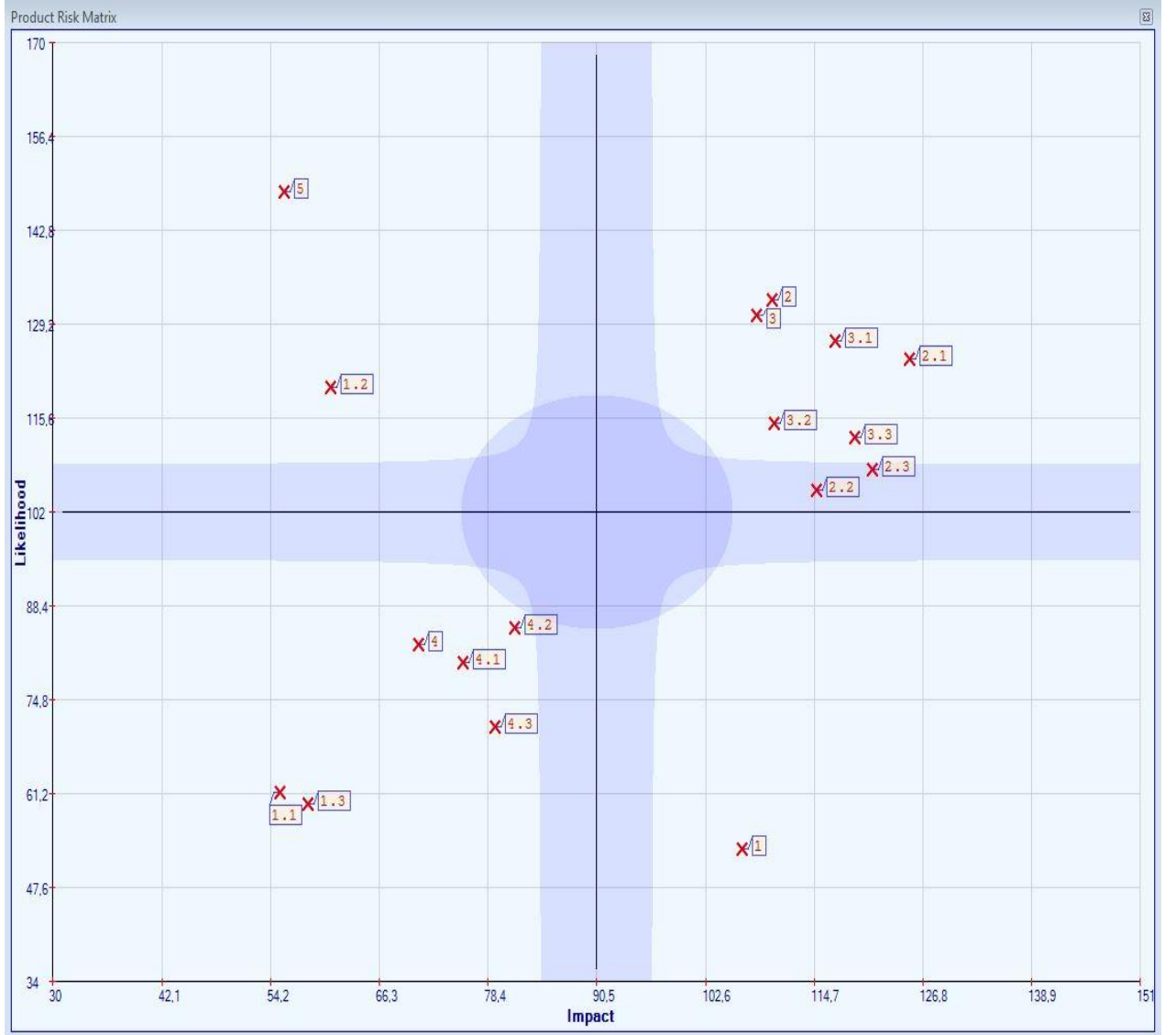


Şekil 7.6 Taslak ürün risk matrisi

c) Risk seviyelerinin doğrulanması ve özetlenmesi

Toplantının sonunda final puanlarına karar verilerek ve kullanılacak olan ürün risk matrisinin son hali ortaya çıkmıştır. Üzerinde anlaşmaya varılamayan test maddeleri ile ilgili konuda en uzman olan paydaşın görüşü referans alınarak puanlama yapılmıştır. (4) numaralı test maddesinin düşük öncelikli test maddeleri arasına konmasına karar verilmiştir. Toplantıda matriste etki faktörleri yüksek çıkan (4.1) ve (4.2) maddelerinin etkisinin beklendiği kadar yüksek olmayacağına karar verilmiştir. (3.2), (3.3) ve (2.3) numaralı maddeler (2) ve (3) numaralı maddelerin testlerine bağımlı oldukları için ve yüksek risk içeren II numaralı alana yakın oldukları için yüksek öncelikli risk maddeleri

olmalarına karar verilmiştir. Bu matris tüm paydaşlar tarafından doğrulanmış ve tartışılması gereken başka bir konu kalmamıştır. Karar verilen ürün risk matrisinin son hali Şekil 7.7’de görülebilir.



Şekil 7.7 Karar verilen ürün risk matrisi

7.6.8 Farklılaştırılmış risk odaklı test yaklaşımının uygulanması

Altıncı bölümde anlatıldığı gibi PRISMA yaklaşımının son aşaması farklılaştırılmış risk odaklı test yaklaşımı aşamasıdır. Bu aşamada üzerinde anlaşmaya varılan ürün risk matrisine uygun bir test yaklaşımı tanımlanarak tüm test yapanların test aktivitelerini ürün risklerine göre uygulandığından emin olmak için denetim süreci geliştirilmiştir. Dört çeyrek bölgedeki ürün risk maddeleri farklı şekilde test edilmiştir. Bir maddenin test edilmesinden kimin sorumlu olduğu, nasıl ve ne zaman

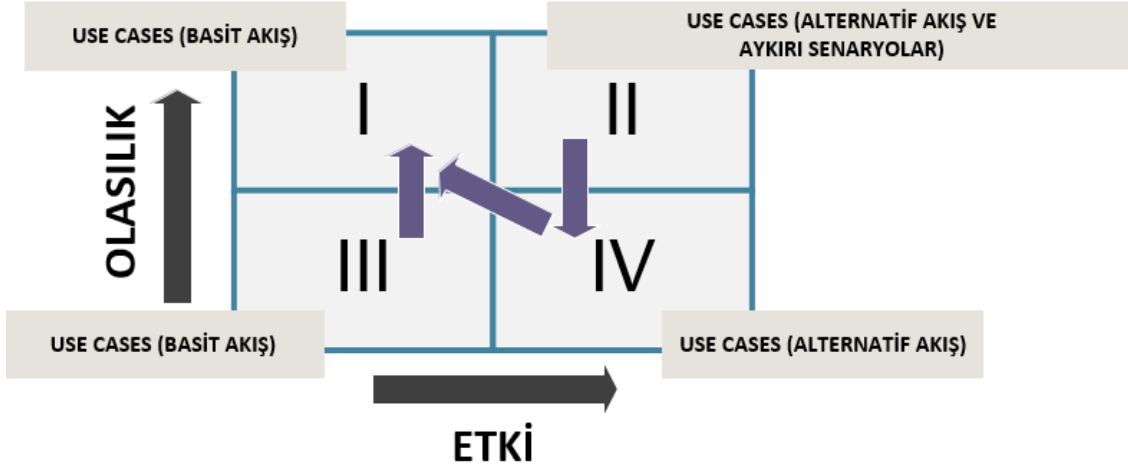
hangi maddenin test edilmesi gerektiği ve ürün risk değerlendirme sonuçlarıyla örtüşüp örtüşmediği dokümente edilmiştir.

a) Önceliklendirmenin yapılması

Ürün risk matrisindeki lokasyonlara bağlı olarak, test maddeleri önceliklendirilmiştir. Test için sıraya konan tüm risk maddelerinin sonucunda en riskli olan madde test planında ilk sıraya alınmıştır. Geliştirme gerçekleştirilirken test öncelikleri ile ilgili bilgi verilmiştir, test yöneticisi ve proje yöneticisi bunu analiz ederek ve geliştirme, entegrasyon ve hizmet takviminin önceliklerini belirlemişlerdir.

Ortaya çıkan son ürün risk matrisinden referans alınarak en yüksek öncelikli risk maddesi (2.1) numaralı maddedir. Yüksek risk seviyesinde olan toplam 8 tane madde vardır. Bunlar sırasıyla (2.1), (3.1),(3.3),(2.3),(2),(3),(2.2) ve (3.2) numaralı test maddeleridir. (3.2), (3.3) ve (2.3) numaralı maddeler paydaşların toplantıda aldıkları kararlar doğrultusunda yüksek öncelikli olarak işaretlenmişlerdir.

Düşük risk seviyesinde olan maddeler ise sırasıyla (1.1),(1.3), (4.3),(4),(4.1) ve (4.2) numaralı risk maddeleridir. Burda da aynı şekilde (4.1) ve (4.2) maddeleri paydaşların toplantıda aldıkları kararlar doğrultusunda düşük öncelikli olarak işaretlenmişlerdir.(5) ve (1.2) numaralı test maddeleri için ise olasılık faktörlerinin gerçekleşme olasılıkları oldukça yüksek ancak oluşurlarsa etkileri düşük olacak maddelerdir.(1) numaralı test maddesi ise sistemin giriş ekranıdır. Bu ekranda problem çıkma olasılığı düşük ancak bir problem çıkarsa yaratacağı etki yüksek olacaktır. Şekil 7.8'de matristeki pozisyonlara yönelik test yaklaşımları görülebilir.



Şekil 7.8 Ürün risk matrisi çeyreklerine yönelik test yaklaşımları

Şekil 7.8'de görüleceği gibi en kritik bölge olan II bölgede senaryolar karar ağacı yapısı şeklinde oluşturulmuştur. Bu bölge kritik bölge olduğu için aykırı senaryo sayıları diğer bölgelere göre oldukça fazladır. Önceliklendirmeler oklar yönünde yapılmıştır. İkinci yüksek öncelikli test alanı IV. Bölgedir. Senaryolar alternatif akışları ile birlikte oluşturulmuştur. II. bölgedeki karar ağacı test yapısı test tekniğine göre çok daha az sayıda test senaryosu yazılmıştır. Üçüncü öncelikli test alanı I numaralı alandır. Bu bölgede senaryolar yalnızca basit akış şeklinde oluşturulmuştur. Test senaryosu sayısı fazladır ancak test senaryoları basit yapıdadır. Son olarak en düşük öncelikli alan III numaralı alandır. Bu alanda basit akışlı senaryolar uygulanmıştır. Testlerin önceliklendirilmesinde etkisi yüksek olan maddelerin testleri olasılığı yüksek olan maddelere göre öncelik kazanmıştır.

b) Test tasarım tekniklerinin belirlenmesi ve geliştirme yaklaşımı

Yüksek riskli maddeler için daha deneyimli test uzmanları atanmıştır ve test etme yoğunluğu bu maddeler için artırılmıştır. Test süreci geliştirme sürecine paralel ilerlediği için bu maddeler çevik yaklaşıma uygun test edilmiştir. Aynı zamanda test planına da sadık kalınmıştır. Risk en önemlisi olsa da kullanılacak test tasarım tekniklerini seçmede tek faktör olmadığı için beklenen hata tipleri, yapılan dokümantasyon, kullanılan modeller, müşterinin öncelikleri zaman ve bütçe de göz önüne alınmıştır. Tanımlanan ve analiz edilmiş ürün risklerine dayanarak kalite ekibi dışardan bir bakış açısıyla gözden geçirmeler yapmıştır. Olasılık faktörü yüksek olan ürün riskleri için teknik destek alınarak

gözden geçirmeler tamamlanmıştır. Belirlenen test yaklaşımları ve önceliklerin anlaşılması için bir toplantı organize edilmiştir. Toplantı son kullanıcı, testçi, yazılım geliştirici ve paydaşlarla yapılmıştır. Toplantı sırasında test maddesi için test düşünceleri ve kritik konular tartışılmıştır. Önemli ürün risklerinin test tasarımları paydaşlar ile tartışılmıştır.

Kritik risk maddeleri için daha fazla dokümantasyon yapılmıştır böylece test senaryolarının gözden geçirmelerini yapmak kalite ekibi ve paydaşlar için daha kolay hale gelmiştir. Test senaryoları ve test prosedürlerini hazırlayan test uzmanı testlerin uygulamasını yapan test uzmanından ayrı tutulmuştur. Böylece testi koşturan kişi test senaryolarına bağımsız bir bakış açısı ile bakarak daha çok hatanın ortaya çıkmasını sağlamıştır.

TMMi modeli 2. seviyedeki Test Planlama aşaması süreç alanları PRISMA yaklaşımı kullanılarak uygulanmıştır. Bu süreç sonucunda elimizde ürün risk matrisine göre önceliklendirilmiş ve ayrıntılandırılmış bir test planı vardır. Şimdi TMMi modeli 2. Seviyesinin diğer süreç alanlarını uygulayabiliriz.

7.7 Testin İzlenmesi ve Kontrolü Süreç Alanının Uygulanması

TMMi modeli 2. seviyedeki üçüncü süreç alanı testin izlenmesi ve kontrolünün yapılması aşamasıdır. Bu süreç alanında sırasıyla aşağıdaki adımlar uygulanmıştır.

- Testin plana uygun ilerleyişinin izlenmesi,
- Plan ve beklentilere göre ürün kalitesinin izlenmesi,
- Test bitiş aktiviteleri için düzeltici faaliyetlerin yönetilmesi.

Testin planına uygun ilerleyişinin izlenmesi aşamasında, PRISMA yaklaşımı sonucu ortaya çıkan test planı, test takviminin bitişine kadar izlenmiştir. Periyodik olarak test görevlerinin güncel sonuçlarına bakılarak, test planında dokümante edilmiş olan test takvimiyle karşılaştırılmıştır. Test planında yapılan tahminlere karşı oluşan önemli sapmalar belirlenmiştir. Aynı şekilde periyodik olarak teste ayrılan bütçe, efor ve bu iş için atanan personel sayısına bakılarak, test planında yapılan tahminler karşılaştırılmıştır ve yine beklenenden büyük sapmalar varsa bunlar belirlenerek önlemler alınmıştır. Test için istenilen ve kullanılan test ortamı kaynakları karşılaştırılmıştır ve eksikler giderilmiştir. Projenin ilerleyen safhalarında

test maddelerinde ortaya çıkan ürün riskleri gözden geçirilerek ürün risk matrisi güncellenmiş ve test planında önceliklendirmeler değiştirilmiştir. Paydaşların proje sürecindeki gözden geçirmeleri takip edilmiştir ve önemli gördükleri konular kayıt altına alınmıştır.

Test süreci boyunca, test ilerleme raporu tutulmuştur ve test ilerleme ölçütleri toplanarak analiz edilmiştir. Test ilerleyişinin güncel durumu ve performans göstergeleri paydaşlarla paylaşılmıştır. Yazılım ürününde ortaya çıkan değişiklik istekleri, test ilerleyişindeki büyük problemler tanımlanarak dokümente edilmiştir. Test süreci boyunca test için atanan çalışanların bilgi seviyesi ve yetenekleri güncel olarak kontrol edilerek , bilgi eksikliklerinin test planını etkileyip etkilemediğine bakılmıştır. Test planındaki her aşamanın bitişinde ilgili paydaşlara test ilerleme analizleri gönderilmiştir.

Ürün kalitesinin plan ve beklentilere göre izlenmesi aşamasında başlangıçta konuşulan kalite ölçütlerine karşılık ortaya çıkan ürün kalitesine bakılmıştır. Seçilen projede ürünün kullanılabilirliği, performansı, güvenilirliği en önemli kalite ölçütlerindedir. Bu aşama da en büyük problem ürünün performansı ile ilgili olmuştur. Web tabanlı bir program olduğu için sayfaların açılma süreleri oldukça geçtir ve müşterinin beklentilerini karşılamamaktadır. Bu durum karşısında performans testleri arttırılmış ve performansı arttırabilecek köklü değişiklikler yapılmıştır.

Tüm test süreci boyunca ortaya çıkan hatalar izlenmiş ve bir hata yönetim aracına kayıt edilmiştir. Beklenen ve tahmin yürütülen hata oranlarıyla karşılaştırmalar yapılmıştır. PRISMA sürecinde önceliklendirilmiş olan test maddeleri ile ilgili toplam hata sayısı, çözümlenen ve çözümlenemeyen hata sayısı, her test tipi için bulunan hata sayısı, bin satır kod başına düşen hata sayısı gibi veriler hata ölçütleri olarak kayıt altına alınmıştır. Toplamda proje boyunca 1378 hata bulunmuştur. Bunların 428 tanesi yüksek öncelikli maddeler ile ilgilidir.

Daha önce planlanan bitiş kriterlerine bakılarak mevcut test kapsamıyla örtüşüp örtüşmediği karşılaştırılmıştır. Plandaki beklentilerin karşılanmadığı durumlar kayıt altına alınmıştır. Kayıt altına alınan veriler erteleme ya da yeniden başlama

kriterleri ile karşılaştırılarak , bu yönde karar verilmiştir ve bitiş kriterlerini desteklemeyen bazı test senaryoları tekrarlanmıştır.

Test bitiş aktiviteleri için düzeltici faaliyetlerin yönetilmesi aşamasında düzeltilmesi gereken konular listelenmiştir . Bu konular gözden geçirme yapılan paydaşlardan, süreç boyunca test planına uygun olmayan durumlardan oluşan bir listedir. Bu listede belirtilen düzeltici faaliyetler paydaşlarla da konuşularak uygulanmıştır. Örneğin müşteriyle yapılan sözleşme güncellenmiştir ayrıca performans testlerinin yapılabilmesi için kaynak eklenmiştir. En önemlisi canlıya çıkış tarihi iki hafta kadar ertelenmiştir.

7.8 Test Ortamı Hazırlık Süreç Alanının Uygulanması

TMMi modeli 2. seviyedeki dördüncü süreç alanı test ortamının hazırlanması aşamasıdır. Bu süreç alanında sırasıyla aşağıdaki adımlar uygulanmıştır.

- Test ortamı gereksinimlerinin oluşturulması,
- Test ortamının hazırlanması,
- Test ortamlarının yönetilmesi ve kontrolü.

Test ortamı gereksinimlerinin oluşturulması aşamasında paydaşların beklentileri ve istekleri toplanarak test ortamı gereksinimleri haline getirilmiştir. Seçilen projede network bileşenleri, arayüz bileşenleri, test verileri üretme araçları, test verilerinin deponlaması ve geri yüklenbilmesi test ortamı gereksinimlerinden bir kaçıdır.

Test ortamının hazırlanması aşamasında, test ortamı test tiplerine ve test seviyelerine göre önceliklendirilmiştir. Örneğin test senaryolarının uygulanması için test ortamı kurulurken, kullanıcı kabul testleri için canlı ortamla birebir aynı olan “Pre-production” ortamı kurulmuştur. Bazı arayüzlerde çok daha fazla sayıda farklı veriyle test yapılması gerektiği için, gerekli olan test verileri belirlenmiştir. Testlerin koşturulması için çeşitli test verileri üretilmiştir.

Test ortamlarının yönetilmesi ve kontrolü aşamasında , test ortamlarında oluşabilecek olan hatalar raporlanmıştır ve operasyon ekibinin düzeltmesi için atama yapılmıştır. Beyaz kutu testlerinin, kara kutu testlerinin ve performans

testlerinin aynı anda yapılabilmesi için test ortamlarının kullanım takvimleri oluşturulmuştur.

7.9 Test Tasarımı Süreç Alanının Uygulanması

TMMi modeli 2. seviyedeki beşinci süreç alanı test tasarımının yapılması ve uygulanması aşamasıdır. Bu süreç alanında sırasıyla aşağıdaki adımlar uygulanmıştır.

- Test tasarım tekniklerinin kullanılarak test analiz ve tasarımının yapılması,
- Test uygulamasının gerçekleştirilmesi,
- Test koşturma işleminin gerçekleştirilmesi,
- Test bitiş aktiviteleri için hata sürecinin yönetilmesi.

Test tasarım tekniklerinin kullanılarak test analiz ve tasarımının yapılması aşamasında, test tasarım teknikleri kullanılarak test senaryoları ve test durumları oluşturulmuştur. Web arayüzündeki her sayfa için en iyi kullanıcı senaryosu adım adım oluşturularak , bu senaryodan ortaya çıkabilecek olan aykırı senaryolar, test adımları değiştirilerek oluşturulmuştur. Tüm test durumları “Test Manager” aracına girilerek gereksinimler ve ürün riskleri ile izlenebilirlik oluşturulmuştur. Geçme ve kalma kriterleri belirlenmiştir. Her adım için karşısına beklenen sonuç yazılmıştır. Şekil 7.9’da bir test senaryosu örneği vardır.

Ürün risk matrisinde ortaya çıkan risk yoğunluklarına göre, risk faktörü yüksek olan arayüzlerin test senaryolarına öncelik verilmiştir. Bu arayüzler için daha fazla aykırı senaryo oluşturulmuştur. Test senaryoları, test durumları ile gereksinimler arasında iki taraflı izlenebilirlik kurulmuştur ve bu haftalık gözden geçirmeler ile kalite ekibi tarafından denetlenmiştir.

Test uygulamasının gerçekleştirilmesi aşamasında bir önceki aşamada belirlenen test verilerine uygun otomasyon yapılmıştır. Test verilerinin değiştirilerek aynı senaryonun uygulanması sağlanmıştır. Test koşturma aşamasına başlamadan önce test ortamlarında, test ortamının doğru çalışıp çalışmadığını ya da bir sorun olup olmadığının görülmesi için yüzeysel sistem testleri(*smoke test*) yapılmıştır. Arayüzlerin birbirleriyle entegre çalışıp çalışmadığı kontrol edilmiştir.

Son olarak bir sonraki aşamada gerçekleştirilecek olan test kořturma iřlemi iin, rn risk matrisi gz nne alınarak test kořturma takvimi oluřturulmuřtur.

Testlerin kořturulması ařamasında bir nceki adımda belirlenen takvime uygun bir řekilde testler kořturulmuřtur. Testler sırasında ortaya ıkan sıradıřı durumlar kayıt altına alınmıřtır. Test senaryolarında, test verilerinde ya da test doęrulama adımlarında hata olduęu farkedilen testler kayıt altına alınarak ilgili test uzmanlarına ynlendirilmiřtir.

Ortaya ıkan test sonuları beklenen ve planlanan test sonularıyla karřılařtırılmıřtır. Hatanın ortaya ıktıęı durumlarda bu hatanın yazılımdan mı yoksa testten mi kaynaklandıęına bakılması iin ilgili ekiplere ynlendirilmiřtir. Test kořturma ařamasındaki test verileri ve ortaya ıkan test ıktıları aıklamaları ile birlikte dokmante edilmiřtir.

| Action | Expected Result |
|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Login Page(BZQ) | |
| 2. Hotel Free Text Searching(BZC) | |
| 3. Select a board type on "Board Type" area. | N/A |
| 4. Enter price range to "Price Per Night" area. | N/A |
| 5. Select some of the type of hotels from "Type of Hotels" selections. | N/A |
| 6. Click "Search" button. | System should be displayed hotels right of the page according to selected options. Hotel name, picture, description, stars, location information and prices with different type of rooms should be shown. |
| 7. Select "Sort By" option. | Hotels should be sorted according to sorting criteria. |
| 8. Select a hotel. | N/A |
| 9. Click "Info" link which is under the hotel price. | Page should be directed to Hotel Description page. Weather condition for hotel location and for related dates, selected city, departure and arrival dates, number of room and number of people, hotel description, hotel type, price per night, hotel photos, room facilities, and activities should be shown. |
| 10. Click "Photos" button. | Hotel Photos page should be opened. Pool, room, restaurant, bar, sports, and general photos of hotel and room type, board, per night price, total price, city, dates should be shown right of the page. Destination or hotel name, dates and number of people and number of room should be shown left of the page. |
| 11. Click "Zoom Plus" icon of picture. | Picture should become bigger. |
| 12. Click "Zoom Minus" icon of picture. | Picture should become smaller. |
| 13. Click "Map" button. | Hotel Map page should be opened. Location of the hotel, destination, dates and number of people and number of room left of the page should be shown. |
| 14. Click "Information" button again. | Hotel Description page should be opened. |
| 15. Select a hotel. | N/A |
| 16. Select a room type. | N/A |
| 17. Click "Book" button. | Page should be directed to "Visitor Information" page. Booking details which are hotel and room information with nightly price and final price with taxes should be shown on "Hotel Purchasing/Visitor Information" page. |

Şekil 7.9 Test senaryosu örneği

Test bitiş aktiviteleri için hata sürecinin yönetilmesi aşamasında bulunan hatalar gözden geçirilerek analiz edilmiştir. Bulunan hatalar ürün risk matrisindeki önceliklerine ve önem seviyelerine göre sıralanmıştır. Gözden geçirmeler sonucunda bulunan hatalar yazılım ile ilgiliyse ilgili yazılım uzmanına, test verileri ya da senaryoları ile ilgiliyse test uzmanına yönlendirilmiştir. Gerekli düzeltmeler yapıldıktan sonra test senaryoları tekrar koşturulmuştur. Test senaryoları dışında bu hataların giderilmesinden etkilenebilecek olan alanlar olabileceği için regresyon testleri de yapılmıştır. Test hata raporu da yeni sonuçlara göre tekrar düzenlenmiştir. Yeni çıkan ya da giderilemeyen hatalar için aynı süreç hata oranı en aza indirilene kadar işletilmiştir.

8. ÖZET VE SONUÇ

Bu çalışmada önce yazılım testi ile ilgili bilgiler derlenmiş, yazılım testi ile yazılımın kalitesi arasındaki ilişki irdelenmiştir. Daha sonra Uluslararası Yazılım Test Yeterlilik Kurulu'nun Yazılım Test Kılavuzu(ISTQB) özetlenmiştir.

TMMi (Test Maturity Model Integration) modelinin temel öğeleri anlatıldıktan sonra Risk Faktörü ve Ürün Risk Yönetimi konusundaki bilgiler derlenmiştir. Yapılan uygulama ile ilgili olduğu için bir sonraki konu olarak PRISMA(Practical Risk-Based Testing) ele alınarak özetlenmiştir.

Tez çalışmasının son bölümünde bir uygulama gösterilmiştir. Uygulama, TMMi modeli içinde bir PRISMA yaklaşımını göstermeyi amaçlamıştır. Sonuç olarak TMMi modelini 2. seviyede uygulayan bir organizasyon için planlama aşamasında PRISMA yaklaşımının kullanılması, 2. Seviyedeki tüm süreç alanlarının uygulanmasını kolaylaştırmaktadır. Risk odaklı test yaklaşımı, test önceliklerini belirleyerek test verilerinin , test senaryolarının ve test durumlarının hata çıkabilecek alanlara yönelik oluşturulmasını, deneyimli test uzmanlarının bu alanlara yönlendirilmesini sağlayarak test sürecinin daha sağlıklı ve etkili yürütülmesini sağlamıştır. Riskli alanları testler esnasında değil öncesinde ortaya çıktığı için test takviminde gecikme yaşanmamıştır. Riskli alanların varlığı projenin erken evrelerinde tespit edildiği için test bütçesi ve personel sayısı buna göre oluşturulmuş böylece ek bütçe ya da personele ihtiyaç duyulmamıştır. Test süreci ve geliştirme süreci paralel ilerlediği için riskli olan alanlarla ilgili hatalar önce tespit edilmiştir. Bu durum proje takviminin de fazla gecikmeye uğramasını önlemiştir.

Yapılan çalışmanın en önemli katkısı, Yazılım Test Süreci gibi karmaşık bir konuda göreceli olarak yeni sayılabilecek TMMi ve PRISMA gibi iki yeni metodolojinin kullanımını göstermesi ve Türkiye değil, yurtdışında da kısıtlı kaynak bulunan TMMi ve PRISMA konularında derleme yapılması ardından da gerçek bir uygulamada kullanılmasıdır.

İlerde yapılacak uygulama çalışmaları ile kullanılan teknikler daha özümserenerek farklı alanlardaki farklı problemlerde kullanılabilecektir.

Çalışmada güçlük çekilen konuların başındabu alandaki kaynakların kısıtlı oluşu ve uluslararası piyasada isim yapmış birkaç kişinin kendi yayınları ve ücretle verdikleri kurslara bağlı kalınmasıdır.

Tez çalışmasının devamı olarak;

- a. Gelecekte yapılacak çalışmalar arasında böyle bir yaklaşım kullanılmasının şirketlere sağlayacağı yararın irdelenmesi,
- b. TMMi uygulamasında PRISMA yerinde kullanılacak başka yaklaşımların tez çalışmasının yapıldığı son birkaç sene içinde kullanılıp kullanılmadığının belirlenmesi,
- c. Gerek TMMi gerekse PRISMA uygulamaları konusunda son birkaç sene içinde uluslararası düzeyde birkaç kitap dışında kaynağa rastlanmamıştır. Önümüzdeki yıllarda bu alandaki dergi ve konferans yayınlarının izlenmesi yararlı olacaktır. Bu paralelde tez çalışmasının bir poster sunumu, uluslararası bir konferansta kabul edilmiştir. Yine yapılacak yeni bir çalışma ile uluslararası düzeyde bir dergide yayınlamak üzere ortak bir makale hazırlanması da planlanmaktadır.

Farklı uygulamalarla tez çalışmasının sonuçlarının gelecekte daha yararlı bir hale gelebileceğine inanılmaktadır.

KAYNAKLAR LİSTESİ

- [1] BADGETT, T. and C, SANDLER, The Art Of Software Testing,Wiley,2012.
- [2] BRAUDE, E.J. and M.E, BERNSTEIN, Software Engineering Modern Approaches, Wiley, 2011.
- [3] CHRISSIS, M.B.,CMMI for Development: Guidelines for Process Integration and Product Improvement (3rd Edition), Addison Wesley, 2011.
- [4] ÇATAL,Ç., Yazılım Mühendisliği Yöntemleri ve İleri Konular, Papatya Yayıncılık, 2012.
- [5] HAMBLING, B., Software Testing, An ISTQB-ISEB Foundation Guide,2nd Edition,2010.
- [6] HOMES, B., Fundamentals of Software Testing, Wiley, 2011.
- [7] ISTQB Foundation Syllabus ,International Software Testing Qualification Board,2011.
- [8] JONES, C. and, O., BONSIGNOUR, Economics of Software Quality, Addison Wesley, 2011.
- [9] KAN, S.H., Metrics and Models in Software Quality Engineering, Addison Wesley, 1995.
- [10] KOOEN, T. and AALST ,L., Map Next, For Result-Driven Testing ,UTN Publishers, 2006.
- [11] KÖSE, Z.N., Toplam Kalite Yönetimi, ISO 9001 TKY Eğitimi,2012.
- [12] MACIASZEK, L.A., Requirements Analysis and System Design 3rd Edition, Addison Wesley, 2007.
- [13] MAJCHRZAK, Tim A., Improving Software Testing, Springer, 2012.
- [14] MATHUR, A., Foundations of Software Testing, Pearson, 2008.
- [15] NAIK, K., Software Testing and Quality Assurance, Wiley, 2008.

- [16] NIELSEN, B. and C. WEISE, (Eds.), Testing Software and Systems, Springer, 2012.
- [17] PFLEEGER, S.L. and J.M, ATLEE, Software Engineering, Pearson, 2010.
- [18] PRESSMAN, R.S., Software Engineering, Mc Graw Hill, 7th Ed, 2010.
- [19] SARIDOĞAN, M.A., Yazılım Mühendisliği, Papatya Yayıncılık, 2008.
- [20] SOGETI, TPI Next - Business Driven Test Process Improvement , UTN Publishers, 2009.
- [21] SOMMERVILLE, I., Software Engineering, Addison Wesley, 8th Ed., 2007.
- [22] SUMMERS, D.C.S, Quality, Prentice Hall, 2nd Edition, 2000.
- [23] TIAN, J., Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, Wiley, 2005.
- [24] VEENENDAAL, E., The PRISMA Approach, UTN, 2012.
- [25] VEENENDAAL, E., The PRISMA Tool, UTN, 2012.
- [26] VEENENDAAL, E., Test Maturity Model Integration, UTN, 2012.